TM78,TU78

TM78 PAK FILE
KKTMABO

AH-F143B-MC
FICHE 1 OF 7

FEB 1981
COPYRIGHT© 1980
MADE IN USA

TM78,TU78 TM78 PAK FILE KKTMABO AH-F143B-MC FICHE 2 OF 7 FEB 1981
COPYRIGHT@ 1980
MADE IN USA
d03020

TM78,TU78 TM78 PAK FILE KKTMABO AH-F143B-MC FICHE 3 OF 7 FEB 1981 COPYRIGHT© 1980 MADE IN USA

TM78,TU78 TM78 PAK FILE KKTMABO AH-F143B-MC FICHE 4 OF 7 FEB 1981 COPYRIGHT@ 1980 MADE IN USA

TM78,TU78 TM78 PAK FILE KKTMABO AH-F143B-MC FICHE 5 OF 7 FEB 1981 COPYRIGHT@ 1980 MADE IN USA

TM78,TU78 TM78 PAK FILE KKTMABO AH-F143B-MC FICHE 6 OF 7 FEB 1981
COPYRIGHT© 1980
MADE IN USA

TM78,TU78 TM78 PAK FILE KKTMABO AH-F143B-MC FICHE 7 OF 7 FEB 1981 COPYRIGHT© 1980 MADE IN USA

```
                                IDENTIFICATION
                                --------------

        PRODUCT CODE :          AC-F144B-MC

        PRODUCT NAME :          KKTMAB0 TM78 PAK FILE

        MAINTAINER :            DIAGNOSTIC ENGINEERING

        DATE :                  OCTOBER 1,1980

        AUTHOR :                G. COOKE
```


```
                        COPYRIGHT (C) 1980 BY
                DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
```

```
38
39
40                              ;                       HISTORY
41                              ;                       -------
42                              ;
43                              ;ORIGINAL RELEASE       KKTMAA0.PAK     JUNE 1,1980
44                              ;VERSION B              KKTMAB0.PAK     OCTOBER 1,1980
45                              ;
46                              ;                       EDIT HISTORY
47                              ;                       ---- -------
48                              ;
49                              ;VERSION A EDIT BY ROBERT G. VANDETTE
50                              ;
51                              ;1.UPDATED MODULE CALLOUTS AS A RESULT OF FAULT INSERTION
52                              ;2.MODIFIED MTA2 TO CHECK IF TU78 WAS REALLY OFFLINE
53                              ;3.ADDED "PUT TU78 ONLINE" MESSAGE TO MTA2
54                              ;4.ADDED NEW MICRO "RPM6" TO TEST AND CALLOUT FAULTY M8950
55                              ;
56                              ;
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(25) 15-OCT-80 09:12
PARAM.M80SEQ 000TABLE OF CONTENTS

3	"MEMORY MAP" EXTENDED ADDRESS BIT
19	READ MICRO CONTROLLER CONTROL & STATUS WORDS
139	TRANSPORT/PORT COMMAND + STATUS WORDS
380	COMMON ADDRESS SPACE REGISTERS
444	MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485	MASSBUS CONTROL WORD DEFINITIONS
557	WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709	KEYBOARD/DISPLAY CONTROL + STATUS WORD
797	MISCELLANEOUS VARIABLE DEFINITIONS
801	RAM ADDRESS DEFINITIONS
819	LINKAGE TABLE ADDRESSES
842	BIT DEFINITIONS
856	RIM AND SIM INSTRUCTION BIT DEFINITIONS
877	INTERRUPTABLE INSTRUCTION MACROS
924	DIAGNOSTIC VARIABLES
945	COMMON VARIABLE STORAGE
959	MACRO CALL DEFINITIONS
1047	TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101	8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1333	CPU/CAS INTERFACE PROGRAM
1367	WRITE THE CAS REGISTERS ROUTINE
1401	READ THE CAS REGISTERS ROUTINE
1421	CAS CONTENTION INT. ROUTINE
1483	PROGRAM VARIABLES

```
961                     ; ************************************************************************
962                     ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
963                     ;ERROR MACRO CALLS
964                     ;
965                     ;       1  -    REQUEST HOST CPU TO PRINT:
966                     ;               "BYTE/SCLK COUNT NUMBER = LLL"
967                     ;               WHERE:
968                     ;               LLL = THE VALUE STORED IN CAS REGISTER 5
969                     ;                     (THE BYTE COUNT REGISTER 16 BITS).
970                     ;
971                     ;       2  -    REQUEST HOST CPU TO PRINT:
972                     ;               DATA FORMAT = MM
973                     ;               SKIP COUNT  = NN
974                     ;               WHERE:
975                     ;               MM = DATA FORMAT FROM CAS REGISTER 2
976                     ;               NN = SKIP COUNT FROM CAS REGISTER 2
977                     ;
978                     ;       3  -    REQUEST HOST CPU TO PRINT:
979                     ;               BYTE-SCLK COUNT = LLL
980                     ;               DATA FORMAT = MM
981                     ;               SKIP COUNT  = NN
982                     ;               WHERE:
983                     ;               LLL = AS ABOVE
984                     ;               MM  = AS ABOVE
985                     ;               NN  = AS ABOVE
986                     ;
987                     ;       4  -    REQUEST HOST TO PRINT:
988                     ;               TRANSITION COUNT = LLL
989                     ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
990                     ;
991                     ;       5  -    REQUEST HOST CPU TO PRINT:
992                     ;               EXPECTED 18 BITS =      E EEEEEE
993                     ;               ACTUAL 18 BITS =        A AAAAAA
994                     ;
995                     ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
996                     ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
997                     ;               OF CAS REG 15 LOW BYTE.
998                     ;
999                     ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
1000                    ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
1001                    ;               17 LOW BYTE SIGN BIT.
1002                    ;
1003                    ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
1004                    ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
1005                    ;               AND/OR ACTUAL DATA.
1006                    ;
1007                    ;       7  -    REQUEST HOST CPU TO PRINT:
1008                    ;               "SUBGROUP NUMBER = LLL"
1009                    ;               WHERE:
1010                    ;               LLL = THE VALUE STORED IN CAS REGISTER 5
1011                    ;                     (THE BYTE COUNT REGISTER 16 BITS).
1012                    ;
1013                    ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
1014                    ; ************************************************************************
```

```
                        ;*************************************************************************
                        ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
                        ;TO CAUSE SOME HOST CPU ACTION.

                        ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
                        ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
                        ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
                        ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
                        ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
                        ;               - HOST RESPONSE CODE 31 OR 33
                        ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
                        ;               - HOST RESPONSE CODE 31 OR 33
                        ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
                        ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
                        ;                       BIT0 = 1 TEST PORT 0
                        ;                       BIT1 = 1 TEST PORT 1
                        ;                       BIT2 = 1 TEST PORT 2
                        ;                       BIT3 = 1 TEST PORT 3
                        ;*************************************************************************
                        ;*************************************************************************
                        ;DATA PATTERN CODES

                        ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
                        ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
                        ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
                        ;               18 BITS OF ALL 1'S
                        ;               18 BITS OF ALL 0'S
                        ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
                        ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
                        ;*************************************************************************
  4300                          .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  BCT1 - BASIC CONFIDENCE TESTS
                                .SBTTL  CPU/CAS INTERFACE PROGRAM
                                ;ID     BCT1-BASIC CONFIDENCE TEST

4300                            S
 (1)                            ; ************************************************************************
                                ;THE VALUE IN THE RAM LOCATION CMD WILL CONTROL THE FUNCTION OF THE
                                ;PROGRAM AS PER THE FOLLOWING TABLE.
                                ;
                                ;               CMD                     FUNCTION
                                ;
                                ;               -0-             READ THE CAS REGISTERS
                                ;                1              WRITE THE CAS REGISTERS
                                ;                2              CAS CONTENTION TEST
                                ;                3              RAM MEMORY TEST
4300                            S
 (1)                            ; ************************************************************************

      4300                              .=      DIAGPG
4300   C3   11   43       10.0 START:    JMP     PROCESS         ;THEN EXECUTE TEST

4303                                    TESTX   @1              ;DUMMY INITIALIZE MACRO
 (1)  4303   3E   01        7.0                          MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4305   CD   03   28  18.0                          CALL    TSET            ;SETUP THE TEST
                                ;%BASIC CONFIDENCE TEST DUMMY TEST HEADER
                                ;&DUMMY MODULE CALLOUT HEADER
      4310                              .=      $4310           ;ADDRESS OF THE "CMD" WORD
4310   00                       CMD:    .BYTE   0               ;ASSUME READ CYCLE

4311   3A   10   43       13.0 PROCESS: LDA     CMD             ;GET ADDR OF COMMAND CONTROL WORD
4314   A7                    4.0         ANA     A               ;RESET CONDITION BITS
4315   CA   5B   43       10.0          JZ      READ            ;IF =0 READ, IF =1 LOAD CAS
4318   FE   01              7.0          CPI     $01             ;LOAD COMMAND?
431A   CA   25   43       10.0          JZ      LOAD            ;YES-GO LOAD CAS REGISTERS
431D   FE   02              7.0          CPI     $02             ;CAS CONTENTION TEST?
431F   CA   73   43       10.0          JZ      CASCTN          ;YES - GO DO THE CONTENTION THING
4322   C3   8C   43       10.0          JMP     MEM             ;NO - ASSUME MEMORY TEST
```

```
1367                                    .SBTTL  WRITE THE CAS REGISTERS ROUTINE
1368
1369  4325  21  82  80      10.0  LOAD:   LXI     H,RO1L!MM       ;GET 1ST CAS ADDRESS
1370  4328  3A  20  44      13.0          LDA     LDAT            ;GET LOW BYTE TO LOAD
1371  432B  47                4.0          MOV     B,A             ;PUT DATA IN REG B
1372
1373  432C  00                4.0  LOADL:  NOP                     ;HACK-TO ALLOW PROPER RETURN FROM CONTENTION INTERRUPT T
1374  432D  70                7.0          MOV     M,B             ;PUT CONTENTS OF "CMD" IN ALL CAS REGISTERS
1375  432E  7F                4.0          MOV     A,A             ;HACK
1376  432F  23                6.0          INX     H               ;UPDATE THE CAS POINTER
1377  4330  23                6.0          INX     H
1378  4331  7D                4.0          MOV     A,L             ;GET LOW BYTE OF POINTER
1379  4332  FE  A0            7.0          CPI     R17L+2          ;DONE? (1ST ADDRESS OF HIGH BYTE AREA)
1380  4334  DA  2C  43       10.0          JC      LOADL           ;NO - DO NEXT CAS ADDRESS
1381
1382  4337  21  83  80      10.0  LOADH:  LXI     H,RO1H!MM
1383  433A  3A  21  44      13.0          LDA     HDAT            ;GET HIGH BYTE DATA WORD
1384  433D  47                4.0          MOV     B,A             ;SAVE THE DATA BYTE IN REG B
1385  433E  00                4.0  LDHILP: NOP                     ;HACK-TO ALLOW PROPER RETURN FORM CONTENTION INTERRUPT T
1386  433F  70                7.0          MOV     M,B             ;STORE IN HIGH BYTE ADDRESS
1387  4340  7F                4.0          MOV     A,A             ;HACK
1388  4341  23                6.0          INX     H               ;POINT TO NEXT ADDRESS
1389  4342  23                6.0          INX     H
1390  4343  7D                4.0          MOV     A,L             ;GET LOW BYTE OF ADDRESS POINTER
1391  4344  FE  A1            7.0          CPI     R17H+2          ;DONE WITH HIGH BYTE ADDRESSES?
1392  4346  DA  3E  43       10.0          JC      LDHILP          ;NO - LOAD NEXT HIGH ADDRESS
1393  4349  3E  FA            7.0  DONE:   MVI     A,@372          ;GET TERMINATE #
1394  434B                                 ROUT    ROOL            ;TELL OUTSIDE WORLD
  (1) 434B  D3  80           10.0                  OUT     ROOL            ;WRITE AC INTO ROOL
  (1) 434D  7F                4.0                  MOV     A,A             ;RETRY LINK
1395  434E                                 DONEX:  RIN     ROOL            ;GET THE FUNCTION CODE REGISTER
  (1) 434E  DB  80           10.0                  IN      ROOL            ;READ ROOL INTO AC
  (1) 4350  7F                4.0                  MOV     A,A             ;RETRY LINK
1396  4351  E6  3F            7.0          ANI     @77             ;DID THE HOST ISSUE
1397  4353  FE  1D            7.0          CPI     @35             ;RESTART?
1398  4355  C2  4E  43       10.0          JNZ     DONEX           ;NO - KEEP ON LOOKING
1399  4358  C3  11  43       10.0          JMP     PROCESS         ;YES - DO IT AGAIN
```

```
1401                                    .SBTTL  READ THE CAS REGISTERS ROUTINE
1402                                    ;HERE TO READ THE CAS REGISTERS FROM THE MICROCOMPUTER SIDE AND SAVE
1403                                    ;THE DATA BYTE READ IN "CSAV" STORAGE AREA
1404
1405  435B  21  82  80      10.0  READ:   LXI     H,R01L!MM       ;SETUP CAS POINTER
1406  435E  11  00  44      10.0          LXI     D,CSAV          ;AND POINTER TO STORAGE
1407
1408  4361  00               4.0  READL:  NOP                     ;HACK-TO ALLOW PROPER RETURN FROM CONTENTION INTERRUPT T
1409  4362                          RMOVAM
 (1)  4362  00          .      4.0                  NOP                     ;FOR INTERRUPT SERVICE SEARCH
 (1)  4363  7E                 7.0                  MOV     A,M             ;READ MEMORY TO AC
 (1)  4364  7F                 4.0                  MOV     A,A             ;RETRY LINK
1410  4365  EB               4.0          XCHG                    ;SWAP D&E WITH H&L
1411  4366  77               7.0          MOV     M,A             ;SAVE THE CAS CONTENTS
1412  4367  EB               4.0          XCHG                    ;RESET D&E AND H&L
1413  4368  23               6.0          INX     H               ;UPDATE CAS POINTER
1414  4369  13               6.0          INX     D               ;AND THE STORAGE POINTER
1415  436A  7D               4.0          MOV     A,L             ;GET LOW BYTE OF CAS POINTER
1416  436B  FE  A0            7.0          CPI     R17H+1          ;DONE YET?
1417  436D  DA  61  43       10.0          JC      READL           ;NO - DO NEXT CAS ADDRESS
1418  4370  C3  49  43       10.0          JMP     DONE            ;WAIT FOR A RESTART COMMAND
1419
```

```
1421                                        .SBTTL  CAS CONTENTION INT. ROUTINE
1422
1423  4373  AF                      4.0   CASCTN: XRA     A               ;CLEAR THE CAS CONTENTION
1424  4374  32  21  4F              13.0           STA     CASCT           ;INTERRUPT
1425  4377  21  80  80              10.0           LXI     H,ROOL!MM               ;LOAD STARTING ADDRESS OF CAS
1426  437A  7E                      7.0   CASLP1: MOV     A,M             ;READ CAS
1427  437B  23                      6.0           INX     H               ;UPDATE THE COUNTER
1428  437C  7D                      4.0           MOV     A,L             ;CHECK FOR DONE
1429  437D  FE  A1                  7.0           CPI     @241            ;
1430  437F  C2  7A  43              10.0          JNZ     CASLP1          ;CONTINUE UNTIL ALL READ
1431  4382  3A  21  4F              13.0          LDA     CASCT           ;GET CAS CONTENTION COUNT
1432  4385  A7                      4.0           ANA     A
1433  4386  CA  73  43              10.0          JZ      CASCTN          ;NO ERROR-CONTINUE
1434  4389  C3  49  43              10.0          JMP     DONE            ;WAIT FOR A RESTART COMMAND
```

```
1436                                        ;MICROCOMPUTER MEMORY EXERCISER ROUTINE
1437
1438  438C  21  B9  43    10.0  MEM:    LXI   H,MDAT          ;POINT TO MEMORY DATA PATTERN TABLE
1439  438F  46                 7.0  MEM1:   MOV   B,M           ;LOAD REG B WITH DATA PATTERN
1440  4390  EB                 4.0          XCHG                ;PUT H&L PATTERN POINTER IN D&E
1441  4391  21  22  44    10.0          LXI   H,ENDPGM        ;POINT TO 1ST FREE ADDRESS TO TEST
1442  4394  70                 7.0  MEM2:   MOV   M,B           ;PUT DATA IN REGISTER "A" IN MEMORY
1443  4395  7E                 7.0          MOV   A,M           ;GET IT FROM MEMORY
1444  4396  B8                 4.0          CMP   B             ;READ/VERIFY DATA IN MEMORY
1445  4397  C2  AC  43    10.0          JNZ   MEMHLT          ;DIE IF ERROR DETECTED
1446  439A  23                 6.0          INX   H             ;UPDATE THE MEMORY POINTER
1447  439B  7C                 4.0          MOV   A,H           ;GET ADDR HI BYTE
1448  439C  FE  50              7.0          CPI   $50           ;DONE WITH MEMORY (16K)??
1449  439E  DA  94  43    10.0          JC    MEM2            ;LOOP AGAIN IF NOT DONE WITH MEMORY PASS
1450  43A1  A7                 4.0  MEMDAT: ANA   A             ;CLEAR THE CONDITION BITS
1451  43A2  78                 4.0          MOV   A,B           ;PUT CURRENT PATTERN IN REG A
1452  43A3  3C                 4.0          INR   A             ;SEE IF REG A GOES TO 0
1453  43A4  CA  49  43    10.0          JZ    DONE            ;PUT PATTERN POINTER IN H&L REG
1454  43A7  13                 6.0          INX   D             ;POINT TO NEXT PATTERN TO USE FROM TABLE
1455  43A8  EB                 4.0          XCHG                ;PUT THE PATTERN POINTER IN H & L REGS
1456  43A9  C3  8F  43    10.0          JMP   MEM1            ;TEST NEXT MEM ADDRESS
1457  43AC  22  CD  43    16.0  MEMHLT: SHLD  BADR            ;STORE THE BAD ADDRESS INFO
1458  43AF  32  CF  43    13.0          STA   BDAT            ;SAVE THE DATA BYTE IN USE WHEN ERROR DETECTED
1459  43B2  78                 4.0          MOV   A,B           ;PUT REAL DATA IN A
1460  43B3  32  D0  43    13.0          STA   ADAT            ;SAVE IT ALSO
1461  43B6  C3  B6  43    10.0          JMP   .               ;STAY HERE FOREVER...OR AT LEAST A WHILE
1462                                    ;HERE IS THE TABLE OF DATA PATTERNS TO USE THROUGH MEMORY
1463  43B9  00                    MDAT:   .BYTE  @000           ;ALL 0'S
1464  43BA  01                            .BYTE  @001           ;FLOATING 1'S
1465  43BB  02                            .BYTE  @002
1466  43BC  04                            .BYTE  @004
1467  43BD  08                            .BYTE  @010
1468  43BE  10                            .BYTE  @020
1469  43BF  20                            .BYTE  @040
1470  43C0  40                            .BYTE  @100
1471  43C1  80                            .BYTE  @200
1472  43C2  7F                            .BYTE  @177           ;FLOATING 0'S
1473  43C3  BF                            .BYTE  @277
1474  43C4  DF                            .BYTE  @337
1475  43C5  EF                            .BYTE  @357
1476  43C6  F7                            .BYTE  @367
1477  43C7  FB                            .BYTE  @373
1478  43C8  FD                            .BYTE  @375
1479  43C9  FE                            .BYTE  @376
1480  43CA  AA                            .BYTE  @252           ;ALTERNATE BITS
1481  43CB  55                            .BYTE  @125
1482  43CC  FF                            .BYTE  @377           ;LAST PATTERN
1483                              .SBTTL  PROGRAM VARIABLES
1484  43CD  00  00                BADR:   .WORD  0             ;ADDRESS OF THE FAILING ADDRESS
1485  43CF  00                    BDAT:   .BYTE  0             ;FAILING DATA
1486  43D0  00                    ADAT:   .BYTE  0             ;EXPECTED DATA
```

```
1488        4400                        .=      $4400           ;START OF CAS REGISTER STORAGE AREA
1489  4400    00                CSAV:   .BYTE   0               ;REG 00 LOW
1490  4401    00                        .BYTE   0               ;REG 00 HIGH
1491  4402    00                        .BYTE   0               ;REG 01 LOW
1492  4403    00                        .BYTE   0               ;REG 01 HIGH
1493  4404    00                        .BYTE   0               ;REG 02 LOW
1494  4405    00                        .BYTE   C               ;REG 02 HIGH
1495  4406    00                        .BYTE   0               ;REG 03 LOW
1496  4407    00                        .BYTE   0               ;REG 03 HIGH
1497                                                ;ETC.
1498                                                ;ETC.
1499
1500        4420                        .=      $4420
1501
1502                              ;"LDAT" FOR LOW BYTE OF CAS REGISTERS 0 - 17
1503                              ;"HDAT" FOR HIGH BYTE OF CAS REGISTERS 0 - 17
1504
1505  4420    00                LDAT:   .BYTE   0               ;STORAGE FOR LOW BYTE DATA
1506  4421    00                HDAT:   .BYTE   0               ;STORAGE FOR HIGH BYTE DATA
1507  4422 0000                 ENDPGM: .END
```

```
A      =%0007          ADAT    43D0          ADATA = 0094          AMTIEP= 0001
AMTIE7= 0002           ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000         BADR    43CD          BADST = 0090
BDAT    43CF           BIT0  = 0001          BIT1  = 0002          BIT15 = 8000
BIT2  = 0004           BIT3  = 0008          BIT4  = 0010          BIT5  = 0020
BIT6  = 0040           BIT7  = 0080          BIT8  = 0100          BIT9  = 0200
BC PBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60          BRKXCT= 4F00
BSAVE   4F9C           BYTCNT= 00D4          BYTEH   4F24          BYTEL   4F23
C      =%0001          CASCT   4F21          CASCTL= 00A0          CASCTN  4373
CASLP1  437A           CASSTA= 00A0          CATTH = 0089          CATTL = 0088
CBUSST= 00A1           CBYTH = 008B          CBYTL = 008A          CDG1H = 0087
CDG1L = 0086           CDG2H = 0093          CDG2L = 0092          CDG3H = 0095
CDG3L = 0094           CDVTH = 008D          CDVTL = 008C          CHPTIE= 0028
CHOTIE= 0020           CH1TIE= 0021          CH2TIE= 0022          CH3TIE= 0023
CH4TIE= 0024           CH5TIE= 0025          CH6TIE= 0026          CH7TIE= 0027
CKLOP = 2815           CLKCTL= 00F0          CLOCK   4F26          CMC0H = 0099
CMC0L = 0098           CMC1H = 009B          CMC1L = 009A          CMC2H = 009D
CMC2L = 009C           CMC3H = 009F          CMC3L = 009E          CMD     4310
CMINH = 0097           CMINL = 0096          CNTCTL= 00D7          CRCWRD= 0018
CSAV    4400           CSAVE   4F9D          CSRLH = 0091          CSRLL = 0090
CTCH  = 0085           CTCL  = 0084          CTSTH = 008F          CTSTL = 008E
CXCTH = 0081           CXCTL = 0080          CXINH = 0083          CXINL = 0082
C.     = 0001          C.AVAI= 0080          C.DP  = 0008          C.DSE = 0010
C.DTU = 0003           C.DVA = 0008          C.FAIL= 00FC          C.FMT = 0070
C.FNCT= 003E           C.GO  = 0001          C.INTC= 00FE          C.MAIN= 0020
C.NSA = 0080           C.RCT = 00FC          C.SER = 0080          C.SHR = 0040
C.SKPC= 000F           C.TAPE= 0040          C.WCS = 0002          D      =%0002
DATACT= 00D0           DBUS    4F28          DBUSCT= 00C0          DBUSST= 00C0
DDRA  = 00D8           DDRAIN= 0010          DDRB  = 00D9          DDRBIN= 0002
DDRC  = 00DA           DDRCIN= 0001          DDRCO = 0088          DDRCTL= 00DB
DIAFLG  4F22           DIAGPG= 4300          DIAGRM= 4F90          DIARD = 000B
DONE    4349           DONEX   434E          DONE1 = 0045          DONINT= 0010
DSAVE   4F9E           DSE   = 0006          D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010           D.LAGC= 0020          D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008           D.WR4 = 0080          E      =%0003         ECCBAD= 0042
ECCCOR= 0019           ECCOK = 0041          ECCSTA= 001A          ECCTST= 000E
EDATA = 0095           ENDPGM  4422          EOTCLR= 0003          ERFLG   4F93
ERLP  = 2809           ERLPA = 280F          ERLPB = 2812          ERLPE = 280C
ERNUM   4F90           ERRCNT= 00D6          ESAVE   4F9F          E.ACRC= 0010
E.AMT = 0020           E.CDP = 0080          E.CRC = 0080          E.PNTR= 0008
E.RPE = 0040           E.STEC= 0001          E.TTEC= 0002          E.UNC = 0004
FIFORD= 006A           FORMAT  4F25          FWDTST= 0061          GCRID = 0089
GCRSET= 0002           GOODTM= 0092          H      =%0004         HDAT    4421
HLSAVE  4FA0           IE    = 0008          INTSTA= 00E0          ITERA   4F9A
I.PWR = 0020           I.RMPE= 0040          I5.5  = 001D          I6.5  = 0020
I7.5  = 0040           KCALL = 005F          KCLR  = 007B          KDEP  = 003F
KENAB = 0078           KEXAM = 003E          KEYBRD= 00C8          KEY1  = 0078
KEY10 = 006D           KEY11 = 006E          KEY12 = 006F          KEY13 = 005C
KEY14 = 005D           KEY15 = 005E          KEY16 = 005F          KEY17 = 003C
KEY18 = 003D           KEY19 = 003E          KEY2  = 0079          KEY20 = 003F
KEY3  = 007A           KEY4  = 007B          KEY5  = 0074          KEY6  = 0075
KEY7  = 0076           KEY8  = 0077          KEY9  = 006C          KINTA = 006F
KLDAD = 003D           KN0   = 003C          KN1   = 005C          KN2   = 005D
KN3   = 005E           KN4   = 006C          KN5   = 006D          KN6   = 006E
```

```
KN7    = 0074   KN8    = 0075   KN9    = 0076   KU2    = 0079
KU3    = 007A   KU8    = 0077   L      =%0005   LBLANK= 000F
LCE    = 000B   LCH    = 000C   LCL    = 000D   LCP    = 000E
LC0    = 0000   LC1    = 0001   LC2    = 0002   LC3    = 0003
LC4    = 0004   LC5    = 0005   LC6    = 0006   LC7    = 0007
LC8    = 0008   LC9    = 0009   LDAT     4420   LDHILP  433E
LDLEDA= 00CA    LDLEDB= 00CB    LDLEDC= 00CC    LDLEDD= 00CD
LDLEDE= 00CE    LDLEDF= 00CF    LKDIAG= 2800    LKKBD = 004C
LKKEY  = 0049   LKLWMG= 0058    LKLWMP= 0055    LKLWPG= 0052
LKLWPP= 004F    LKMOD7= 0046    LKOPR  = 0046   LOAD     4325
LOADH    4337   LOADL    432C   LPFLG    4F94   LPNUM    4F92
M      =%0006   MBSEL  = 00E0   MB.A   = 0008   MB.B   = 0004
MDAT     43B9   MEM      438C   MEMDAT   43A1   MEMHLT   43AC
MEMTOP= 4FFF    MEM1     438F   MEM2     4394   MINUS  = 000A
MM     = 8000   MSE    = 0008   MSGN   = 0000   MTACLR= 0000
MT.ARA= 0020    MT.CPE= 0080    MT.DSE= 0001    MT.FWD= 0040
MT.INH= 0008    MT.LWR= 0004    MT.MOT= 0002    MT.NWT= 0080
MT.PEC= 0040    MT.PSB= 0004    MT.PS0= 0001    MT.PS1= 0002
MT.REV= 0020    MT.WRT= 0010    MT.Z   = 0008   M.ATA  = 0080
M.CAPE= 0020    M.CONT= 0080    M.DEM  = 0020   M.EBL  = 0010
M.EXC  = 0008   M.FAIL= 0008    M.ILP  = 0010   M.INIT= 0010
M.OCC  = 0020   M.ONLI= 0001    M.PE   = 0040   M.PORT= 0060
M.RDEN= 0002    M.RDPE= 0008    M.RUN  = 0004   M.SCLK= 0001
M.TRA  = 0040   M.UNIT= 0007    M.WCLK= 0040    M.WCLN= 0080
M.WREN= 0080    M5.5   = 0001    M6.5   = 0002   M7.5    = 0004
NOTCAP= 0088    OKAY   = 00FF    OPRRAM= 4300    OPSTRT= 0058
OPVER  = 0040   PADCNT= 00D5    PADCRC= 0080    PDIAG  = 0048
PEID   = 008A   PENAB  = 004C   PESET  = 0001   PL     = 00B1
PRDD   = 004C   PRENF  = 009C   PROCES   4311   PS     = 00B2
PSTAT  = 0048   PSW    =%0009   P.AMTP= 0001   P.BCTC= 0040
P.CMDP= 0020    P.INTE= 0080    P.LCS  = 0040   P.LWR  = 0020
P.RDP  = 0002   P.RPEN= 0004    P.RPS1= 0002    P.RP0E= 0020
P.RP1E= 0010    P.RP2E= 0020    P.RP3E= 0010    P.SING= 0080
P.STAT= 0002    P.STPE= 0080    P.TACH= 0008    P.TUPR= 0010
P.WCSP= 0004    P.WDS  = 0040   P.WFLP= 0001    P.WPEN= 0010
P.WP0E= 0008    P.WP1E= 0004    P.WP2E= 0008    P.WP3E= 0004
P.5VOK= 0002    QUE    = 281B   QUEM   = 281E   RAMT   = 0010
RARA   = 0006   RARAI  = 0004   RCHBD0= 0048   RCHBD1= 0047
RCHOK  = 0046   RCHTST= 000C    RCLRT  = 000D   RCMD   = 000B
RCMLP  = 0003   RCONT  = 0080   RDATA  = 0017   RDCLK  = 0010
RDON   = 0011   READ     435B   READG  = 0007   READL    4361
REND   = 0014   REQST  = 2806   RESCHR= 00D1   REVTST= 0064
REWIND= 0004    RFIFOL= 0008    RGCLK  = 0002   RGCRI  = 0003
RIBG   = 0001   RILL   = 0012   RINST  = 000C   RMCTST= 0008
RMK2   = 0013   RNOP   = 0000   RPATH  = 0001   RPBAD  = 0044
RPCHI  = 0001   RPCLK  = 0003   RPCTL  = 0009   RPEI   = 0002
RPFAIL= 0000    RPIT   = 009D   RPF2   = 009E   RPOK   = 0043
RPOSTN= 0016    RPSTA  = 0015   RRCMT  = 00CA   RSTAT  = 0002
RTIEB  = 000A   RTIER  = 0030   RTM    = 0005   RUNKI  = 0009
RUPTST= 005E    RWDUNL= 0005    R.AMT  = 0001   R.BOP  = 0008
R.DATA= 0040    R.DON  = 0002   R.DRDY= 0010    R.END  = 0010
R.ILL  = 0004   R.JVOK= 0004    R.MK2  = 0008   R.PLOD= 0008
R.PL00= 0010    R.PL01= 0020    R.POST= 0020    R.STNM= 0002
R.STOP= 0004    R.STPC= 0001    R.TBJN= 0080    R.TSTD= 0040
```

```
R.VOK = 0080    R00H  = 0081    R00L  = 0080    R01H  = 0083
R01L  = 0082    R02H  = 0085    R02L  = 0084    R03H  = 0087
R03L  = 0086    R04H  = 0089    R04L  = 0088    R05H  = 008B
R05L  = 008A    R06H  = 008D    R06L  = 008C    R07H  = 008F
R07L  = 008E    R10H  = 0091    R10L  = 0090    R11H  = 0093
R11L  = 0092    R12H  = 0095    R12L  = 0094    R13H  = 0097
R13L  = 0096    R14H  = 0099    R14L  = 0098    R15H  = 009B
R15L  = 009A    R16H  = 009D    R16L  = 009C    R17H  = 009F
R17L  = 009E    R7.5  = 0010    SELCLR= 0000    SETATA= 00A1
SID   = 0080    SOD   = 0080    SOE   = 0040    SP    =%0008
SSCLK = 0040    SSTEP = 0005    STACK = 4FFF    START   4300
STATRM= 4F20    STPCT   4F20    STRSP = 5000    TADR00= 0080
TADR01= 0081    TADR02= 0082    TADR03= 0083    TADR04= 0084
TADR05= 0085    TADR06= 0086    TADR07= 0087    TADR10= 0088
TADR11= 0089    TADR12= 008A    TADR13= 008B    TAMT  = 0044
TASEL = 0080    TCMD  = 0040    TC.FWD= 0040    TC.INH= 0008
TC.LWR= 0004    TC.REV= 0020    TC.WRT= 0010    TEMP    4F99
TMF   = 0099    TMRDY = 0040    TRKENA= 00D2    TSET  = 2803
TSTEND= 2818    TSTS  = 0040    TUSEL0= 00D1    TUSEL1= 00D2
TU78  = 0010    T.ATH0= 0001    T.ATH1= 0002    T.BOT = 0004
T.EOT = 0002    T.FPT = 0001    T.NTHR= 0004    T.ONL = 0020
T.PES = 0008    T.PSBJ= 0020    T.PSOJ= 0008    T.PS1J= 0010
T.RDY = 0080    T.RDYO= 0040    T.RWD = 0010    T.SCLK= 0002
UIBG  = 00A1    VALFC   4F98    VALTB   4F95    VELTST= 005B
WDR.P = 0010    WMCCTL= 00D3    WMCERR= 00DA    WMCSTA= 00D0
WRTCLK= 0000    WRTDAT= 00D3    W.ACRC= 0004    W.CRC = 0008
W.DIAG= 0002    W.DONN= 0040    W ECC = 0010    W.ENAB= 0080
W.ERR = 0020    W.FMT = 0070    W.GCR = 0010    W.LEFT= 0004
W.ONES= 0020    W.RESI= 0002    W.REV = 0004    W.ROME= 0010
W.RST = 0001    W.SKIP= 000F    W.WRIT= 0008    W.XFER= 0020
X     =%000A    X.DONN= 0080    X.ENAB= 0040    X.PEPE= 0002
X.ROME= 0001    X.WCLK= 0001    Y     =%000B    .     = 4422


 ERRORS DETECTED:  0

 *BCT1.A78/PTP,BCT1=NLIST,PARAM,MACRO,LIST,BCT1
 RUN-TIME: 3 3 0 SECONDS
 CORE USED:  10K
```

3	"MEMORY MAP" EXTENDED ADDRESS BIT
19	READ MICRO CONTROLLER CONTROL & STATUS WORDS
139	TRANSPORT/PORT COMMAND + STATUS WORDS
380	COMMON ADDRESS SPACE REGISTERS
444	MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485	MASSBUS CONTROL WORD DEFINITIONS
557	WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709	KEYBOARD/DISPLAY CONTROL + STATUS WORD
797	MISCELLANEOUS VARIABLE DEFINITIONS
801	RAM ADDRESS DEFINITIONS
819	LINKAGE TABLE ADDRESSES
842	BIT DEFINITIONS
856	RIM AND SIM INSTRUCTION BIT DEFINITIONS
877	INTERRUPTABLE INSTRUCTION MACROS
924	DIAGNOSTIC VARIABLES
945	COMMON VARIABLE STORAGE
959	MACRO CALL DEFINITIONS
1047	TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101	8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1333	TEST 1 - WMC SELF DIAGNOSTICS
1455	TEST 2 - WMC BASIC SANITY CHECK
1556	TEST 3 - WMC RE-INITIALIZE TEST
1620	TEST 4 - BYTE COUNTER TEST
1744	TEST 5 - WMC PAD COUNTER TEST
1853	TEST 6 - WMC ECODE COUNTER TEST
1962	TEST 7 - WMC 'ECODE' TEST FOR ILLEGAL FORMAT CODE #7
2103	TEST 10 - WMC SYSTEM CLOCK TEST

```
961                     ;*************************************************************************
962                     ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
963                     ;ERROR MACRO CALLS
964                     ;
965                     ;       1  -    REQUEST HOST CPU TO PRINT:
966                     ;               "BYTE/SCLK COUNT NUMBER = LLL"
967                     ;               WHERE:
968                     ;               LLL = THE VALUE STORED IN CAS REGISTER 5
969                     ;                     (THE BYTE COUNT REGISTER 16 BITS).
970                     ;
971                     ;       2  -    REQUEST HOST CPU TO PRINT:
972                     ;               DATA FORMAT = MM
973                     ;               SKIP COUNT  = NN
974                     ;               WHERE:
975                     ;               MM = DATA FORMAT FROM CAS REGISTER 2
976                     ;               NN = SKIP COUNT FROM CAS REGISTER 2
977                     ;
978                     ;       3  -    REQUEST HOST CPU TO PRINT:
979                     ;               BYTE-SCLK COUNT = LLL
980                     ;               DATA FORMAT = MM
981                     ;               SKIP COUNT  = NN
982                     ;               WHERE:
983                     ;               LLL = AS ABOVE
984                     ;               MM  = AS ABOVE
985                     ;               NN  = AS ABOVE
986                     ;
987                     ;       4  -    REQUEST HOST TO PRINT:
988                     ;               TRANSITION COUNT = LLL
989                     ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
990                     ;
991                     ;       5  -    REQUEST HOST CPU TO PRINT:
992                     ;               EXPECTED 18 BITS =      E EEEFEE
993                     ;               ACTUAL 18 BITS =        A AAAAAA
994                     ;
995                     ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
996                     ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
997                     ;               OF CAS REG 15 LOW BYTE.
998                     ;
999                     ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
1000                    ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
1001                    ;               17 LOW BYTE SIGN BIT.
1002                    ;
1003                    ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
1004                    ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
1005                    ;               AND/OR ACTUAL DATA.
1006                    ;
1007                    ;       7  -    REQUEST HOST CPU TO PRINT:
1008                    ;               "SUBGROUP NUMBER = LLL"
1009                    ;               WHERE:
1010                    ;               LLL = THE VALUE STORED IN CAS REGISTER 5
1011                    ;                     (THE BYTE COUNT REGISTER 16 BITS).
1012                    ;
1013                    ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
1014                    ;*************************************************************************
```

```
1016                    ; ************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ; ************************************************************************
1035                    ; ************************************************************************
1036                    ;DATA PATTERN CODES
1037
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ; ************************************************************************
1329    4300                    .-      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  WMC1 - WRITE MICROCONTROLLER PART #1
                                .SBTTL  TEST 1 - WMC SELF DIAGNOSTICS
                                ;ID     WMC1-WRITE MICRO CONTROLLER PART #1

                                ;THIS SERIES OF WRITE MICROCONTROLLER TESTS ARE DESIGNED TO TEST
                                ;THE WMC INTERFACE LOGIC. THE "BYTE COUNTER", "PAD COUNTER", AND
                                ;"ECODE COUNTER" LOGIC IS TESTED AS WELL AS THE GROSS WMC 'READY
                                ;CHECK.
                                ;
                                ;THESE TESTS HAVE TO BE PERFORMED BEFORE ANY FORMAT CONTROL OR
                                ;SKIP COUNT TESTS CAN BE DONE.
                                ;THIS TEST IS TO CHECK THE WMC SELF-DIAGNOSIS CAPABILITIES

4300                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*WMC SELF DIAGNOSTICS
4300                            SM
 (1)                            ; ***************************************************************************
 (1)                            ;*MODULE(S) UNDER TEST
 (1)                            ;*--------------------
                                ;*M8959, M8957
4300                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  CLEAR THE READ PATH MACHINE
                                ;*  SET DDR CONTROL TO 'IN'
                                ;*  SET CLOCK CONTROL TO 'NORMAL'
                                ;*  ISSUE A WMC RESTART
                                ;*  IF WMC NOT-DONE IS SET
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT ERROR-SELF-DIAGNOSTIC FAILURE
                                ;*  ENDIF
                                ;*  IF CBUSSTA HAS WMC 'LEFT' CLEAR
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT WMC RESTART FAILED TO CLEAR 'LEFT' MODE
                                ;*  ENDIF
                                ;*  SMALL DELAY FOR WMC SELF-DIAGNOSTIC TO FINISH
                                ;*  IF WMCERR STATUS INDICATES NO ERROR BITS
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT SELF-DIAGNOSTICS DETECT ERROR
                                ;*  ENDIF
                                ;*ENDTST
4300                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*WMC1 MICRO TEST 01
                                ;*WMC1 MICRO ERROR 01
                                ;*WMC1-WMC SELF-DIAGNOSTICS
                                ;*M8959, M8957
```

```
                                        ;*WMC FAILED TO SET 'WMC NOT DONE' TO INDICATE SELF-DIAG. RUNNING
                                        ;*
                                        ;*WMC1 MICRO TEST 01
                                        ;*WMC1 MICRO ERROR 02
                                        ;*WMC1-WMC SELF-DIAGNOSTICS
                                        ;*M8959, M8957
                                        ;*WMC RESTART FAILED TO CLEAR THE WMC 'LEFT' BIT IN CBUSSTA WORD (241)
                                        ;*ACTUAL = NNNN
                                        ;*
                                        ;*WMC1 MICRO TEST 01
                                        ;*WMC1 MICRO ERROR 03
                                        ;*WMC1-WMC SELF-DIAGNOSTICS
                                        ;*M8959, M8957
                                        ;*WMC SELF-TESTS SUSPECTED OF FINDING A FAILURE! ACTUAL WORD IS THE
                                        ;*WMC ERROR STATUS BYTE FROM WMCERR (332)
                                        ;*ACTUAL = NNNN
4300                                    $
 (1)                                    ; ************************************************************************

                                        TEST1:  TESTX   @1
 (1) 4300  3E  01            7.0                 MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1) 4302  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
                                        ;%WMC1-WMC SELF-DIAGNOSTICS
                                        ;&M8959, M8957
4305  3E  10                 7.0                 MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
4307  D3  F0                10.0                 OUT     CLKCTL
4309  3E  10                 7.0                 MVI     A,W.GCR         ;SET GCR MODE
430B  D3  D3                10.0                 OUT     WMCCTL
430D  3E  20                 7.0                 MVI     A,R.PLO1        ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
430F  D3  09                10.0                 OUT     RPCTL
4311  AF                     4.0                 XRA     A
4312  D3  0A                10.0                 OUT     RTIEB           ;CLEAR THE TIE BUS
4314  3E  0U                 7.0                 MVI     A,RCLRT         ;ISSUE CLEAR ALL COMMAND
4316  D3  0B                10.0                 OUT     RCMD
4318  00                     4.0                 NOP                     ;WAIT
4319  00                     4.0                 NOP                     ;WAIT
431A  00                     4.0                 NOP
431B  00                     4.0                 NOP
431C  00                     4.0                 NOP
431D  3E  21                 7.0                 MVI     A,R.PLO1!R.STPC ;STOP THE READ PATH
431F  D3  09                10.0                 OUT     RPCTL

                                        ;SET DDR CONTROL SO CORRECT ERROR INFO CAN BE FOUND

4321  3E  89                 7.0                 MVI     A,@211          ;SET DDRC TO 'IN'
4323  D3  DB                10.0                 OUT     DDRCTL          ;LOAD THE DDR CONTROL WORD

                                        ;RESTART WMC

4325  AF                     4.0  TST1L:    XRA     A                   ;SET THE SYSTEM CLOCK TO 'NORMAL'
4326  D3  F0                10.0                 OUT     CLKCTL          ;WRITE THE CLOCK CONTROL WORD
4328  3E  01                 7.0                 MVI     A,W.RST         ;THEN RESTART THE WMC
432A  D3  D3                10.0                 OUT     WMCCTL
432C  00                     4.0                 NOP                     ;SMALL DELAY
```

```
1425  432D  AF                    4.0          XRA     A                ;FINISH THE RESTART
1426  432E  D3  D3                10.0          OUT     WMCCTL           ;CLEAR THE RESTART BIT
1427  4330  DB  D0                10.0          IN      WMCSTA           ;GET WMC STATUS WORD
1428  4332  E6  40                 7.0          ANI     W.DONN           ;SAVE ONLY 'WMC NOT DONE' BIT
1429  4334  C2  3C  43           10.0          JNZ     TST1AC
1430  4337                                      ERR     TST1L,TST1AC
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4337  CD  09  28           18.0                  CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)       0001                               MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  433A  01                              .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  433B  00                              .BYTE
 (1)  433C  CD  15  28           18.0          TST1AC::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  433F  DA  25  43           10.0                  JC      TST1L            ;LOOP ADDRESS IF LOOP SPECIFIED
1431                                      ;>WMC FAILED TO SET 'WMC NOT DONE' TO INDICATE SELF-DIAG. RUNNING
1432                                      ;THIS SECTION CHECKS THAT THE RESTART COMMAND RESET THE WMC 'LEFT' BIT
1433                                      ;IN THE 'CBUSSTA' WORD (241(8)).
1434
1435  4342  DB  A1                 10.0  TST1B:  IN      CBUSSTA          ;GET THE STATUS DATA
1436  4344                                      ROUT    ADATA
 (1)  4344  D3  94                 10.0                  OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  4346  7F                      4.0                  MOV     A,A              ;RETRY LINK
1437  4347  E6  04                 7.0          ANI     W.LEFT           ;SAVE ONLY THE 'LEFT' BIT
1438  4349  CA  51  43           10.0          JZ      TST1BC           ;OK IF =0
1439  434C                                      ERRA    TST1L,TST1BC
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  434C  CD  0F  28           18.0                  CALL    ERLPA            ;PROCESS ERROR - DO 2.3
 (1)       0002                               MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  434F  02                              .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  4350  00                              .BYTE                    ;PRINT ROUTINE NUMBER
 (1)  4351  CD  15  28           18.0          TST1BC::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4354  DA  25  43           10.0                  JC      TST1L            ;LOOP ADDRESS IF LOOP SPECIFIED
1440                                      ;>WMC RESTART FAILED TO CLEAR THE WMC 'LEFT' BIT IN CBUSSTA WORD (241)
1441
1442  4357  3E  28                 7.0  TST1C:  MVI     A,40             ;TEST FOR ERROR DETECTED BY SELF TESTS
1443  4359  3D                      4.0  TST1W:  DCR     A                ;DOWNCOUNT THE WAIT COUNTER
1444  435A  C2  59  43           10.0          JNZ     TST1W            ;STAY HERE TILL =0
1445  435D  DB  DA                10.0          IN      WMCERR           ;GET THE WMC ERROR STATUS BYTE
1446  435F                                      ROUT    ADATA            ;SAVE FOR ERROR REPORT
 (1)  435F  D3  94                 10.0                  OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  4361  7F                      4.0                  MOV     A,A              ;RETRY LINK
1447  4362  E6  30                 7.0          ANI     W.ERR!W.ROME     ;SAVE ONLY THE ERROR BITS
1448  4364  CA  6C  43           10.0          JZ      TST1CC           ;JUMP OVER IF NO ERROR DETECTED
```

```
1450  4367                                          ERRA    TST1L,TST1CC
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4367  CD  0F  28        18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0003                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  436A  03                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  436B  00                                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  436C  CD  15  28        18.0                  TST1CC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  436F  DA  25  43        10.0                          JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1451                                        ;>WMC SELF-TESTS SUSPECTED OF FINDING A FAILURE. ACTUAL WORD IS THE
1452                                        ;>WMC ERROR STATUS BYTE FROM WMCERR (332)
1453  4372                                    ENDTST  TST1L
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4372                                  REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4372  CD  06  28        18.0                  CALL    REQST
 (2)  4375  00                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  4376  00  00                                  .WORD                   ;SYSTEM "" COUNT
 (2)  4378  00  00                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  437A  00                                      .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  437B  07                                      .BYTE   7               ;REQUEST CODE
 (1)  437C  3A  9A  4F        13.0          LDA     ITERA                   ;GET ITERATION COUNT
 (1)  437F  3D                 4.0          DCR     A                       ;DOWNCOUNT
 (1)  4380  32  9A  4F        13.0          STA     ITERA                   ;SAVE COUNT
 (1)  4383  F2  25  43        10.0          JP      TST1L                   ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 2 - WMC BASIC SANITY CHECK
4386                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*WMC BASIC SANITY CHECK
4386                            SM
                                ; ***************************************************************************
                                ;*MODULE(S) UNDER TEST
                                ;*--------------------
                                ;*M8959
4386                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A BASIC SANITY CHECK OF THE WRITE MICRO CONTROLLER
                                ;*BY SETTING BOTH THE 'RESTART' AND 'ENABLE WRITE MICRO CONTROLLER' BITS
                                ;*IN THE WMC CONTROL REGISTER.  THIS BIT COMBINATION SHOULD CAUSE THE
                                ;*WRITE MICRO CONTROLLER STATUS BIT 'XMC NOT DONE' TO RESET AND STAY
                                ;*RESET.
4386                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  CLEAR DATACTL BITS SO WMC WON'T HAVE INIT ERRORS
                                ;*  SET WMC RESTART + ENABLE BITS
                                ;*  CLEAR WMC RESTART BIT FINISH RESTART CYCLE
                                ;*  SMALL DELAY
                                ;*  IF WMCSTA HAS NOT-DONE SET
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT WMC ROM PGM FAILURE
                                ;*  ENDIF
                                ;*  CLEAR WMC ENABLE BIT IN WMCCTL-CAUSE A WMC CLEAR FUNCTION
                                ;*  IF WMCERR INDICATES NO ERROR
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT WMC RESTART FAILURE
                                ;*  ENDIF
                                ;*  IF INSTA X.ROME BIT=0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT WMC RESTART FAILED TO CLEAR TRANSLATION
                                ;*  ENDIF
                                ;*  IF WMCSTA NOT-DONE BIT=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT WMC FAILED TO FINISH
                                ;*  ENDIF
                                ;*ENDTST
4386                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*WMC1 MICRO TEST 02
                                ;*WMC1 MICRO ERROR 04
                                ;*WMC1-WMC BASIC SANITY CHECK
```

```
1494                                    ;*M8959
1495                                    ;*WMC STATUS BIT "WMC NOT DONE" CLEAR AFTER ENABLE/RESTART ISSUED
1496                                    ;*
1497                                    ;*WMC1 MICRO TEST 02
1498                                    ;*WMC1 MICRO ERROR 05
1499                                    ;*WMC1-WMC BASIC SANITY CHECK
1500                                    *M8959
1501                                    ;*WMC RESTART DID NOT CLEAR THE WMC ERRORS BYTE
1502                                    ;*ACTUAL = NNNN
1503                                    ;*
1504                                    ;*WMC1 MICRO TEST 02
1505                                    ;*WMC1 MICRO ERROR 06
1506                                    ;*WMC1-WMC BASIC SANITY CHECK
1507                                    ;*M8959
1508                                    ;*WMC RESTART DiD NOT CLEAR THE XMC ROM PARITY ERRORS BIT
1509                                    ;*ACTUAL = NNNN
1510                                    ;*
1511                                    ;*WMC1 MICRO TEST 02
1512                                    ;*WMC1 MICRO ERROR 07
1513                                    ;*WMC1-WMC BASIC SANITY CHECK
1514                                    ;*M8959
1515                                    ;*WMC STATUS BIT 'WMC NOT DONE' NOT SET AFTER 'ENABLE'+'RESTART' DROPPED
1516  4386                              S
 (1)                                    ; *****************************************************************************
1517  4386                              TEST2:  TESTX   @2
 (1)  4386  3E  02            7.0                 MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4388  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
1518                                    ;%WMC1-WMC BASIC SANITY CHECK
1519                                    ;&M8959
1520  438B  AF                  4.0   TST2L:  XRA     A                 ;CLEAR THE FORMAT BITS
1521  438C  D3  D0              10 0          OUT     DATACTL           ;SO WMC WON'T HAVE ERROR WHEN STARTED
1522  438E  3E  81               7.0          MVI     A,W.RST.W.ENAB    ;SET 'EN WMC' AND 'RESTART'
1523  4390  D3  D3              10.0          OUT     WMCCTL            ;
1524  4392  3E  80               7.0          MVI     A,W.ENAB          ;CLEAR THE RESTART BIT
1525  4394  D3  D3              10.0          OUT     WMCCTL
1526  4396  00                   4.0          NOP                       ;DELAY
1527  4397  00                   4.0          NOP                       ;DELAY
1528  4398  DB  D0              10.0          IN      WMCSTA            ;GET WMC STATUS BYTE
1529  439A  E6  40               7.0          ANI     W.DONN            ;CHECK FOR 'WMC NOT DONE'
1530  439C  C2  A4  43          10.0          JNZ     TST2AC            ;CONTINUE IF SET
1531  439F                                    ERR     TST2L,TST2AC
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  439F  CD  09  28       18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0004                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43A2  04                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43A3  00                                           .BYTE
 (1)  43A4  CD  15  28       18.0          TST2AC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43A7  DA  8B  43       10.0                        JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
1532                                    ;>WMC STATUS BIT "WMC NOT DONE" CLEAR AFTER ENABLE/RESTART ISSUED
1533  43AA  AF                  4.0          XRA     A                 ;NOW DROP ENABLE TO CAUSE A CLEAR FUNCTION
1534  43AB  D3  D3              10.0          OUT     WMCCTL            ;IN THE WMC ROM PROGRAM
1535  43AD  DB  DA              10.0          IN      WMCERR            ;GET THE WMC ERRORS BYTE
1536  43AF  E6  78               7.0          ANI     M.PE!M.RDPE!W.ROME!W.ERR
1537  43B1  CA  BC  43          10.0          JZ      TST2BC            ;JUMP IF NO ERROR BIT UP
```

```
1538  43B4                                        ROUT    ADATA
 (1)  43B4  D3  94          10.0                          OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43B6  7F               4.0                          MOV     A,A             ;RETRY LINK
1539  43B7                                        ERRA    TST2L,TST2BC
 (1)                                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  43B7  CD  0F  28      18.0                                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0005                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43BA  05                                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43BB  00                                                  .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  43BC  CD  15  28      18.0                          TST2BC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43BF  DA  8B  43      10.0                                  JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
1540                                        ;>WMC RESTART DID NOT CLEAR THE WMC ERRORS BYTE
1541  43C2  DB  E0          10.0                  IN      INTSTA          ;GET THE INTERRUPT STATUS WORD
1542  43C4  E6  01           7.0                  ANI     X.ROME
1543  43C6                                        ROUT    ADATA
 (1)  43C6  D3  94          10.0                          OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43C8  7F               4.0                          MOV     A,A             ;RETRY LINK
1544  43C9  CA  D1  43      10.0                  JZ      TST2CC
1545  43CC                                        ERRA    TST2L,TST2CC
 (1)                                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  43CC  CD  0F  28      18.0                                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0006                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43CF  06                                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43D0  00                                                  .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  43D1  CD  15  28      18.0                          TST2CC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43D4  DA  8B  43      10.0                                  JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
1546                                        ;>WMC RESTART DID NOT CLEAR THE XMC ROM PARITY ERRORS BIT
1547
1548  43D7  00                4.0  TST2D:  NOP
1549  43D8  DB  D0           10.0          IN      WMCSTA          ;GET WMC STATUS WORD
1550  43DA  E6  40            7.0          ANI     W.DONN          ;SAVE ONLY 'WMC NOT DONE' BIT
1551  43DC  C2  E4  43       10.0          JNZ     TST2DC
1552  43DF                                 ERR     TST2L,TST2DC
 (1)                                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  43DF  CD  09  28      18.0                                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43E2  07                                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43E3  00                                                  .BYTE
 (1)  43E4  CD  15  28      18.0                          TST2DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43E7  DA  8B  43      10.0                                  JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
1553                                        ;>WMC STATUS BIT 'WMC NOT DONE' NOT SET AFTER 'ENABLE' + 'RESTART' DROPPED
1554  43EA                                        ENDTST  TST2L
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43EA                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43EA  CD  06  28      18.0                                  CALL    REQST
 (2)  43ED  00                                                  .BYTE                   ;DATA PATTERN NUMBER
 (2)  43EE  00  00                                              .WORD                   ;SYSTEM "" COUNT
 (2)  43F0  00  00                                              .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43F2  00                                                  .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  43F3  07                                                  .BYTE   7               ;REQUEST CODE
 (1)  43F4  3A  9A  4F      13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  43F7  3D               4.0          DCR     A               ;DOWNCOUNT
 (1)  43F8  32  9A  4F      13.0          STA     ITERA           ;SAVE COUNT
 (1)  43FB  F2  8B  43      10.0          JP      TST2L           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 3 - WMC RE-INITIALIZE TEST
1557  43FE                      ST
 (1)                            ;*****************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1558                            ;*WMC RE-INITIALIZE TEST
1559  43FE                      SM
 (1)                            ;*****************************************************************************
 (1)                            ;*MODULE(S) UNDER TEST
 (1)                            ;*--------------------
1560                            ;*M8959
1561  43FE                      SD
 (1)                            ;*****************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1562                            ;*THIS TEST IS CHECKING THAT THE WMC STATUS WORD (100320)
1563                            ;*DOES NOT INDICATE "WMC NOT READY" STATUS AFTER THE WMC CONTROL
1564                            ;*WORD (100323) IS LOADED WITH A RESTART COMMAND - A RESTART WILL
1565                            ;*CAUSE THE WMC ROM PROGRAM TO RE-INITIALIZE TO A KNOWN STATE.
1566  43FE                      SP
 (1)                            ;*****************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1567                            ;*BGNTST
1568                            ;*  SET WMCCTL RESTART BIT
1569                            ;*  IF WMCERR INDICATES ERROR
1570                            ;*  :  THEN-REPORT ERROR AFTER WMC RESTART SET
1571                            ;*  :  ELSE-CONTINUE
1572                            ;*  ENDIF
1573                            ;*  FINISH WMC RESTART CYCL
1574                            ;*  IF WMCERR INDICATES ERROR
1575                            ;*  :  THEN-REPORT ERROR AFTER WMC RESTART CLEARED
1576                            ;*  :  ELSE-CONTINUE
1577                            ;*  ENDIF
1578                            ;*ENDTST
1579  43FE                      SE
 (1)                            ;*****************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1580                            ;*WMC1 MICRO TEST 03
1581                            ;*WMC1 MICRO ERROR 10
1582                            ;*WMC1-WMC RE-INITIALIZE TEST
1583                            ;*M8959
1584                            ;*WMC ERROR STATUS WORD INDICATES 'ERROR' AFTER WMC 'RESTART' SET
1585                            ;*
1586                            ;*WMC1 MICRO TEST 03
1587                            ;*WMC1 MICRO ERROR 11
1588                            ;*WMC1-WMC RE-INITIALIZE TEST
1589                            ;*M8959
1590                            ;*WMC ERROR STATUS WORD INDICATES 'ERROR' AFTER WMC 'RESTART' CLEARED
1591  43FE                      S
 (1)                            ;*****************************************************************************
```

```
1593  43FE                              TEST3:  TESTX   @3
 (1)  43FE  3E  03            7.0                        MVI     A,@3            ;DEFINE THE TEST NUMBER
 (1)  4400  CD  03  28       18.0                        CALL    TSET            ;SETUP THE TEST
1594                                    ;%WMC1-WMC RE-INITIALIZE TEST
1595                                    ;&M8959
1596
1597                                    ;ISSUE A 'RESTART' COMMAND TO WMC CONTROL WORD
1598
1599  4403  3E  81            7.0  TST3L:  MVI     A,W.ENAB!W.RST          ;GET THE RESTART BIT
1600  4405  D3  D3           10.0           OUT     WMCCTL          ;ISSUE THE COMMAND
1601
1602                                    ;WMC ROM ERROR TEST
1603
1604  4407  DB  DA           10.0  TST3A:  IN      WMCERR          ;GET THE WMC ERROR WORD
1605  4409  E6  30            7.0           ANI     W.ROME!W.ERR    ;SAVE 'WMC ROM ERR' & 'ERROR'
1606  440B  CA  13  44        10.0           JZ      TST3AC          ;JUMP OVER IF =0 - NO ERROR SET
1607  440E                                  ERR     TST3L,TST3AC
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  440E  CD  09  28        18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0008                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4411  08                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4412  00                                            .BYTE
 (1)  4413  CD  15  28        18.0                TST3AC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4416  DA  03  44        10.0                        JC      TST3L           ;LOOP ADDRESS IF LOOP SPECIFIED
1608                                    ;>WMC ERROR STATUS WORD INDICATES 'ERROR' AFTER WMC 'RESTART' SET
1609  4419  3E  80            7.0           MVI     A,W.ENAB        ;FINISH THE RESTART CYCLE
1610  441B  D3  D3           10.0           OUT     WMCCTL          ;CLEAR RESTART BIT TO FINISH THE RESTART
1611
1612                                    ;WMC ROM ERROR TEST
1613  441D  DB  DA           10.0  TST3B:  IN      WMCERR          ;GET THE WMC ERROR WORD
1614  441F  E6  30            7.0           ANI     W.ROME!W.ERR    ;SAVE 'WMC ROM ERR' & 'ERROR'
1615  4421  CA  29  44        10.0           JZ      TST3BC          ;JUMP OVER IF =0 - NO ERROR SET
1616  4424                                  ERR     TST3L,TST3BC
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4424  CD  09  28        18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0009                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4427  09                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4428  00                                            .BYTE
 (1)  4429  CD  15  28        18.0                TST3BC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  442C  DA  03  44        10.0                        JC      TST3L           ;LOOP ADDRESS IF LOOP SPECIFIED
1617                                    ;>WMC ERROR STATUS WORD INDICATES 'ERROR' AFTER WMC 'RESTART' CLEARED
1618  442F                                    ENDTST  TST3L
 (1)                                      ;TEST ITERATION CONTROL - ONCE FOR QUICK VFRIFY
 (2)  442F                                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  442F  CD  06  28        18.0                        CALL    REQST
 (2)  4432  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2)  4433  00  00                                        .WORD                   ;SYSTEM "" COUNT
 (2)  4435  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4437  00                                            .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4438  07                                            .BYTE   7               ;REQUEST CODE
 (1)  4439  3A  9A  4F        13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1)  443C  3D                 4.0           DCR     A               ;DOWNCOUNT
 (1)  443D  32  9A  4F        13.0           STA     ITERA           ;SAVE COUNT
 (1)  4440  F2  03  44        10.0           JP      TST3L           ;DO TEST UNTIL TILL = 0
```

```
                        .SBTTL  TEST 4 - BYTE COUNTER TEST
1621  4443              ST
 (1)                    ; ***************************************************************************
 (1)                    ;*TEST TITLE
 (1)                    ;*---- -----
1622                    ;*BYTE COUNTER TEST
1623  4443              SM
 (1)                    ; ***************************************************************************
 (1)                    ;*MODULE(S) UNDER TEST
 (1)                    ;*--------------------
1624                    ;*M8959
1625  4443              SD
 (1)                    ; ***************************************************************************
 (1)                    ;*DESCRIPTION
 (1)                    ;*-----------
1626                    ;*THIS TEST CHECKS THE BYTE COUNTERS ABILITY TO BE LOADED WITH ALL 0'S
1627                    ;*AND ALL 1'S
1628  4443              SP
 (1)                    ; ***************************************************************************
 (1)                    ;*PROCEDURE
 (1)                    ;*---------
1629                    ;*BGNTST
1630                    ;*  ISSUE A WMC RESTART
1631                    ;*  SET WMCCTL 'DIAG' BIT
1632                    ;*  SET WMCCTL 'DIAG' + 'ENABLE'-ENTER SPECIAL WMC ROM CODE
1633                    ;*  SMALL DELAY FOR WMC TO CLOCK THE 8253 COUNTER CHIP
1634                    ;*  LOAD WMC BYTE COUNTER WITH 0'S
1635                    ;*  IF BYTECT=0
1636                    ;*  :  THEN-CONTINUE
1637                    ;*  :  ELSE-REPORT 0'S LOAD FAILURE
1638                    ;*  ENDIF
1639                    ;*  LOAD BYTE COUNTER WITH 177777
1640                    ;*  IF BYTECT=ALL 1'S
1641                    ;*  :  THEN-CONTINUE
1642                    ;*  :  ELSE-REPORT 1'S LOAD FAILURE
1643                    ;*  ENDIF
1644                    ;*ENDTST
1645  4443              SE
 (1)                    ; ***************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
1646                    ;*WMC1 MICRO TEST 04
1647                    ;*WMC1 MICRO ERROR 12
1648                    ;*WMC1-WMC BYTE COUNTER TESTS
1649                    ;*M8959
1650                    ;*WMC BYTE COUNTER BITS 7-0 LOADED TO 0'S BUT READ NON-0
1651                    ;*ACTUAL = NNNN
1652                    ;*
1653                    ;*WMC1 MICRO TEST 04
1654                    ;*WMC1 MICRO ERROR 13
1655                    ;*WMC1-WMC BYTE COUNTER TESTS
1656                    ;*M8959
1657                    ;*WMC BYTE COUNTER BITS 15-8 LOADED TO 0'S BUT READ NON-0
1658                    ;*ACTUAL = NNNN
```

```
1659                                    ;*
1660                                    ;*WMC1 MICRO TEST 04
1661                                    ;*WMC1 MICRO ERROR 14
1662                                    ;*WMC1-WMC BYTE COUNTER TESTS
1663                                    ;*M8959
1664                                    ;*WMC BYTE COUNTER BITS 7-0 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
1665                                    ;*ACTUAL = NNNN
1666                                    ;*
1667                                    ;*WMC1 MICRO TEST 04
1668                                    ;*WMC1 MICRO ERROR 15
1669                                    ;*WMC1-WMC BYTE COUNTER TESTS
1670                                    ;*M8959
1671                                    ;*WMC BYTE COUNTER BITS 15-8 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
1672                                    ;*ACTUAL = NNNN
1673  4443                              $
 (1)                                    ; *****************************************************************************
1674
1675  4443                              TEST4:  TESTX   @4
 (1)  4443  3E  04           7.0                        MVI     A,@4            ;DEFINE THE TEST NUMBER
 (1)  4445  CD  03  28      18.0                        CALL    TSET            ;SETUP THE TEST
1676                                    ;%WMC1-WMC BYTE COUNTER TESTS
1677                                    ;&M8959
1678                                    ;INIT THE WRITE MICRO TO ENTER A SPECIAL LOOP TO JUST 'FEED' THE COUNTER
1679                                    ;CHIP (INTEL 8253) FOR BYTE COUNTER, PAD COUNTER, AND ECODE COUNTER.
1680
1681  4448  3E  01           7.0  TST4ST: MVI     A,W.RST         ;RESTART THE WRITE MICRO
1682  444A  D3  D3          10.0         OUT     WMCCTL
1683  444C  3E  02           7.0         MVI     A,W.DIAG
1684  444E  D3  D3          10.0         OUT     WMCCTL
1685  4450  3E  82           7.0         MVI     A,W.ENAB!W.DIAG ;FEED CLOCKS TO THE COUNTER CHIP
1686  4452  D3  D3          10.0         OUT     WMCCTL
1687  4454  3E  0A           7.0         MVI     A,10            ;GET A 'STALL' TIME
1688  4456  3D               4.0  T4W:    DCR     A
1689  4457  C2  56  44      10.0         JNZ     T4W             ;WAIT TILL COUNT TO 0
1690
1691                                    ;LOAD WMC BYTE COUNTER WITH ALL 0'S
1692
1693  445A  3E  30           7.0  TST4L:  MVI     A,@60           ;SELECT THE BYTE COUNTER IN COUNTER
1694  445C  D3  D7          10.0         OUT     CNTCTL          ;CONTROL WORD
1695  445E  3E  00           7.0         MVI     A,0             ;THEN LOAD THE COUNTER WITH 0'S
1696  4460  D3  D4          10.0         OUT     BYTCNT          ;LOAD BITS 7-0
1697  4462  D3  D4          10.0         OUT     BYTCNT          ;LOAD BITS 15-8
1698  4464  00               4.0         NOP
1699  4465  DB  D4          10.0         IN      BYTCNT          ;GET BITS 7-0 FROM COUNTER
1700  4467  47               4.0         MOV     B,A             ;TEMP SAVE THE DATA
1701  4468                               ROUT    ADATA           ;SAVE AS 'ACTUAL'
 (1)  4468  D3  94          10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  446A  7F               4.0                 MOV     A,A             ;RETRY LINK
1702  446B  DB  D4          10.0         IN      BYTCNT          ;GET BITS 15-8
1703  446D  4F               4.0         MOV     C,A             ;TEMP SAVE THEM
1704  446E  78               4.0         MOV     A,B             ;RESET REG A TO BITS 7-0
1705  446F  FE  00           7.0         CPI     0               ;TEST FOR 0'S DATA BYTE
1706  4471  CA  79  44      10.0         JZ      TST4AC          ;JUMP IF =0 - ALL OK
```

```
1708  4474                                        ERRA    TST4L,TST4AC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4474  CD  0F  28        18.0                        CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000A                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4477  0A                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4478  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4479  CD  15  28        18.0                TST4AC::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  447C  DA  5A  44        10.0                        JC      TST4L           ;LOOP ADDRESS IF LOOP SPECIFIED
1709                                    ;>WMC BYTE COUNTER BITS 7-0 LOADED TO 0'S BUT READ NON-0
1710
1711  447F  79                    4.0   TST4B:  MOV     A,C             ;SET REG A = BITS 15-8
1712  4480                                        ROUT    ADATA           ;SAVE 'ACTUAL' DATA
 (1)  4480  D3  94                10.0                    OUT     ADATA           ;WR  F AC INTO ADATA
 (1)  4482  7F                     4.0                    MOV     A,A             ;RETRY LINK
1713  4483  FE  00                7.0           CPI     0               ;TEST FOR 0'S
1714  4485  CA  8D  44            10.0           JZ      TST4BC          ;JUMP IF =0 - ALL OK
1715  4488                                        ERRA    TST4L,TST4BC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4488  CD  0F  28        18.0                        CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000B                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  448B  0B                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  448C  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  448D  CD  15  28        18.0                TST4BC::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4490  DA  5A  44        10.0                        JC      TST4L           ;LOOP ADDRESS IF LOOP SPECIFIED
1716                                    ·>WMC BYTE COUNTER BITS 15-8 LOADED TO 0'S BUT READ NON-0
1717  4493  3E  30                7.0   TST4C:  MVI     A,@60           ;SELECT BYTE COUNTER IN COUNTER CONTROL
1718  4495  D3  D7                10.0           OUT     CNTCTL
1719  4497  3E  FF                 7.0           MVI     A,@377          ;AND WRITE THE COUNTER TO ALL 1'S
1720  4499  D3  D4                10.0           OUT     BYTCNT          ;LOAD BITS 7-0
1721  449B  D3  D4                10.0           OUT     BYTCNT          ;LOAD BITS 15-8
1722  449D  00                     4.0           NOP
1723  449E  DB  D4                10.0           IN      BYTCNT          ;GET BITS 7-0 FROM COUNTER
1724  44A0  47                     4.0           MOV     B,A             ;TEMP SAVE THEM
1725  44A1                                        ROUT    ADATA           ;SAVE 'ACTUAL' DATA BYTE
 (1)  44A1  D3  94                10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44A3  7F                     4.0                    MOV     A,A             ;RETRY LINK
1726  44A4  DB  D4                10.0           IN      BYTCNT          ;GET BITS 15-8 FROM COUNTER
1727  44A6  4F                     4.0           MOV     C,A             ;TEMP SAVE THEM
1728
1729                                    ;TEST BYTE COUNTER FOR ALL 1'S NOW
1730  44A7  78                     4.0           MOV     A,B             ;SET REG A = BITS 7-0
1731  44A8  FE  FF                 7.0           CPI     @377            ;ALL 1'S?
1732  44AA  CA  B2  44            10.0           JZ      TST4CA          ;JUMP IF OK
1733  44AD                                        ERRA    TST4C,TST4CA
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  44AD  CD  0F  28        18.0                        CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000C                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44B0  0C                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44B1  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  44B2  CD  15  28        18.0                TST4CA::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44B5  DA  93  44        10.0                        JC      TST4C           ;LOOP ADDRESS IF LOOP SPECIFIED
1734                                    ;>WMC BYTE COUNTER BITS 7-0 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
1735
1736  44B8  79                     4.0   TST4D:  MOV     A,C             ;GET BITS 15-8
```

```
1737  44B9                                        ROUT    ADATA           ;SAVE 'ACTUAL' DATA BYTE
 (1)  44B9  D3  94          10.0                          OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44BB  7F               4.0                          MOV     A,A             ;RETRY LINK
1738  44BC  FE  FF           7.0                  CPI     @377            ;ALL 1'S?
1739  44BE  CA  C6  44       10.0                 JZ      TST4DC          ;JUMP IF ALL 1'S
1740  44C1                                        ERRA    TST4C,TST4DC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  44C1  CD  0F  28       18.0                         CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000D                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44C4  0D                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44C5  00                                         .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  44C6  CD  15  28       18.0                 TST4DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44C9  DA  93  44       10.0                         JC      TST4C           ;LOOP ADDRESS IF LOOP SPECIFIED
1741                                        ;>WMC BYTE COUNTER BITS 15-8 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
1742  44CC                                          ENDTST  TST4ST
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44CC                                          REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44CC  CD  06  28       18.0                         CALL    REQST
 (2)  44CF  00                                         .BYTE                   ;DATA PATTERN NUMBER
 (2)  44D0  00  00                                     .WORD                   ;SYSTEM "" COUNT
 (2)  44D2  00  00                                     .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  44D4  00                                         .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  44D5  07                                         .BYTE   7               ;REQUEST CODE
 (1)  44D6  3A  9A  4F       13.0                 LDA     ITERA           ;GET ITERATION COUNT
 (1)  44D9  3D               4.0                  DCR     A               ;DOWNCOUNT
 (1)  44DA  32  9A  4F       13.0                 STA     ITERA           ;SAVE COUNT
 (1)  44DD  F2  48  44       10.0                 JP      TST4ST          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 5 - WMC PAD COUNTER TEST
1745  44E0                      ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1746                            ;*PAD COUNTER TEST
1747  44E0                      SM
 (1)                            ; ***************************************************************************
 (1)                            ;*MODULE(S) UNDER TEST
 (1)                            ;*--------------------
1748                            ;*M8959
1749  44E0                      SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1750                            ;*THIS TEST CHECKS THE PAD COUNTERS ABILITY TO BE LOADED WITH ALL 0'S AND
1751                            ;*ALL 1'S
1752  44E0                      SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1753                            ;*BGNTST
1754                            ;*  SET WMCCTL RESTART BIT
1755                            ;*  IF WMCERR INDICATES ERROR
1756                            ;*  :  THEN-REPORT ERROR AFTER WMC RESTART SET
1757                            ;*  :  ELSE-CONTINUE
1758                            ;*  ENDIF
1759                            ;*  FINISH WMC RESTART CYCLE
1760                            ;*  IF WMCERR INDICATES ERROR
1761                            ;*  :  THEN-REPORT ERROR AFTER WMC RESTART CLEARED
1762                            ;*  :  ELSE-CONTINUE
1763                            ;*  ENDIF
1764                            ;*ENDTST
1765  44E0                      SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1766                            ;*WMC1 MICRO TEST 05
1767                            ;*WMC1 MICRO ERROR 16
1768                            ;*WMC1-WMC PAD COUNTER TESTS
1769                            ;*M8959
1770                            ;*WMC PAD COUNTER BITS 7-0 LOADED TO 0'S BUT READ NON-0
1771                            ;*ACTUAL = NNNN
1772                            ;*
1773                            ;*WMC1 MICRO TEST 05
1774                            ;*WMC1 MICRO ERROR 17
1775                            ;*WMC1-WMC PAD COUNTER TESTS
1776                            ;*M8959
1777                            ;*WMC PAD COUNTER BITS 15-8 LOADED TO 0'S BUT READ NON-0
1778                            ;*ACTUAL = NNNN
1779                            ;*
1780                            ;*WMC1 MICRO TEST 05
1781                            ;*WMC1 MICRO ERROR 20
1782                            ;*WMC1-WMC PAD COUNTER TESTS
```

```
                                        ;*M8959
                                        ;*WMC PAD COUNTER BITS 7-0 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
                                        ;*ACTUAL = NNNN
                                        ;*
                                        ;*WMC1 MICRO TEST 05
                                        ;*WMC1 MICRO ERROR 21
                                        ;*WMC1-WMC PAD COUNTER TESTS
                                        ;*M8959
                                        ;*WMC PAD COUNTER BITS 15-8 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
                                        ;*ACTUAL = NNNN
44E0                                    $
 (1)                                    ; ************************************************************************
                                        
44E0                                    TEST5:  TESTX   @5
 (1) 44E0  3E  05            7.0                         MVI     A,@5            ;DEFINE THE TEST NUMBER
 (1) 44E2  CD  03  28       18.0                         CALL    TSET            ;SETUP THE TEST
                                        ;%WMC1-WMC PAD COUNTER TESTS
                                        ;&M8959
                                        ;LOAD WMC PAD COUNTER WITH ALL 0'S
                                        
44E5  3E  70                7.0 TST5L:  MVI     A,@160          ;SELECT THE PAD COUNTER IN COUNTER
44E7  D3  D7               10.0        OUT     CNTCTL          ;CONTROL WORD
44E9  3E  00                7.0        MVI     A,0             ;THEN LOAD THE COUNTER WITH 0'S
44EB  D3  D5               10.0        OUT     PADCNT          ;LOAD BITS 7-0
44ED  D3  D5               10.0        OUT     PADCNT          ;LOAD BITS 15-8
44EF  00                    4.0        NOP
44F0  DB  D5               10.0        IN      PADCNT          ;GET BITS 7-0 FROM COUNTER
44F2  47                    4.0        MOV     B,A             ;TEMP SAVE THE DATA
44F3                                   ROUT    ADATA           ;SAVE AS 'ACTUAL'
 (1) 44F3  D3  94           10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1) 44F5  7F                4.0                MOV     A,A             ;RETRY LINK
44F6  DB  D5               10.0        IN      PADCNT          ;GET BITS 15-8
44F8  4F                    4.0        MOV     C,A             ;TEMP SAVE THEM
44F9  78                    4.0        MOV     A,B             ;RESET REG A TO BITS 7-0
44FA  FE  00                7.0        CPI     0               ;TEST FOR 0'S DATA BYTE
44FC  CA  04  45           10.0        JZ      TST5AC          ;JUMP IF =0 - ALL OK
44FF                                   ERRA    TST5L,TST5AC
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1) 44FF  CD  0F  28       18.0                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       000E                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 4502  0E                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 4503  00                                .BYTE                   ;PRINT ROUTINE NUMBER
 (1) 4504  CD  15  28       18.0        TST5AC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1) 4507  DA  E5  44       10.0                JC      TST5L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>WMC PAD COUNTER BITS 7-0 LOADED TO 0'S BUT READ NON-0
                                        ;CHECK BITS 15-0 FOR 0'S
                                        
450A  79                    4.0 TST5B:  MOV     A,C             ;SET REG A = BITS 15-8
450B                                   ROUT    ADATA           ;SAVE 'ACTUAL' DATA
 (1) 450B  D3  94           10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1) 450D  7F                4.0                MOV     A,A             ;RETRY LINK
450E  FE  00                7.0        CPI     0               ;TEST FOR 0'S
4510  CA  18  45           10.0        JZ      TST5BC          ;JUMP IF  0 - ALL OK
```

```
1823  4513                                        ERRA    TST5L,TST5BC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4513  CD  0F  28      18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       000F                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4516  0F                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4517  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4518  CD  15  28      18.0                TST5BC::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  451B  DA  E5  44      10.0                          JC      TST5L           ;LOOP ADDRESS IF LOOP SPECIFIED
1824                                    ;>WMC PAD COUNTER BITS 15-8 LOADED TO 0'S BUT READ NON-0
1825  451E  3E  70           7.0   TST5C:  MVI     A,@160          ;SELECT PAD COUNTER IN COUNTER CONTROL
1826  4520  D3  D7          10.0           OUT     CNTCTL
1827  4522  3E  FF           7.0           MVI     A,@377          ;AND WRITE THE COUNTER TO ALL 1'S
1828  4524  D3  D5          10.0           OUT     PADCNT          ;LOAD BITS 7-0
1829  4526  D3  D5          10.0           OUT     PADCNT          ;LOAD BITS 15-8
1830  4528  00           4.0               NOP
1831  4529  DB  D5          10.0           IN      PADCNT          ;GET BITS 7-0 FROM COUNTER
1832  452B  47           4.0               MOV     B,A             ;TEMP SAVE THEM
1833  452C                                  ROUT    ADATA           ;SAVE 'ACTUAL' DATA BYTE
 (1)  452C  D3  94          10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  452E  7F           4.0                       MOV     A,A             ;RETRY LINK
1834  452F  DB  D5          10.0           IN      PADCNT          ;GET BITS 15-8 FROM COUNTER
1835  4531  4F           4.0               MOV     C,A             ;TEMP SAVE THEM
1836
1837                                    ;TEST PAD COUNTER FOR ALL 1'S NOW
1838
1839  4532  78           4.0               MOV     A,B             ;SET REG A = BITS 7-0
1840  4533  FE  FF           7.0           CPI     @377            ;ALL 1'S?
1841  4535  CA  3D  45      10.0           JZ      TST5CA          ;JUMP IF OK
1842  4538                                  ERRA    TST5C,TST5CA
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4538  CD  0F  28      18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0010                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  453B  10                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  453C  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  453D  CD  15  28      18.0                TST5CA::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4540  DA  1E  45      10.0                          JC      TST5C           ;LOOP ADDRESS IF LOOP SPECIFIED
1843                                    ;>WMC PAD COUNTER BITS 7-0 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
1844  4543  79           4.0   TST5D:  MOV     A,C             ;GET BITS 15-8
1845  4544                                  ROUT    ADATA           ;SAVE 'ACTUAL' DATA BYTE
 (1)  4544  D3  94          10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4546  7F           4.0                       MOV     A,A             ;RETRY LINK
1846  4547  FE  FF           7.0           CPI     @377            ;ALL 1'S?
1847  4549  CA  51  45      10.0           JZ      TST5DC          ;JUMP IF ALL 1'S
1848  454C                                  ERRA    TST5C,TST5DC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  454C  CD  0F  28      18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0011                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  454F  11                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4550  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4551  CD  15  28      18.0                TST5DC::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4554  DA  1E  45      10.0                          JC      TST5C           ;LOOP ADDRESS IF LOOP SPECIFIED
1849                                    ;>WMC PAD COUNTER BITS 15-8 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
```

```
1851  4557                                        ENDTST  TST5L
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4557                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4557   CD   06   28        18.0                                CALL    REQST
 (2)  455A   00                                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  455B   00   00                                                 .WORD                   ;SYSTEM "" COUNT
 (2)  455D   00   00                                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  455F   00                                                      .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4560   07                                                      .BYTE   7               ;REQUEST CODE
 (1)  4561   3A   9A   4F        13.0            LDA     ITERA               ;GET ITERATION COUNT
 (1)  4564   3D                  4.0             DCR     A                   ;DOWNCOUNT
 (1)  4565   32   9A   4F        13.0            STA     ITERA               ;SAVE COUNT
 (1)  4568   F2   E5   44        10.0            JP      TST5L               ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 6 - WMC ECODE COUNTER TEST
                        456B    ST
                                ; ******************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*ECODE COUNTER TEST
                        456B    SM
                                ; ******************************************************************************
                                ;*MODULE(S) UNDER TEST
                                ;*--------------------
                                ;*M8959
                        456B    SD
                                ; ******************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ECODE COUNTERS ABILITY TO BE LOADED WITH ALL 0'S AND
                                ;*ALL 1'S
                        456B    SP
                                ; ******************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  SET WMCCTL RESTART BIT
                                ;*  IF WMCERR INDICATES ERROR
                                ;*  :  THEN-REPORT ERROR AFTER WMC RESTART SET
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  FINISH WMC RESTART CYCLE
                                ;*  IF WMCERR INDICATES ERROR
                                ;*  :  THEN-REPORT ERROR AFTER WMC RESTART CLEARED
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*ENDTST
                        456B    SE
                                ; ******************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*WMC1 MICRO TEST 06
                                ;*WMC1 MICRO ERROR 22
                                ;*WMC1-WMC ECODE COUNTER TESTS
                                ;*M8959
                                ;*WMC ECODE COUNTER BITS 7-0 LOADED TO 0'S BUT READ NON-0
                                ;*ACTUAL = NNNN
                                ;*
                                ;*WMC1 MICRO TEST 06
                                ;*WMC1 MICRO ERROR 23
                                ;*WMC1-WMC ECODE COUNTER TESTS
                                ;*M8959
                                ;*WMC ECODE COUNTER BITS 15-8 LOADED TO 0'S BUT READ NON-0
                                ;*ACTUAL = NNNN
                                ;*
                                ;*WMC1 MICRO TEST 06
                                ;*WMC1 MICRO ERROR 24
                                ;*WMC1-WMC ECODE COUNTER TESTS
```

```
                                        ;*M8959
                                        ;*WMC ECODE COUNTER BITS 7-0 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
                                        ;*ACTUAL = NNNN
                                        ;*
                                        ;*WMC1 MICRO TEST 06
                                        ;*WMC1 MICRO ERROR 25
                                        ;*WMC1-WMC ECODE COUNTER TESTS
                                        ;*M8959
                                        ;*WMC ECODE COUNTER BITS 15-8 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
                                        ;*ACTUAL = NNNN
456B                                    $
 (1)                                    ; ***************************************************************************
                                        TEST6:  TESTX   @6
 (1) 456D  3E  06            7.0                        MVI     A,@6            ;DEFINE THE TEST NUMBER
 (1) 456F  CD  03  28       18.0                        CALL    TSET            ;SETUP THE TEST
                                        ;%WMC1-WMC ECODE COUNTER TESTS
                                        ;&M8959
                                        ;LOAD WMC ECODE COUNTER WITH ALL 0'S
4570  3E  B0                 7.0  TST6L:  MVI     A,@260          ;SELECT THE ECODE COUNTER IN COUNTER
4572  D3  D7                10.0         OUT     CNTCTL          ;CONTROL WORD
4574  3E  00                 7.0         MVI     A,0             ;THEN LOAD THE COUNTER WITH 0'S
4576  D3  D6                10.0         OUT     ERRCNT          ;LOAD BITS 7-0
4578  D3  D6                10.0         OUT     ERRCNT          ;LOAD BITS 15-8
457A  00                     4.0         NOP
457B  DB  D6                10.0         IN      ERRCNT          ;GET BITS 7-0 FROM COUNTER
457D  47                     4.0         MOV     B,A             ;TEMP SAVE THE DATA
457E                                     ROUT    ADATA           ;SAVE AS 'ACTUAL'
 (1) 457E  D3  94            10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1) 4580  7F                 4.0                 MOV     A,A             ;RETRY LINK
4581  DB  D6                10.0         IN      ERRCNT          ;GET BITS 15-8
4583  4F                     4.0         MOV     C,A             ;TEMP SAVE THEM
4584  78                     4.0         MOV     A,B             ;RESET REG A TO BITS 7-0
4585  FE  00                 7.0         CPI     0               ;TEST FOR 0'S DATA BYTE
4587  CA  8F  45            10.0         JZ      TST6AC          ;JUMP IF =0 - ALL OK
458A                                     ERRA    TST6L,TST6AC
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1) 458A  CD  0F  28       18.0                 CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0012                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 458D  12                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 458E  00                             .BYTE                   ;PRINT ROUTINE NUMBER
 (1) 458F  CD  15  28       18.0  TST6AC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1) 4592  DA  70  45       10.0                 JC      TST6L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>WMC ECODE COUNTER BITS 7-0 LOADED TO 0'S BUT READ NON-0
                                        ;CHECK BITS 15-0 FOR 0'S
4595  79                     4.0  TST6B:  MOV     A,C             ;SET REG A = BITS 15-8
4596                                     ROUT    ADATA           ;SAVE 'ACTUAL' DATA
 (1) 4596  D3  94            10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1) 4598  7F                 4.0                 MOV     A,A             ;RETRY LINK
4599  FE  00                 7.0         CPI     0               ;TEST FOR 0'S
459B  CA  A3  45            10.0         JZ      TST6BC          ;JUMP IF =0 - ALL OK
```

```
1932  459E                                          ERRA    TST6L,TST6BC
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  459E   CD   0F   28        18.0                      CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0013                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45A1   13                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45A2   00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45A3   CD   15   28        18.0              TST6BC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45A6   DA   70   45        10.0                      JC      TST6L           ;LOOP ADDRESS IF LOOP SPECIFIED
1933                                        ;>WMC ECODE COUNTER BITS 15-8 LOADED TO 0'S BUT READ NON-0
1934  45A9   3E   B0                 7.0  TST6C:  MVI     A,@260          ;SELECT ECODE COUNTER IN COUNTER CONTROL
1935  45AB   D3   D7                10.0          OUT     CNTCTL
1936  45AD   3E   FF                 7.0          MVI     A,@377          ;AND WRITE THE COUNTER TO ALL 1'S
1937  45AF   D3   D6                10.0          OUT     ERRCNT          ;LOAD BITS 7-0
1938  45B1   D3   D6                10.0          OUT     ERRCNT          ;LOAD BITS 15-8
1939  45B3   00                     4.0          NOP
1940  45B4   DB   D6                10.0          IN      ERRCNT          ;GET BITS 7-0 FROM COUNTER
1941  45B6   47                     4.0          MOV     B,A             ;TEMP SAVE THEM
1942  45B7                                       ROUT    ADATA           ;SAVE 'ACTUAL' DATA BYTE
 (1)  45B7   D3   94                10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  45B9   7F                     4.0                  MOV     A,A             ;RETRY LINK
1943  45BA   DB   D6                10.0          IN      ERRCNT          ;GET BITS 15-8 FROM COUNTER
1944  45BC   4F                     4.0          MOV     C,A             ;TEMP SAVE THEM
1945
1946                                        ;TEST ECODE COUNTER FOR ALL 1'S NOW
1947  45BD   78                     4.0          MOV     A,B             ;SET REG A = BITS 7-0
1948  45BE   FE   FF                 7.0          CPI     @377            ;ALL 1'S?
1949  45C0   CA   C8   45            10.0         JZ      TST6CA          ;JUMP IF OK
1950  45C3                                       ERRA    TST6C,TST6CA
 (1)                                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  45C3   CD   0F   28        18.0                      CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0014                                           MSGN    =       MSGN+1          ;UPDATE, MESSAGE NUMBER FOR THIS
 (1)  45C6   14                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45C7   00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45C8   CD   15   28        18.0              TST6CA::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45CB   DA   A9   45        10.0                      JC      TST6C           ;LOOP ADDRESS IF LOOP SPECIFIED
1951                                        ;>WMC ECODE COUNTER BITS 7-0 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
1952
1953  45CE   79                     4.0  TST6D:  MOV     A,C             ;GET BITS 15-8
1954  45CF   D3   94                10.0          OUT     @224            ;SAVE 'ACTUAL' DATA BYTE
1955  45D1   FE   FF                 7.0          CPI     @377            ;ALL 1'S?
1956  45D3   CA   DB   45            10.0         JZ      TST6DC          ;JUMP IF ALL 1'S
1957  45D6                                       ERRA    TST6C,TST6DC
 (1)                                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  45D6   CD   0F   28        18.0                      CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0015                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45D9   15                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45DA   00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45DB   CD   15   28        18.0              TST6DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45DE   DA   A9   45        10.0                      JC      TST6C           ;LOOP ADDRESS IF LOOP SPECIFIED
1958                                        ;>WMC ECODE COUNTER BITS 15-8 LOADED WITH ALL 1'S BUT NOT 1'S WHEN READ
```

```
 1960  45E1                                        ENDTST  TST6L
  (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  45E1                                       REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  45E1  CD   06   28       18.0                                CALL    REQST
  (2)  45E4  00                                                     .BYTE                   ;DATA PATTERN NUMBER
  (2)  45E5  00   00                                                .WORD                   ;SYSTEM '*' COUNT
  (2)  45E7  00   00                                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  45E9  00                                                     .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  45EA  07                                                     .BYTE   7               ;REQUEST CODE
  (1)  45EB  3A   9A   4F       13.0          LDA     ITERA           ;GET ITERATION COUNT
  (1)  45EE  3D                  4.0          DCR     A               ;DOWNCOUNT
  (1)  45EF  32   9A   4F       13.0          STA     ITERA           ;SAVE COUNT
  (1)  45F2  F2   70   45       10.0          JP      TST6L           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 7 - WMC 'ECODE' TEST FOR ILLEGAL FORMAT CODE #7
45F5                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*ILLEGAL PACKING MODE TEST
45F5                            SM
 (1)                            ; ************************************************************************
 (1)                            ;*MODULE(S) UNDER TEST
 (1)                            ;*--------------------
                                ;*M8959
45F5                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST IS ATTEMPTING TO SEE IF THE WMC ROM PROGRAM CAN DETECT
                                ;*WRONG INFORMATION IN THE WMC DATA CONTROL WORD FOR THE 'FORMAT'
                                ;*DATA...AN ILLEGAL FORMAT CODE IS WRITTEN IN THE DATA CONTROL WORD
                                ;*AND THEN SET 'WMC ENABLE' IN THE WMC CONTROL WORD TO ALLOW THE
                                ;*WMC ROM PROGRAM DETECT THE ERROR CONDITION.
                                ;*
                                ;*FORMAT CODE 7 SHOULD CAUSE THE 'ECODE' COUNTER TO CONTAIN A
                                ;*-3 COUNT WHEN THE ERROR IS DETECTED. WMC ERROR STATUS WORD SHOULD
                                ;*HAVE 'ERROR' SET ON THESE CODES.
                                ;*
                                ;*FORMAT CODES 0-6 SHOULD CAUSE NO ERROR.
45F5                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  ISSUE A WMC RESTART
                                ;*  CLEAR ERROR CODE COUNTER
                                ;*  LOAD DATACTL WITH INVALID PACKING MODE #7
                                ;*  SET WMCCTL 'ENABLE' BIT OT STAPT WMC ROM PROGRAM PROCESS
                                ;*  SMALL DELAY SO WMC CAN DETECT ILLEGAL FORMAT NUMBER
                                ;*  IF WMCSTA NOT-DONE=0 (WMC FINISHED)
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT POSSIBLE WMC SELF-DIAG. FAILURE
                                ;*  ENDIF
                                ;*  IF ERROR COUNTER DECREMENTED TO -3 (ILLEG. FMT CODE ISSUED)
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT 'ECODE' COUNT NOT=-3
                                ;*  ENDIF
                                ;*  IF WMCERR WMC ROM ERROR BIT=1 AND 'ERROR'=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT FMT CODE 7 FAILED TO CAUSE PROPER WMC ERROR STATUS
                                ;*  ENDIF
                                ;*ENDTST
45F5                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*WMC1 MICRO TEST 07
```

```
2001                                    ;*WMC1 MICRO ERROR 26
2002                                    ;*WMC1-WMC 'ECODE' TEST FOR ILLEGAL FORMAT CODE #7
2003                                    ;*M8959
2004                                    ;*WMC NOT READY - WMC GROSS ERROR! POSSIBLE WMC SELF-DIAGNOSIS FAILURE.
2005                                    ;*
2006                                    ;*WMC1 MICRO TEST 07
2007                                    ;*WMC1 MICRO ERROR 27
2008                                    ;*WMC1-WMC 'ECODE' TEST FOR ILLEGAL FORMAT CODE #7
2009                                    ;*M8959
2010                                    ;*WMC 'ECODE' COUNT NOT -3 (375) AFTER ILLEGAL FORMAT CODE #7 USED
2011                                    ;*ACTUAL = NNNN
2012                                    ;*
2013                                    ;*WMC1 MICRO TEST 07
2014                                    ;*WMC1 MICRO ERROR 30
2015                                    ;*WMC1-WMC 'ECODE' TEST FOR ILLEGAL FORMAT CODE #7
2016                                    ;*M8959
2017                                    ;*WMC 'ECODE' COUNTER BITS 15-8 NOT ALL 1'S ON DETECTION OF ILLEGAL
2018                                    ;*FORMAT CODE #7
2019                                    ;*ACTUAL = NNNN
2020                                    ;*
2021                                    ;*WMC1 MICRO TEST 07
2022                                    ;*WMC1 MICRO ERROR 31
2023                                    ;*WMC1-WMC 'ECODE' TEST FOR ILLEGAL FORMAT CODE #7
2024                                    ;*M8959
2025                                    ;*WMC ERROR STATUS DOES NOT INDICATE 'ROM ERROR' FROM ILLEGAL FORMAT CODE #7
2026                                    ;*ACTUAL = NNNN
2027                                    ;*
2028                                    ;*WMC1 MICRO TEST 07
2029                                    ;*WMC1 MICRO ERROR 32
2030                                    ;*WMC1-WMC 'ECODE' TEST FOR ILLEGAL FORMAT CODE #7
2031                                    ;*M8959
2032                                    ;*WMC ERROR STATUS DOES NOT INDICATE 'ERROR' FROM ILLEGAL FORMAT CODE #7
2033                                    ;*ACTUAL = NNNN
2034  45F5                              $
 (1)                                    ; *****************************************************************************
2035
2036  45F5                              TEST7:  TESTX   @7
 (1)  45F5  3E  07          7.0                         MVI     A,@7            ;DEFINE THE TEST NUMBER
 (1)  45F7  CD  03  28     18.0                         CALL    TSET            ;SETUP THE TEST
2037                                    ;%WMC1-WMC 'ECODE' TEST FOR ILLEGAL FORMAT CODE #7
2038                                    ;&M8959
2039
2040                                    ;RESTART THE WMC PROGRAM
2041  45FA  3E  01               7.0   TST7L:  MVI     A,W.RST
2042  45FC  D3  D3              10.0           OUT     WMCCTL          ;ISSUE WMC RESTART
2043  45FE  AF                   4.0           XRA     A
2044  45FF  D3  D3              10.0           OUT     WMCCTL          ;CLEAR THE RESTART BIT
2045
2046                                    ;INIT THE 'ECODE' COUNTER TO 0
2047
2048  4601                              CLRECT
 (1)  4601  AF                   4.0                         XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  4602  D3  D6              10.0                         OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  4604  D3  D6              10.0                         OUT     ERRCNT          ;CLEAR BITS 15-8
```

```
2049
2050                                    ;LOAD INVALID FORMAT CODE #7 IN WMC DATA CONTROL WORD
2051
2052  4606  3E  70          7.0         MVI     A,@160          ;GET CODE #7
2053  4608  D3  DC         10.0         OUT     DATACTL         ;LOAD IT IN WMC DATA CONTROL WORD
2054                                    ;SET WMC CONTROL TO 'WMC ENABLE' SO THE ERROR CAN BE DETECTED
2055
2056  460A  3E  80          7.0  TST7A: MVI     A,W.ENAB        ;ENABLE WMC
2057  460C  D3  D3         10.0         OUT     WMCCTL
2058  460E  3E  0A          7.0         MVI     A,10            ;GET A DELAY COUNT
2059  4610  3D              4.0  T7W:    DCR     A
2060  4611  C2  10  46      10.0         JNZ     T7W             ;WAIT TILL COUNT TO 0
2061  4614  DB  D0         10.0         IN      WMCSTA          ;GET WMC STATUS INFO
2062  4616  E6  40          7.0         ANI     W.DONN          ;CHECK TO SEE IF WMC PROGRAM IS READY
2063  4618  CA  20  46      10.0         JZ      TST7AC          ;JUMP OVER IF WMC IS DONE
2064  461B                               ERR     TST7L,TST7AC
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  461B  CD  09  28      18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0016                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  461E  16                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  461F  00                          .BYTE
 (1)  4620  CD  15  28      18.0  TST7AC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4623  DA  FA  45      10.0                 JC      TST7L           ;LOOP ADDRESS IF LOOP SPECIFIED
2065                                    ;>WMC NOT READY - WMC GROSS ERROR! POSSIBLE WMC SELF-DIAGNOSIS FAILURE.
2066
2067  4626  DB  D6         10.0  TST7B: IN      ERRCNT          ;GET BITS 7-0
2068  4628  47              4.0         MOV     B,A             ;TEMP SAVE
2069  4629                              ROUT    ADATA           ;SAVE 'ACTUAL' DATA BYTE
 (1)  4629  D3  94         10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  462B  7F              4.0                 MOV     A,A             ;RETRY LINK
2070  462C  DB  D6         10.0         IN      ERRCNT          ;GET BITS 15-8 OF 'ECODE' COUNTER
2071  462E  4F              4.0         MOV     C,A             ;TEMP SAVE BITS
2072  462F  78              4.0         MOV     A,B             ;GET 7-0 AGAIN
2073  4630  FE  FD          7.0         CPI     @375            ;SET TO MINUS 3??
2074  4632  CA  3A  46      10.0         JZ      TST7C           ;JUMP IF CORRECT
2075  4635                               ERRA    TST7L,TST7BC
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4635  CD  0F  28      18.0                 CALL    ERLPA           ;PROCESS ERROR - DC 2.3
 (1)       0017                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4638  17                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4639  00                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  463A  CD  15  28      18.0  TST7BC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  463D  DA  FA  45      10.0                 JC      TST7L           ;LOOP ADDRESS IF LOOP SPECIFIED
2076                                    ;>WMC 'ECODE' COUNT NOT -3 (375) AFTER ILLEGAL FORMAT CODE #7 USED
2077
2078  4640  79              4.0  TST7C: MOV     A,C             ;GET BITS 15-8 OF 'ECODE'
2079  4641                              ROUT    ADATA           ;SAVE 'ACTUAL' DATA BYTE
 (1)  4641  D3  94         10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4643  7F              4.0                 MOV     A,A             ;RETRY LINK
2080  4644  FE  FF          7.0         CPI     @377            ;SHOULD BE ALL 1'S
208   4646  CA  4E  46      10.0         JZ      TST7CC          ;JUMP IF CORRECT
```

```
2083  4649                                    ERRA    TST7L,TST7CC
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4649  CD  0F  28      18.0                      CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0018                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  464C  18                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  464D  00                                        .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  464E  CD  15  28      18.0                TST7CC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4651  DA  FA  45      10.0                        JC      TST7L           ;LOOP ADDRESS IF LOOP SPECIFIED
2084                                    ;>WMC 'ECODE' COUNTER BITS 15-8 NOT ALL 1'S ON DETECTION OF ILLEGAL
2085                                    ;>FORMAT CODE #7
2086
2087                                    ;CHECK WMC ERROR STATUS WORD FOR 'ERROR' INDICATION FROM THE ILLEGAL
2088                                    ;FORMAT CODE
2089
2090  4654  DB  DA          10.0  TST7D:  IN      WMCERR          ;GET THE ERROR STATUS
2091  4656  E6  10           7.0         ANI     W.ROME          ;SEE IF 'ERROR' IS SET
2092  4658  C2  60  46       10.0         JNZ     TST7DC          ;JUMP IF IT IS SET
2093  465B                                ERRA    TST7L,TST7DC
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  465B  CD  0F  28      18.0                      CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0019                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  465E  19                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  465F  00                                        .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4660  CD  15  28      18.0                TST7DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4663  DA  FA  45      10.0                        JC      TST7L           ;LOOP ADDRESS IF LOOP SPECIFIED
2094                                    ;>WMC ERROR STATUS DOES NOT INDICATE 'ROM ERROR' FROM ILLEGAL FORMAT CODE #7
2095
2096  4666  DB  DA          10.0  TST7E:  IN      WMCERR          ;GET THE ERROR STATUS
2097  4668  E6  20           7.0         ANI     W.ERR           ;SEE IF 'ERROR' IS SET
2098  466A  C2  72  46       10.0         JNZ     TST7EC          ;JUMP IF IT IS SET
2099  466D                                ERRA    TST7L,TST7EC
 (1)                                          ,FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  466D  CD  0F  28      18.0                      CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        001A                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4670  1A                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4671  00                                        .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4672  CD  15  28      18.0                TST7EC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4675  DA  FA  45      10.0                        JC      TST7L           ;LOOP ADDRESS IF LOOP SPECIFIED
2100                                    ;>WMC ERROR STATUS DOES NOT INDICATE 'ERROR' FROM ILLEGAL FORMAT CODE #7
2101  4678                                  ENDTST  TST7L
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4678                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4678  CD  06  28      18.0                      CALL    REQST
 (2)  4678  00                                        .BYTE                   ;DATA PATTERN NUMBER
 (2)  467C  00  00                                    .WORD                   ;SYSTEM '*' COUNT
 (2)  467E  00  00                                    .WORD   ?               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4680  00                                        .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4681  07                                        .BYTE   7               ;REQUEST CODE
 (1)  4682  3A  9A  4F      13.0                LDA     ITERA           ;GET ITERATION COUNT
 (1)  4685  3D               4.0                 DCR     A               ;DOWNCOUNT
 (1)  4686  32  9A  4F      13.0                STA     ITERA           ;SAVE COUNT
 (1)  4689  F2  FA  45      10.0                JP      TST7L           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 10 - WMC SYSTEM CLOCK TEST
 468C                           ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*SYSTEM CLOCK TEST
 468C                           SM
                                ;*****************************************************************************
                                ;*MODULE(S) UNDER TEST
                                ;*--------------------
                                ;*M8959,M8960
 468C                           SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST VERIFYS ALL SYSTEM CLOCKS FOR THE WMC
                                ;*BY UTALIZING THE ILLEGAL FORMAT CODE #7 JUST TESTED
                                ;*
                                ;*FIRST THE -20% CLOCK IS APPLIED TO THE WMC - THEN
                                ;*A RESTART AND ILLEGAL FORMAT 7 OPERATION ARE PERFORMED
                                ;*AND THE WMCERR WORD IS MONITORED FOR THE EXPECTED
                                ;*ERROR - FAILURE OF THE EXPECTED ERROR CONDITION
                                ;*TO SET INDICATES THE WMC IS NOT RECEIVENG THE SELECTED CLOCK.
                                ;*THE PROCEDURE IS THEN REPEATED FOR THE +10%, +20%, AND NORMAL CLOCKS.
 468C                           SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    SET SYSTEM CCLOCK TO -20%
                                ;*       BGNDO
                                ;*          ISSUE A WMC RESTART
                                ;*          SET FORMAT 7 - ILLEGAL
                                ;*          SET 'WMC ENABLE' TO START PROCESSING
                                ;*          SMALL DELAY
                                ;*          IF ROM ERROR AND ERROR SET
                                ;*             THEN - CONTINUE
                                ;*             ELSE - ERROR
                                ;*          ENDIF
                                ;*          ISSUE A 'WMC RESTART' TO CLEANUP
                                ;*          IF ROM ERROR AND ERROR CLEAR
                                ;*             THEN - CONTINUE
                                ;*             ELSE - ERROR
                                ;*          ENDIF
                                ;*          CHANGE SYSTEM CLOCKS AND DO AGAIN UNTIL CLOCK=NORMAL
                                ;*       ENDDO
                                ;*ENDTST
 468C                           SE
                                ;*****************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*WMC1 MICRO TEST 10
                                ;*WMC1 MICRO ERROR
                                ;*WMC1-SYSTEN CLOCK TEST
```

```
2142                                    ;*M8959, M8960
2143                                    ;*NO ERROR SET AFTER ILLEGAL FORMAT CODE 7
2144                                    ;*ACTUAL = NNNN
2145                                    ;*ACTUAL DATA IS THE CLOCK CONTROLE WORD AT THE TIME OF FAILURE
2146  468C                              $
 (1)                                    ; ************************************************************************
2147
2148  468C                              TEST10: TESTX   @10
 (1)  468C  3E  08            7.0                       MVI     A,@10           ;DEFINE THE TEST NUMBER
 (1)  468E  CD  03  28       18.0                       CALL    TSET            ;SETUP THE TEST
2149                                    ;%WMC1 - SYSTEM CLOCK TEST
2150                                    ;&M8959, M8960
2151
2152  4691  3E  03               7.0  TST10R: MVI     A,@03           ;SET THE SYSTEM CLOCK TO -20%
2153  4693  32  EA  46          13.0          STA     SPEED           ;SAVE THE CLOCK FOR POSSIBLE ERROR REPORT
2154  4696  D3  F0              10.0  TST10L: OUT     CLKCTL
2155
2156  4698  3E  01               7.0          MVI     A,W.RST         ;ISSUE WMC RESET
2157  469A  D3  D3              10.0          OUT     WMCCTL          ;
2158  469C  AF                   4.0          XRA     A               ;CLEAR THE RESTART BIT
2159  469D  D3  D3              10.0          OUT     WMCCTL          ;TO FINISH THE RESET
2160
2161  469F  3E  70               7.0          MVI     A,@160          ;LOAD FORMAT CODE 7 - ILLEGAL
2162  46A1  D3  D0              10.0          OUT     DATACTL         ;
2163  46A3  3E  80               7.0          MVI     A,W.ENAB        ;ENABLE WMC
2164  46A5  D3  D3              10.0          OUT     WMCCTL          ;
2165                                                                  ;
2166
2167  46A7  3E  0A               7.0          MVI     A,10            ;DELAY FOR OPERATION TO COMPLEAT
2168  46A9  3D                   4.0  T10W:   DCR     A               ;
2169  46AA  C2  A9  46          10.0          JNZ     T10W            ;
2170
2171  46AD  DB  DA              10.0          IN      WMCERR          ;CHECKL FOR EXPECTED ERROR
2172  46AF  E6  30               7.0          ANI     W.ROME+W.ERR    ;
2173  46B1  C2  BF  46          10.0          JNZ     TST10A          ;CORRECT - CONTINUE
2174
2175  46B4  3A  EA  46          13.0          LDA     SPEED           ;GET SYSTEM CLOCK FOR ERROR REPORT
2176  46B7                                    ROUT    ADATA           ;
 (1)  46B7  D3  94            10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46B9  7F                 4.0                  MOV     A,A             ;RETRY LINK
2177
2178  46BA                                    ERRA    TST10L,TST10A   ;
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  46BA  CD  0F  28        18.0                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       001B                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46BD  1B                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46BE  00                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  46BF  CD  15  28        18.0                  TST10A::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46C2  DA  96  46        10.0                          JC      TST10L          ;LOOP ADDRESS IF LOOP SPECIFIED
2179                                    ;>NO ERROR SET AFTER ILLEGAL FORMAT CODE 7
2180                                    ;<ACTUAL DATA IS THE CLOCK CONTROLE WORD AT THE TIME OF FAILURE
2181
2182  46C5  3A  EA  46          13.0          LDA     SPEED           ;GET THE CLOCK CONTROLE WORD
2183  46C8  A7                   4.0          ANA     A               ;SET CONDITION BITS
```

```
2184  46C9  CA  D3  46   10.0          JZ      TST10B          ;ALL CLOCKS TESTED - GET OUT
2185  46CC  3D                 4.0     DCR     A               ;NO - GO DO IT AGAIN WITH NEXT CLOCK
2186  46CD  32  EA  46   13.0          STA     SPEED           ;SAVE THE CLOCK CONTROLE WORD
2187  46D0  C3  96  46   10.0          JMP     TST10L          ;
2188
2189  46D3                        TST10B: ENDTST  TST10R
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  46D3                          REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  46D3  CD  06  28   18.0                          CALL    REQST
 (2)  46D6  00                                        .BYTE                   ;DATA PATTERN NUMBER
 (2)  46D7  00  00                                    .WORD                   ;SYSTEM '*' COUNT
 (2)  46D9  00  00                                    .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  46DB  00                                        .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  46DC  07                                        .BYTE   7               ;REQUEST CODE
 (1)  46DD  3A  9A  4F   13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  46E0  3D                 4.0     DCR     A               ;DOWNCOUNT
 (1)  46E1  32  9A  4F   13.0          STA     ITERA           ;SAVE COUNT
 (1)  46E4  F2  91  46   10.0          JP      TST10R          ;DO TEST UNTIL TILL = 0
2190  46E7  C3  18  28   10.0          JMP     TSTEND          ;END OF THE TEST
2191
2192
2193
2194  46EA  00                    SPEED:  .BYTE   0               ;STORAGE FOR CLOCK CONTROLE WORD
2195        0000                  .END
```

```
A      =%0007   ADATA  = 0094   AMTIEP= 0001   AMTIE7= 0002
ARAIDF= 0098    ASAVE   4F9B    ATTCD   4F97    AXNUM   4F91
B      =%0000   BADST  = 0090   BIT0   = 0001   BIT1   = 0002
BIT15  = 8000   BIT2   = 0004   BIT3   = 0008   BIT4   = 0010
BIT5   = 0020   BIT6   = 0040   BIT7   = 0080   BIT8   = 0100
BIT9   = 0200   BRKPBC= 4F0A    BRKRAM= 4F10    BRKSTR= 4E60
BRKXCT= 4F00    BSAVE   4F9C    BYTCNT= 00D4    BYTEH   4F24
BYTEL   4F23    C      =%0001   CASCT   4F21    CASCTL= 00A0
CASSTA= 00A0    CATTH  = 0089   CATTL  = 0088   CBUSST= 00A1
CBYTH  = 008B   CBYT.  = 008A   CDG1H  = 0087   CDG1L  = 0086
CDG2H  = 0093   CDG2L  = 0092   CDG3H  = 0095   CDG3L  = 0094
CDVTH  = 008D   CDVTL  = 008C   CHPTIE= 0028   CHOTIE= 0020
CH1TIE= 0021    CH2TIE= 0022    CH3TIE= 0023    CH4TIE= 0024
CH5TIE= 0025    CH6TIE= 0026    CH7TIE= 0027    CKLOP  = 2815
CLKCTL= 00F0    CLOCK   4F26    CMC0H  = 0099   CMC0L  = 0098
CMC1H  = 009B   CMC1L  = 009A   CMC2H  = 009D   CMC2L  = 009C
CMC3H  = 009F   CMC3L  = 009E   CMINH  = 0097   CMINL  = 0096
CNTCTL= 00D7    CRCWRD= 0018    CSAVE   4F9D    CSRLH  = 0091
CSRLL  = 0090   CTCH   = 0085   CTCL   = 0084   CTSTH  = 008F
CTSTL  = 008E   CXCTH  = 0081   CXCTL  = 0080   CXINH  = 0083
CXINL  = 0082   C.     = 0001   C.AVAI= 0080   C.DP   = 0008
C.DSE  = 0010   C.DTU  = 0003   C.DVA  = 0008   C.FAIL= 00FC
C.FMT  = 0070   C.FNCT= 003E    C.GO   = 0001   C.INTC= 00FE
C.MAIN= 0020    C.NSA  = 0080   C.RCT  = 00FC   C.SER  = 0080
C.SHR  = 0040   C.SKPC= 000F    C.TAPE= 0040    C.WCS  = 0002
D      =%0002   DATACT= 00D0    DBUS    4F28    DBUSCT= 00C0
DBUSST= 00C0    DDRA   = 00D8   DDRAIN= 0010    DDRB   = 00D9
DDRBIN= 0002    DDRC   = 00DA   DDRCIN= 0001    DDRCO  = 0088
DDRCTL= 00DB    DIAFLG  4F22    DIAGPG= 4300    DIAGRM= 4F90
DIARD  = 000B   DONE1  = 0045   DONINT= 0010    DSAVE   4F9E
DSE    = 0006   D.ATH0= 0001   D.ATH1= 0002    D.EOTD= 0010
D.LAGC= 0020    D.NOTW= 0040    D.NTHR= 0004    D.TACH= 0008
D.WR4  = 0080   E      =%0003   ECCBAD= 0042   ECCCOR= 0019
ECCOK  = 0041   ECCSTA= 001A    ECCTST= 000E    EDATA  = 0095
EOTCLR= 0003    ERFLG   4F93    ERLP   = 2809   ERLPA  = 280F
ERLPB  = 2812   ERLPE  = 280C   ERNUM   4F90    ERRCNT= 00D6
ESAVE   4F9F    E.ACRC= 0010    E.AMT  = 0020   E.CDP  = 0080
E.CRC  = 0080   E.PNTR= 0008    E.RPE  = 0040   E.STEC= 0001
E.TTEC= 0002    E.UNC  = 0004   FIFORD= 006A    FORMAT  4F25
FWDTST= 0061    GCRID  = 0089   GCRSET= 0002   GOODTM= 0092
H      =%0004   HLSAVE  4FA0    IE     = 0008   INTSTA= 00E0
ITERA   4F9A    I.PWR  = 0020   I.RMPE= 0040   I5.5   = 0010
I6.5   = 0020   I7.5   = 0040   KCALL  = 005F   KCLR   = 007B
KDEP   = 003F   KENAB  = 0078   KEXAM  = 003E   KEYBRD= 00C8
KEY1   = 0078   KEY10  = 006D   KEY11  = 006E   KEY12  = 006F
KEY13  = 005C   KEY14  = 005D   KEY15  = 005E   KEY16  = 005F
KEY17  = 003C   KEY18  = 003D   KEY19  = 003E   KEY2   = 0079
KEY20  = 003F   KEY3   = 007A   KEY4   = 007B   KEY5   = 0074
KEY6   = 0075   KEY7   = 0076   KEY8   = 0077   KEY9   = 006C
KINTA  = 006F   KLDAD  = 003D   KN0    = 003C   KN1    = 005C
KN2    = 005D   KN3    = 005E   KN4    = 006C   KN5    = 006D
KN6    = 006E   KN7    = 0074   KN8    = 0075   KN9    = 0076
KU2    = 0079   KU3    = 007A   KU8    = 0077   L      =%0005
LBLANK= 000F    LCE    = 000B   LCH    = 000C   LCL     000D
```

```
LCP    = 000E          LC0    = 0000          LC1    = 0001          LC2    = 0002
LC3    = 0003          LC4    = 0004          LC5    = 0005          LC6    = 0006
LC7    = 0007          LC8    = 0008          LC9    = 0009          LDLEDA= 00CA
LDLEDB= 00CB           LDLEDC= 00CC           LDLEDD= 00CD           LDLEDE= 00CE
LDLEDF= 00CF           LKDIAG= 2800           LKKBD  = 004C          LKKEY  = 0049
LKLWMG= 0058           LKLWMP= 0055           LKLWPG= 0052           LKLWPP= 004F
LKMOD7= 0046           LKOPR  = 0046          LPFLG    4F94          LPNUM    4F92
M      =%0006          MBSEL  = 0CE0          MB.A   = 0008          MB.B   = 0004
MEMTOP= 4FFF           MINUS  = 000A          MM     = 8000          MSE    = 0008
MSGN   = C01B          MTACLR= 0000           MT.ARA= 0020           MT.CPE= 0080
MT.DSE= 0001           MT.FWD= 0040           MT.INH= 0008           MT.LWR= 0004
MT.MOT= 0002           MT.NWT= 0080           MT.PEC= 0040           MT.PSB= 0004
MT.PS0= 0001           MT.PS1= 0002           MT.REV= 0020           MT.WRT= 0010
MT.Z   = 0008          M.ATA  = 0080          M.CAPE= 0020           M.CONT= 0080
M.DEM  = 0020          M.EBL  = 0010          M.EXC  = 0008          M.FAIL= 0008
M.ILR  = 0010          M.INIT= 0010           M.OCC  = 0020          M.ONLI= 0001
M.PE   = 0040          M.PORT= 0080           M.RDEN= 0002           M.RDPE= 0008
M.RUN  = 0004          M.SCLK= 0001           M.TRA  = 0040          M.UNIT= 0007
M.WCLK= 0040           M.WCLN= 0080           M.WREN= 0080           M5.5   = 0001
M6.5   = 0002          M7.5   = 0004          NOTCAP= 0088           OKAY   = 00FF
OPRRAM= 4300           OPSTRT= 0058           OPVER  = 0040          PADCNT= 00D5
PADCRC= 0080           PDIAG  = 0048          PEID   = 008A          PENAB  = 004C
PESET  = 0001          PL     = 00B1          PRDD   = 004C          PRENF  = 009C
PS     = 00B2          PSTAT  = 0048          PSW    =%0009          P.AMTP= 0001
P.BCTC= 0040           P.CMDP= 0020           P.INTE= 0080           P.LCS  = 0040
P.LWR  = 0020          P.RDP  = 0002          P.RPEN= 0004           P.RPST= 0002
P.RPOE= 0020           P.RP1E= 0010           P.RP2E= 0020           P.RP3E= 0010
P.SING= 0080           P.STAT= 0002           P.STPE= 0080           P.TACH= 0008
P.TUPR= 0010           P.WCSP= 0004           P.WDS  = 0040          P.WFLP= 0001
P.WPEN= 0010           P.WPOE= 0008           P.WP1E= 0004           P.WP2E= 0008
P.WP3E= 0004           P.5VOK= C002           QUE    = 281B          QUEM   = 281E
RAMT   = 0010          RARA   = 0006          RARAI  = 0004          RCHBD0= 0048
RCHBD1= 0047           RCHOK  = 0046          RCHTST= 000C           RCLRT  = 000D
RCMD   = 000B          RCMLP  = 0003          RCONT  = 0080          RDATA  = 0017
RDCLK  = 0010          RDON   = 0011          READG  = 0007          REND   = 0014
REQST  = 2806          RESCHR= 00D1           REVTST= 0064           REWIND= 0004
RFIFOL= 0008           RGCLK  = 0002          RGCRI  = 0003          RIBG   = 0001
RILL   = 0012          RINST  = 000C          RMCTST= 0008           RMK2   = 0013
RNOP   = 0000          RPATH  = 0001          RPBAD  = 0044          RPCHI  = 0001
RPCLK  = 0003          RPCTL  = 0009          RPEI   = 0002          RPFAIL= 0000
RPF1   = 009D          RPF2   = 009E          RPOK   = 0043          RPOSTN= 0016
RPSTA  = 0015          RRCMT  = 000A          RSTAT  = 0002          RTIEB  = 000A
RTIER  = 0030          RTM    = 0005          RUNKI  = 0009          RUPTST= 005E
RWDUNL= 0005           R.AMT  = 0001          R.BOP  = 0008          R.DATA= 0040
R.DON  = 0002          R.DRDY= 0010           R.END  = 0010          R.ILL  = 0004
R.JVOK= 0004           R.MK2  = 0008          R.PLOD= 0008           R.PLOO= 0010
R.PLO1= 0020           R.POST= 0020           R.STNM= 0002           R.STOP= 0004
R.STPC= 0001           R.TBJN= 0080           R.TSTD= 0040           R.VOK  = 0080
R00H   = 0081          R00L   = 0080          R01H   = 0083          R01L   = 0082
R02H   = 0085          R02L   = 0084          R03H   = 0087          R03L   = 0086
R04H   = 0089          R04L   = 0088          R05H   = 008B          R05L   = 008A
R06H   = 008D          R06L   = 008C          R07H   = 008F          R07L   = 008F
R10H   = 0091          R10L   = 0090          R11H   = 0093          R11L   = 0092
R12H   = 0095          R12L   = 0094          R13H   = 0097          R13L   = 0096
```

```
R14H  = 0099          R14L  = 0098          R15H  = 009B          R15L  = 009A
R16H  = 009D          R16L  = 009C          R17H  = 009F          R17L  = 009E
R7.5  = 0010          SELCLR= 0000          SETATA= 00A1          SID   = 0080
SOD   = 0080          SOE   = 0040          SP    =%0008          SPEED   46EA
SSCLK = 0040          SSTEP = 0005          STACK = 4FFF          STATRM= 4F20
STPCT   4F20          STRSP = 5000          TADR00= 0080          TADR01= 0081
TADR02= 0082          TADR03= 0083          TADR04= 0084          TADR05= 0085
TADR06= 0086          TADR07= 0087          TADR10= 0088          TADR11= 0089
TADR12= 008A          TADR13= 008B          TAMT  = 0044          TASEL = 0080
TCMD  = 0040          TC.FWD= 0040          TC.INH= 0008          TC.LWR= 0004
TC.REV= 0020          TC.WRT= 0010          TEMP    4F99          TEST1   4300
TEST10  468C          TEST2   4386          TEST3   43FE          TEST4   4443
TEST5   44E0          TEST6   456B          TEST7   45F5          TMF   = 0099
TMRDY = 0040          TRKENA= 00D2          TSET  = 2803          TSTEND= 2818
TSTS  = 0040          TST1AC  433C  G       TST1B   4342          TST1BC  4351  G
TST1C   4357          TST1CC  436C  G       TST1L   4325          TST1W   4359
TST10A  46BF  G       TST10B  46D3          TST10L  4696          TST10R  4691
TST2AC  43A4  G       TST2BC  43BC  G       TST2CC  43D1  G       TST2D   43D7
TST2DC  43E4  G       TST2L   438B          TST3A   4407          TST3AC  4413  G
TST3B   441D          TST3BC  4429  G       TST3L   4403          TST4AC  4479  G
TST4B   447F          TST4BC  448D  G       TST4C   4493          TST4CA  44B2  G
TST4D   44B8          TST4DC  44C6  G       TST4L   445A          TST4ST  4448
TST5AC  4504  G       TST5B   450A          TST5BC  4518  G       TST5C   451E
TST5CA  453D  G       TST5D   4543          TST5DC  4551  G       TST5L   44E5
TST6AC  458F  G       TST6B   4595          TST6BC  45A3  G       TST6C   45A9
TST6CA  45C8  G       TST6D   45CE          TST6DC  45DB  G       TST6L   4570
TST7A   460F          TST7AC  4620  G       TST7B   4626          TST7BC  463A  G
TST7C   4640          TST7CC  464E  G       TST7D   4654          TST7DC  4660  G
TST7E   4666          TST7EC  4672  G       TST7L   45FA          TUSEL0= 00D1
TUSEL1= 00D2          TU78  = 0010          T.ATH0= 0001          T.ATH1= 0002
T.BOT = 0004          T.EOT = 0002          T.FPT = 0001          T.NTHR= 0004
T.ONL = 0020          T.PES = 0008          T.PSBJ= 0020          T.PSOJ= 0008
T.PSIJ= 0010          T.RDY = 0080          T.RDYO= 0040          T.RWD = 0010
T.SCLK= 0002          T10W    46A9          T4W     4456          T7W     4610
UIBG  = 00A1          VALFC   4F98          VALTB   4F95          VELTST= 005B
WDR.P = 0010          WMCCTL= 00D3          WMCERR= 00DA          WMCSTA= 00D0
WRTCLK= 0000          WRTDAT= 00D3          W.ACRC= 0004          W.CRC = 0008
W.DIAG= 0002          W.DONN= 0040          W.ECC = 0010          W.ENAB= 0080
W.ERR = 0020          W.FMT = 0070          W.GCR = 0010          W.LEFT= 0004
W.ONES= 0020          W.RESI= 0002          W.REV = 0004          W.ROME= 0010
W.RST = 0001          W.SKIP= 000F          W.WRIT= 0008          W.XFER= 0020
X     =%000A          X.DONN= 0080          X.ENAB= 0040          X.PEPE= 0002
X.ROME= 0001          X.WCLK= 0001          Y     =%000B          .     = 46EB


 ERRORS DETECTED:  0

 *WMC1.A78/PTP,WMC1=NLIST,PARAM,MACRO,LIST,WMC1
 RUN-TIME: 4 6 0 SECONDS
 CORE USED:  10K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(25) 15-OCT-80 09:12
PARAM.M80 TABLE OF CONTENTS SEQ 0049

3	"MEMORY MAP" EXTENDED ADDRESS BIT
19	READ MICRO CONTROLLER CONTROL & STATUS WORDS
139	TRANSPORT/PORT COMMAND + STATUS WORDS
380	COMMON ADDRESS SPACE REGISTERS
444	MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485	MASSBUS CONTROL WORD DEFINITIONS
557	WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709	KEYBOARD/DISPLAY CONTROL + STATUS WORD
797	MISCELLANEOUS VARIABLE DEFINITIONS
801	RAM ADDRESS DEFINITIONS
819	LINKAGE TABLE ADDRESSES
842	BIT DEFINITIONS
856	RIM AND SIM INSTRUCTION BIT DEFINITIONS
877	INTERRUPTABLE INSTRUCTION MACROS
924	DIAGNOSTIC VARIABLES
945	COMMON VARIABLE STORAGE
959	MACRO CALL DEFINITIONS
1047	TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101	8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1333	TEST 1 - WRITE DATA TRANSFER USING THE "DDR" REGISTERS - WMC IN "RIGHT" MODE
1666	TEST 2 - WRITE DATA TRANSFER USING THE 'DDR" REGISTERS - WMC IN "LEFT" MODE
2378	TEST 3 - READ DATA TRANSFER USING THE "DDR" REGISTERS - DDR IN RIGHT MODE
2589	TEST 4 - READ DATA TRANSFER USING THE "DDR" REGISTERS - DDR IN "LEFT" MODE

```
961                     ;*************************************************************************
962                     ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
963                     ;ERROR MACRO CALLS
964                     ;
965                     ;       1  -    REQUEST HOST CPU TO PRINT:
966                     ;               "BYTE/SCLK COUNT NUMBER = LLL"
967                     ;               WHERE:
968                     ;               LLL = THE VALUE STORED IN CAS REGISTER 5
969                     ;                     (THE BYTE COUNT REGISTER 16 BITS).
970                     ;
971                     ;       2  -    REQUEST HOST CPU TO PRINT:
972                     ;               DATA FORMAT = MM
973                     ;               SKIP COUNT  = NN
974                     ;               WHERE:
975                     ;               MM = DATA FORMAT FROM CAS REGISTER 2
976                     ;               NN = SKIP COUNT FROM CAS REGISTER 2
977                     ;
978                     ;       3  -    REQUEST HOST CPU TO PRINT:
979                     ;               BYTE-SCLK COUNT = LLL
980                     ;               DATA FORMAT = MM
981                     ;               SKIP COUNT  = NN
982                     ;               WHERE:
983                     ;               LLL = AS ABOVE
984                     ;               MM  = AS ABOVE
985                     ;               NN  = AS ABOVE
986                     ;
987                     ;       4  -    REQUEST HOST TO PRINT:
988                     ;               TRANSITION COUNT = LLL
989                     ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
990                     ;
991                     ;       5  -    REQUEST HOST CPU TO PRINT:
992                     ;               EXPECTED 18 BITS =      E EEEEEE
993                     ;               ACTUAL 18 BITS =        A AAAAAA
994                     ;
995                     ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
996                     ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
997                     ;               OF CAS REG 15 LOW BYTE.
998                     ;
999                     ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
1000                    ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
1001                    ;               17 LOW BYTE SIGN BIT.
1002                    ;
1003                    ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
1004                    ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
1005                    ;               AND/OR ACTUAL DATA.
1006                    ;
1007                    ;       7  -    REQUEST HOST CPU TO PRINT:
1008                    ;               "SUBGROUP NUMBER = LLL"
1009                    ;               WHERE:
1010                    ;               LLL = THE VALUE STORED IN CAS REGISTER 5
1011                    ;                     (THE BYTE COUNT REGISTER 16 BITS).
1012                    ;
1013                    ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
1014                    ;*************************************************************************
```

```
1016                    ;*************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 7'
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FPOM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ;*************************************************************************
1035                    ;*************************************************************************
1036                    ;DATA PATTERN CODES
1037
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ;*************************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  MBD1 - MASSBUS DATA TEST PART #1
                                .SBTTL  TEST 1 - WRITE DATA TRANSFER USING THE "DDR" REGISTERS - WMC IN "RIGHT" MODE
                                ;ID     MBD1-MASSBUS DATA CONTROLLER
1335  4300                      ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1336                            ;*DATA TRANSFER FROM HOST TO TM78 IN 'RIGHT' MODE
1337  4300                      SM
 (1)                            ; ***************************************************************************
 (1)                            ;*MODULE(S) UNDER TEST
 (1)                            ;*--------------------
1338                            ;*M8956, M8957, M8959
1339  4300                      SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1340                            ;*THIS TEST REQUIRES THE HOST CPU TO GENERATE THE "MBCPAR" DATA PATTERN
1341                            ;*TO SEND OVER THE MASSBUS ON A WRITE GCR COMMAND (FUNCTION CODE 63). IF
1342                            ;*THE HOST CPU IS NOT CAPABLE OF AN 18 BIT DATA TRANSFER OVER THE
1343                            ;*MASSBUS, THEN THE TEST EXPECTS A FUNCTION CODE OF 65 TO BE RETURNED!
1344                            ;*
1345                            ;*THE HOST WILL BE REQUESTED TO TRANSFER 100 BYTES (18 BIT) OF DATA USING THE
1346                            ;*10 DUMP MODE PACKING FORMAT (5 BYTES PER 10 CPU WORD). THE BYTE COUNT
1347                            ;*REQUESTED WILL APPEAR IN THE MASSBUS REGISTER 05 (BYTE COUNT REGISTER).
1348                            ;*
1349                            ;*MASSBUS REGISTER 02 WILL CONTAIN THE RECORD COUNT (SET TO 1) AND THE
1350                            ;*PACKING MODE TO USE (CODE =2 FOR 10 DUMP) - NO UNIT IS NEEDED FOR THE
1351                            ;*TEST.
1352                            ;*
1353                            ;*THIS TEST WILL ISSUE 40 "SCLK" CYCLES TO TRANSFER THE DATA FROM THE
1354                            ;*HOST CPU. DATA IS CHECKED AFTER EACH CLOCK CYCLE AND WILL CAUSE AN
1355                            ;*ERROR REQUEST TO THE HOST IF FOUND TO BE INCORRECT!
1356                            ;*
1357                            ;*DETECTION OF A MASSBUS DATA PARITY ERROR OR "OTHER" ERROR STATUS IN
1358                            ;*THE WMC ERROR STATUS WORD (100332) ALSO CAUSES AN ERROR REPORT.
1359                            ;*
1360                            ;*DATA ERRORS OR STATUS ERRORS WILL TERMINATE THE DATA TRANSFER IF HOST
1361                            ;*CPU HAS REQUESTED THE MICROCOMPUTER TO "LOOP-ON-ERROR". INTERNAL STATUS
1362                            ;*ERRORS WILL BE CLEARED IF REPORTED TO BE SET.
1363                            ;*
1364                            ;*MASSBUS DATA TEST - USE HOST CPU TO CHECK MBD MODULE OK FOR WRITE
1365                            ;*AND READ DATA TRANSFERS.
1366                            ;*WRITE XFR TEST
1367  4300                      SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1368                            ;*BGNIST.
1369                            ;*   INIT TEST - TESIX MACRO
1370                            ;*   ISSUE WMC RESTART TO INIT WRITE PATH
1371                            ;*   INIT ECODE COUNTER TO 0 AND SELECT WMC CONTROL TO 'WRITE'
1372                            ;*   SET DATA BUS CONTROL TO 0 + WAIT SMALL AMOUNT OF TIME
1373                            ;*   IF DBUS STATUS STILL=0
```

```
1374                                    ;*  :  THEN-CONTINUE
1375                                    ;*  :  ELSE-REPORT ERROR - INCORRECT DBUS
1376                                    ;*  :       STATUS - MAYBE CABLE FAILURE
1377                                    ;*  ENDIF
1378                                    ;*  INIT DDR REGS TO "IN" + REQUEST HOST CPU FOR A 'WRITE' GCR DATA XFR
1379                                    ;*  CONTINUE WITH HOST RESPONSE - CODE 63 OR 67 (IF HOST NOT CAPABLE OF 18 BIT XFR)
1380                                    ;*  DATA SENT BY HOST IS 'MBDPAR' IN 10 DUMP MODE FORMAT
1381                                    ;*  INIT 'SCLK' COUNT MASSBUS CONTROL TO 'OCC'-'RUN'-'MBRDEN'
1382                                    ;*  SET 'SCLK' + WAIT FOR HOST RESPONSE WITH 'WCLK'
1383                                    ;*  IF 'WCLK' RESPONSE BY HOST
1384                                    ;*  :  THEN-REPORT NO HOST WCLK' RESPONSE-OPI ERROR
1385                                    ;*  :  ELSE-CONTINUE
1386                                    ;*  ENDIF
1387                                    ;*  FINISH 1ST 18 BIT 'SCLK' CYCLE BY CLEARING 'SCLK' BIT FROM DBUS CONTROL
1388                                    ;*  IF 'SCLK' GO AWAY
1389                                    ;*  :  THEN-REPORT INCORRECT DBUS STATUS
1390                                    ;*  :  ELSE-CONTINUE
1391                                    ;*  ENDIF
1392                                    ;*  GET DATA IN DDR REGS 'A', 'B', + 'C' + CHECK IF DATA OK
1393                                    ;*  IF DATA RECEIVED OK
1394                                    ;*  :  THEN-REPORT READ FAILURE OF DATA FROM HOST CPU
1395                                    ;*  :  ELSE-CONTINUE
1396                                    ;*  ENDIF
1397                                    ;*  BGNDO
1398                                    ;*  :  DO UNTIL ALL DATA CLOCKED FROM HOST AND CHECKED
1399                                    ;*  :  CLOCK NEXT 18 BITS ACROSS MASSBUS
1400                                    ;*  :  IF DATA RECEIVED OK
1401                                    ;*  :  :  THEN-CONTINUE
1402                                    ;*  :  :  ELSE-REPORT DATA XFR ERROR + 'SCLK' COUNT
1403                                    ;*  :  ENDIF
1404                                    ;*  ENDDO
1405                                    ;*  DROP MASSBUS SIGNALS 'OCC' + 'RUN" TO RELEASE THE HOST CPU
1406                                    ;*ENDTST
1407  4300                              SE
 (1)                                    ; ********************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1408                                    ;*MBD1 MICRO TEST 01
1409                                    ;*MBD1 MICRO ERROR 01
1410                                    ;*MBD1-DATA WRITE TO TM78 TEST USING DIAG DATA REG.-"DDR" IN "RIGHT" MODE
1411                                    ;*M8956, M8957, M8959
1412                                    ;*DATA BUS CONTROL WORD INITED TO 0 AND WAS READ NON-0
1413                                    ;*POSSIBLE MASSBUS CABLE FAILURE
1414                                    ;*ACTUAL = NNNN
1415                                    ;*EXPECTED = NNNN
1416                                    ;*
1417                                    ;*MBD1 MICRO TEST 01
1418                                    ;*MBD1 MICRO ERROR 02
1419                                    ;*MBD1-DATA WRITE TO TM78 TEST USING DIAG DATA REG.-"DDR" IN "RIGHT" MODE
1420                                    ;*M8956, M8957, M8959
1421                                    ;*OPI ERROR - HOST CPU FAILED TO SET "RUN" ON A WRITE COMMAND (6X CODE)
1422                                    ;*TEST ABORTED!
1423                                    ;*
1424                                    ;*MBD1 MICRO TEST 01
```

```
1425                                    ;*MBD1 MICRO ERROR 03
1426                                    ;*MBD1-DATA WRITE TO TM78 TEST USING DIAG DATA REG.-"DDR" IN "RIGHT" MODE
1427                                    ;*M8956, M8957, M8959
1428                                    ;*HOST CPU FAILED TO RESPOND TO "SCLK" COMMAND WITH A "WCLK" ON THE
1429                                    ;*FIRST "SCLK" ISSUED - DATA FAILED TO TRANSFER OVER MASSBUS
1430                                    ;*ACTUAL = NNNN
1431                                    ;*EXPECTED = NNNN
1432                                    ;*TEST ABORTED!
1433                                    ;*
1434                                    ;*MBD1 MICRO TEST 01
1435                                    ;*MBD1 MICRO ERROR 04
1436                                    ;*MBD1-DATA WRITE TO TM78 TEST USING DIAG DATA REG.-"DDR" IN "RIGHT" MODE
1437                                    ;*M8956, M8957, M8959
1438                                    ;*EXPECTED CBUSSTA BYTE NOT = ACTUAL
1439                                    ;*"SCLK" FAILED TO CLEAR AFTER IT WAS SET.
1440                                    ;*ACTUAL = NNNN
1441                                    ;*EXPECTED = NNNN
1442                                    ;*TEST ABORTED!
1443                                    ;*
1444                                    ;*MBD1 MICRO TEST 01
1445                                    ;*MBD1 MICRO ERROR 05
1446                                    ;*MBD1-DATA WRITE TO TM78 TEST USING DIAG DATA REG.-"DDR" IN "RIGHT" MODE
1447                                    ;*M8956, M8957, M8959
1448                                    ;*MBD BITS 7-0 COMPARE ERROR ON THIS "SCLK" XFR
1449                                    ;*ACTUAL = NNNN
1450                                    ;*EXPECTED = NNNN
1451                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1452                                    ;*EXPECTED 18 BITS = E EEEEEE
1453                                    ;*ACTUAL 18 BITS   = A AAAAAA
1454                                    ;*TEST ABORTED!
1455                                    ;*
1456                                    ;*MBD1 MICRO TEST 01
1457                                    ;*MBD1 MICRO ERROR 06
1458                                    ;*MBD1-DATA WRITE TO TM78 TEST USING DIAG DATA REG.-"DDR" IN "RIGHT" MODE
1459                                    ;*M8956, M8957, M8959
1460                                    ;*MBD BITS 15-8 COMPARE ERROR ON THIS "SCLK" XFR
1461                                    ;*ACTUAL = NNNN
1462                                    ;*EXPECTED = NNNN
1463                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1464                                    ;*EXPECTED 18 BITS = E EEEEEE
1465                                    ;*ACTUAL 18 BITS   = A AAAAAA
1466                                    ;*
1467                                    ;*MBD1 MICRO TEST 01
1468                                    ;*MBD1 MICRO ERROR 07
1469                                    ;*MBD1-DATA WRITE TO TM78 TEST USING DIAG DATA REG.-"DDR" IN "RIGHT" MODE
1470                                    ;*M8956, M8957, M8959
1471                                    ;*MBD BITS 17-16 AND "P" INCORRECT OR STATUS ERROR DETECTED (BITS 7-3
1472                                    ;*OF WMC ERROR WORD - 100332).
1473                                    ;*ACTUAL = NNNN
1474                                    ;*EXPECTED = NNNN
1475                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1476                                    ;*EXPECTED 18 BITS = E EEEEEE
1477                                    ;*ACTUAL 18 BITS   = A AAAAAA
1478                                    ;*
```

```
                                        ;*MBD1 MICRO TEST 01
                                        ;*MBD1 MICRO ERROR 10
                                        ;*MBD1-DATA WRITE TO TM78 TEST USING DIAG DATA REG.-"DDR" IN "RIGHT" MODE
                                        ;*M8956, M8957, M8959
                                        ;*MASSBUS PARITY ERROR DETECTED BY THE TM78
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*TEST ABORTED!
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*EXPECTED 18 BITS = E EEEEEE
                                        ;*ACTUAL 18 BITS   = A AAAAAA
1490  4300                              S
 (1)                                    ; ************************************************************************************
1491
1492  4300                              TEST1:  TESTX   @1
 (1)  4300  3E  01            7.0                MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                CALL    TSET            ;SETUP THE TEST
1493                                    ;%MBD1-DATA WRITE TO TM78 TEST USING DIAG. DATA REG.-"DDR" IN "RIGHT" MODE
1494                                    ;&M8956, M8957, M8959
1495                                    ;DISABLE THE MICROCOMPUTER INTERRUPT SYSTEM
1496
1497  4305  AF                    4.0   TST1L:  XRA     A               ;SET THE SYSTEM CLOCK TO 'NORMAL'
1498  4306  D3  F0              10.0           OUT     CLKCTL          ;WRITE THE CLOCK CONTROL WORD
1499
1500  4308  3E  01               7.0           MVI     A,W.RST         ;RESTART THE WMC
1501  430A  D3  D3              10.0           OUT     WMCCTL
1502  430C  AF                    4.0         XRA     A               ;FINISH THE RESTART CYCLE
1503  430D  D3  D3              10.0           OUT     WMCCTL          ;DROP THE RST BIT IN WMC CONTROL WORD
1504  430F                                    INICNT                  ;INIT THE COUNTERS
 (1)  430F  3E  30               7.0                MVI     A,@060          ;INIT THE
 (1)  4311  D3  D7              10.0                OUT     CNTCTL          ;BYTE COUNTER
 (1)  4313  3E  70               7.0                MVI     A,@160          ;INIT THE
 (1)  4315  D3  D7              10.0                OUT     CNTCTL          ;PAD COUNTER
 (1)  4317  3E  B0               7.0                MVI     A,@260          ;INIT THE
 (1)  4319  D3  D7              10.0                OUT     CNTCTL          ;ERROR COUNTER
1505  431B                              CLRECT
 (1)  431B  AF                    4.0                XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  431C  D3  D6              10.0                OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  431E  D3  D6              10.0                OUT     ERRCNT          ;CLEAR BITS 15-8
1506  4320  3E  08               7.0   MVI     A,W.WRITE               ;ENABLE ONLY "WRITE" IN WMC CNTRL WORD
1507  4322  D3  D3              10.0   OUT     WMCCTL                  ;SET WRITE STATUS
1508
1509  4324  3E  00               7.0   MVI     A,0                     ;AGAIN FOR THE LOOP
1510  4326                              ROUT    EDATA                   ;SAVE "EDATA" ON STATUS CHECK
 (1)  4326  D3  95              10.0           OUT     EDATA                   ;WRITE AC INTO EDATA
 (1)  4328  7F                    4.0           MOV     A,A                     ;RETRY LINK
1511  4329  D3  C0              10.0   OUT     DBUSCTL                 ;CLEAR THE DBUS CNTRL WORD
1512  432B  00                    4.0   NOP                             ;SMALL DELAY
1513  432C  DB  C0              10.0   IN      DBUSSTA                 ;GET THE DBUS STATUS
1514  432E  A7                    4.0   ANA     A                       ;SET THE CONDITION BITS
1515  432F                              ROUT    ADATA                   ;SAVE THE "ADATA" IN CASE OF ERROR
 (1)  432F  D3  94              10.0           OUT     ADATA                   ;WRITE AC INTO ADATA
 (1)  4331  7F                    4.0           MOV     A,A                     ;RETRY LINK
1516  4332  CA  3A  43          10.0   JZ      TST1AC                  ;JUMP OVER IF NO ERROR - SHOULD BE -0
```

```
1517  4335                                        ERRB    TST1L,TST1AC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  4335  CD  12  28        18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0001                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4338  01                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4339  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  433A  CD  16  28        18.0        TST1AC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  433D  DA  05  43        10.0                JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1518                                    ;>DATA BUS CONTROL WORD INITED TO 0 AND WAS READ NON-0
1519                                    ;>POSSIBLE MASSBUS CABLE FAILURE
1520                                    ;STATUS WAS OK - PROCEED TO SETUP THE DDR REGISTERS (CLEAR THEM)
1521
1522  4340  3E  9B          7.0   TST1B:  MVI     A,@233          ;SET DDR CNTRL WORD TO ''IN''
1523  4342  D3  DB         10.0           OUT     DDRCTL
1524                                    ;MASSBUS REGISTERS 02 (RECORD COUNT, FMT TYPE, SKIP CNT) AND REGISTER
1525                                    ;05 (BYTE COUNT REG) WILL BE SET UP ON THE DIAGNOSTIC MONITOR CALL.
1526
1527                                    ;REQUEST THE DIAG. MON. TO SIGNAL THE HOST CPU FOR A WRITE GCR DATA
1528                                    ;TRANSFER USING ''MBDPAR'' DATA PATTERN (2ND ARG IN CALL). BYTE COUNT
1529                                    ;IS 100. (40 SCLK * 2.5 BYTES PER WORD IN 10 DUMP MODE). REG 02 GETS
1530                                    ;LOADED TO 1 RECORD @ 10 DUMP FORMAT.
1531
1532  4344                                          REQ     2,1,40,@30004
 (1)  4344  CD  06  28        18.0                CALL    REQST
 (1)  4347  01                                    .BYTE   1               ;DATA PATTERN NUMBER
 (1)  4348  00  28                                .WORD   40              ;SYSTEM ''40'' COUNT
 (1)  434A  30  04                                .WORD   @30004          ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  434C  00                                    .BYTE                   ;DATA COMPARE FLAG IF =1
 (1)  434D  02                                    .BYTE   2               ;REQUEST CODE
1533
1534                                    ;INIT THE SCLK COUNTER TO 0 ... TEST WILL STOP WHEN COUNT = 41.
1535
1536  434E  3E  01          7.0           MVI     A,1
1537  4350  32  53  46    13.0           STA     CLKCNT          ;INIT THE CLOCK COUNTER
1538
1539                                    ;SET THE DBUS CNTRL WORD TO ''OCC'', AND ''SCLK OUT''
1540
1541  4353  3E  22          7.0           MVI     A,M.OCC!T.SCLK
1542  4355  D3  C0         10.0           OUT     DBUSCTL         ;SET THE STATUS - NOW HAVE MASSBUS
1543                                    ;NOW WAIT FOR HOST CPU TO ASSERT ''RUN'' ON THE MASSBUS...COULD TAKE UP TO
1544                                    ;10 MILLISEC (MASSBUS SPEC.)
1545
1546  4357  0E  14          7.0           MVI     C,20            ;LOOK FOR 100 M-SEC.
1547  4359  DB  C0         10.0   T1W:    IN      DBUSSTA         ;GET DBUS STATUS - CHECK FOR ''RUN''
1548  435B  E6  04          7.0           ANI     M.RUN           ;SAVE ONLY ''RUN'' BIT
1549  435D  C2  7B  43     10.0           JNZ     TST1C           ;JUMP IF RUN IS UP - OK TO PROCEED
1550  4360  3E  FF          7.0   T1WL:   MVI     A,@377          ;GET A DELAY TIME
1551  4362  3D              4.0   T1WL1:  DCR     A               ;DECREMENT
1552  4363  C2  62  43     10.0           JNZ     T1WL1           ;STAY IN LOOP TILL =0
1553
1554                                    ;DOWNCOUNT GROSS TIMER COUNT
1555
1556  4366  0D              4.0           DCR     C
1557  4367  C2  59  43     10.0           JNZ     T1W             ;THEN CHECK IF ''RUN'' IS UP
```

```
1558  436A  CD  80  45      18.0          CALL    QUIT
1559  436D                                ERR     TSTEND,T1WLC
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  436D  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4370  02                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4371  00                           .BYTE
 (1)  4372  CD  15  28      18.0          T1WLC:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4375  DA  18  28      10.0                  JC      TSTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
1560                                ;>OPI ERROR - HOST CPU FAILED TO SET "RUN" ON A WRITE COMMAND (6X CODE)
1561                                ;<TEST ABORTED!
1562  4378  C3  18  28      10.0          JMP     TSTEND          ;FATAL ERROR! - QUIT NOW
1563                                ;START THE CLOCK CYCLE (SCLK) BY SETTING THE SCLK BIT IN DBUS CNTRL
1564                                ;WORD
1565
1566  437B  3E  23           7.0  TST1C:  MVI     A,M.OCC!T.SCLK!M.SCLK  ;"OCC" + "SCLK OUT" + "SCLK"
1567  437D  D3  C0          10.0          OUT     DBUSCTL         ;SET SCLK BIT
1568  437F  F6  44           7.0          ORI     M.RUN!M.WCLK    ;ADD IN OTHER EXPECTED BITS
1569  4381                               ROUT    EDATA           ;SAVE FOR ERROR REPORTING
 (1)  4381  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4383  7F               4.0                  MOV     A,A             ;RETRY LINK
1570  4384  00               4.0          NOP
1571  4385  DB  C0          10.0          IN      DBUSSTA         ;GET THE DBUS STATUS - CHECK FOR WCLK
1572  4387                               ROUT    ADATA           ;SAVE ACTUAL DATA READ
 (1)  4387  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4389  7F               4.0                  MOV     A,A             ;RETRY LINK
1573  438A  E6  40           7.0          ANI     M.WCLK          ;SAVE ONLY "WCLK" BIT
1574  438C  C2  A0  43      10.0          JNZ     TST1CB          ;JUMP OVER IF SET - NO ERROR DETECTED
1575  438F  CD  80  45      18.0          CALL    QUIT            ;STOP THE XFR IN PROGRESS!
1576  4392                               ERRB    TST1L,TST1CA
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4392  CD  12  28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0003                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4395  03                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4396  00                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4397  CD  15  28      18.0          TST1CA::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  439A  DA  05  43      10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1577                                ;>HOST CPU FAILED TO RESPOND TO "SCLK" COMMAND WITH A "WCLK" ON THE
1578                                ;>FIRST "SCLK" ISSUED - DATA FAILED TO TRANSFER OVER MASSBUS
1579                                ;<TEST ABORTED!
1580  439D  C3  18  28      10.0          JMP     TSTEND          ;ABORT NOW IF PROCEED ON ERROR SET
1581                                ;FINISH THE "SCLK" CYCLE BY CLEARING THE "SCLK" BIT IN DBUS CNTRL WORD
1582
1583  43A0  CD  95  46      18.0  TST1CB: CALL    GETDDR          ;PULL DATA OFF MASSBUS VIA DDR REGS
1584
1585  43A3  3E  22           7.0          MVI     A,M.OCC!T.SCLK  ;SAME STATUS MINUS THE "SCLK" BIT
1586  43A5  D3  C0          10.0          OUT     DBUSCTL         ;SET STATUS
1587  43A7  F6  04           7.0          ORI     M.RUN           ;ADD IN OTHER EXPECTED BITS
1588  43A9                               ROUT    EDATA           ;SAVE THE EXPECTED DATA
 (1)  43A9  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  43AB  7F               4.0                  MOV     A,A             ;RETRY LINK
1589  43AC  00               4.0          NOP                     ;SMALL WAIT
1590  43AD  DB  C0          10.0          IN      DBUSSTA         ;GET THE DBUS STATUS
1591  43AF                               ROUT    ADATA           ;SAVE ACTUAL DATA READ
```

```
  (1)  43AF  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  43B1  7F               4.0                MOV     A,A             ;RETRY LINK
1592   43B2  E6  01           7.0        ANI     M.SCLK          ;SAVE ONLY "SCLK" STATUS BIT
1593   43B4  CA  C8  43       10.0       JZ      T1CHK           ;JUMP OVER IF THE BIT IS CLEARED
1594   43B7  CD  80  45       18.0       CALL    QUIT            ;ABORT XFR IF ERROR DETECTED!
1595   43BA                               ERRB    TST1L,TST1CC
  (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  43BA  CD  12  28       18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0004                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  43BD  04                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  43BE  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  43BF  CD  15  28       18.0                TST1CC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  43C2  DA  05  43       10.0                        JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1596                                       ;>EXPECTED CBUSSTA BYTE NOT = ACTUAL
1597                                       ;>"SCLK" FAILED TO CLEAR AFTER IT WAS SET.
1598                                       ;<TEST ABORTED!
1599   43C5  C3  18  28       10.0                JMP     TSTEND
1600                                       ;DATA NOW TRANSFERED IN AN 18 BIT CHUNK FROM THE HOST CPU.
1601                                       ;CHECK THE DDR REGISTERS FOR EXPECTED RESULTS. REG "B" CONTAINS
1602                                       ;THE "SCLK" COUNT -1 (USED FOR A TABLE POINTER)
1603
1604   43C8  AF                    4.0    T1CHK:  XRA     A               ;CLEAR CAS REG 05 HIGH BYTE
1605   43C9                               ROUT    R05H            ;FOR ERROR PRINTING
  (1)  43C9  D3  8B           10.0                OUT     R05H            ;WRITE AC INTO R05H
  (1)  43CB  7F                4.0                MOV     A,A             ;RETRY LINK
1606   43CC  3A  53  46       13.0       LDA     CLKCNT          ;GET THE CLOCK COUNT
1607   43CF                               ROUT    R05L            ;SAVE IT IN REG 05 LOW BYTE FOR ERROR REPORT
  (1)  43CF  D3  8A           10.0                OUT     R05L            ;WRITE AC INTO R05L
  (1)  43D1  7F                4.0                MOV     A,A             ;RETRY LINK
1608   43D2  CD  D4  45       18.0       CALL    LD18BIT         ;GET THE EXPECTED 18 BIT DATA
1609   43D5  CD  AC  45       18.0       CALL    DDRACK          ;CHECK BITS 7 TO 0 (REG A)
1610   43D8  CA  EC  43       10.0       JZ      TST1D           ;JUMP IF NO ERROR DETECTED
1611   43DB  CD  80  45       18.0       CALL    QUIT            ;ABORT THE XFR NOW!
1612   43DE                               ERRB    TST1L,TST1CD,6
  (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  43DE  CD  12  28       18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0005                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  43E1  05                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  43E2  06                                           .BYTE   6               ;PRINT ROUTINE NUMBER
  (1)  43E3  CD  15  28       18.0                TST1CD::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  43E6  DA  05  43       10.0                        JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1613                                       ;>MBD BITS 7-0 COMPARE ERROR ON THIS "SCLK" XFR
1614                                       ;<TEST ABORTED!
1615   43E9  C3  18  28       10.0                JMP     TSTEND
1616
1617                                       ;CHECK DATA IN BITS 15 TO 8 OF DDR (REG B)
1618
1619   43EC  CD  C3  45       18.0   TST1D:  CALL    DDRBCK          ;CHECK BITS 15 TO 8
1620   43EF  CA  03  44       10.0       JZ      TST1E           ;JUMP IF ALL OK
1621   43F2  CD  80  45       18.0       CALL    QUIT            ;ABORT THE XFR NOW
```

```
1623  43F5                                    ERRB    TST1L,TST1DC,6
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  43F5  CD  12  28        18.0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0006                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43F8  06                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43F9  06                                         .BYTE   6               ;PRINT ROUTINE NUMBER
 (1)  43FA  CD  15  28        18.0               TST1DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43FD  DA  05  43        10.0                       JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1624                                    ;>MBD BITS 15-8 COMPARE ERROR ON THIS "SCLK" XFR
1625  4400  C3  18  28        10.0              JMP     TSTEND
1626                                    ;NOW TEST DATA IN DDR REG "C" - PARITY BIT AND MASSBUS DATA BITS 17-16
1627                                    ;AND THE PARITY ERROR BITS.
1628
1629  4403  CD  FD  45        18.0  TST1E:    CALL    DDRCCK          ;CHECK DDR REG C
1630  4406  CA  1A  44        10.0            JZ      TST1F           ;JUMP IF NO ERROR DETECTED
1631  4409  CD  A1  45        18.0            CALL    QUIT1           ;ABORT WRITE XFR
1632  440C                                    ERRB    TST1L,TST1EC,6
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  440C  CD  12  28        18.0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0007                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  440F  07                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4410  06                                         .BYTE   6               ;PRINT ROUTINE NUMBER
 (1)  4411  CD  15  28        18.0               TST1EC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4414  DA  05  43        10.0                       JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1633                                    ·>MBD BITS 17-16 AND "P" INCORRECT OR STATUS ERROR DETECTED (BITS 7-3
1634                                    ;>OF WMC ERROR WORD - 100332).
1635  4417  C3  18  28        10.0              JMP     TSTEND          ;QUIT!
1636
1637  441A  DB  DA              10.0  TST1F:    IN      WMCERR          ;GET THE WMC ERROR BYTE
1638  441C  E6  40               7.0            ANI     M.PE            ;MASS BUSS PARITY ERROR?
1639  441E  CA  32  44          10.0            JZ      TST1G           ;NO - CONTINUE
1640  4421  CD  A1  45          18.0            CALL    QUIT1           ;TERMINATE THE MASS BUS TRANSFER
1641  4424                                    ERR     TST1L,TST1FC,6
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4424  CD  09  28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0008                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4427  08                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4428  06                                         .BYTE   6
 (1)  4429  CD  15  28        18.0               TST1FC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  442C  DA  05  43        10.0                       JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1642                                    ;>MASSBUS PARITY ERROR DETECTED BY THE TM78
1643                                    ;<TEST ABORTED!
1644  442F  C3  18  28        10.0              JMP     TSTEND          ;ABORT THE TEST SEQUENCE
1645                                    ;HERE WHEN TIME TO DO NEXT "SCLK" CYCLE
1646  4432  3A  53  46        13.0  TST1G:    LDA     CLKCNT          ;GET CURRENT CLOCK COUNT
1647  4435  3C                  4.0            INR     A               ;UPDATE IT
1648  4436  32  53  46        13.0            STA     CLKCNT          ;SAVE THE UPDATED COUNTED
1649  4439  FE  29               7.0            CPI     41              ;DONE WITH THE TEST? (TOTAL OF 40 SCLKS)
1650  443B  CA  44  44          10.0            JZ      T1END           ;YES - TERMINATE TRANSFER AND EXIT
1651
1652  443E  CD  68  46          18.0            CALL    CLOCKO          ;NO - ISSUE NEXT "SCLK" CYCLE
1653  4441  C3  C8  43          10.0            JMP     T1CHK           ;AND CHECK THE DATA IN DDR REGISTERS
```

```
1655                                    ;HERE WHEN THE TRANSFER IS COMPLETE - TERMINATE THE TRANSFER BY ISSUING
1656                                    ;MASSBUS "EBL"
1657
1658  4444  3E  32          7.0  T1END:  MVI   A,M.OCC!M.EBL!T.SCLK  ;GET "OCC" + "EBL" + "SCLK OUT"
1659  4446  D3  C0         10.0          OUT   DBUSCTL               ;SET THE STATUS - TERMINATE THE TRANSFER
1660  4448  3E  22          7.0          MVI   A,M.OCC!T.SCLK        ;DROP "EBL"
1661  444A  D3  C0         10.0          OUT   DBUSCTL
1662  444C  3E  00          7.0          MVI   A,@0                  ;CLEAR DBUS CONTROL - ALL DONE WITH MASSBUS
1663  444E  D3  C0         10.0          OUT   DBUSCTL               ;RELEASE THE MASSBUS DATA LINES
1664  4450                              ENDTST TST1L
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4450                              REQ   7                     ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4450  CD  06  28       18.0                        CALL  REQST
 (2)  4453  00                                           .BYTE                 ;DATA PATTERN NUMBER
 (2)  4454  00  00                                       .WORD                 ;SYSTEM "" COUNT
 (2)  4456  00  00                                       .WORD                 ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4458  00                                           .BYTE                 ;DATA COMPARE FLAG IF =1
 (2)  4459  07                                           .BYTE  7              ;REQUEST CODE
 (1)  445A  3A  9A  4F       13.0          LDA   ITERA                 ;GET ITERATION COUNT
 (1)  445D  3D                4.0          DCR   A                     ;DOWNCOUNT
 (1)  445E  32  9A  4F       13.0          STA   ITERA                 ;SAVE COUNT
 (1)  4461  F2  05  43       10.0          JP    TST1L                 ;DO TEST UNTIL TILL = 0
```

```
                        .SBTTL  TEST 2 - WRITE DATA TRANSFER USING THE "DDR" REGISTERS - WMC IN "LEFT" MODE
4464                    ST
                        ;*********************************************************************
                        ;*TEST TITLE
                        ;*---- -----
                        ;*WRITE DATA TRANSFER USING THE "DDR" REGISTERS - WMC IN "LEFT" MODE
4464                    SM
                        ;*********************************************************************
                        ;*MODULE(S) UNDER TEST
                        ;*--------------------
                        ;*M8956 & M8959
4464                    SD
                        ;*********************************************************************
                        ;*DESCRIPTION
                        ;*-----------
                        ;*THIS TEST REQUIRES THE HOST CPU TO GENERATE THE "MBCPAR" DATA PATTERN
                        ;*TO SEND OVER THE MASSBUS ON A WRITE GCR COMMAND (FUNCTION CODE 63). IF
                        ;*THE HOST CPU IS NOT CAPABLE OF AN 18 BIT DATA TRANSFER OVER THE
                        ;*MASSBUS, THEN THE TEST EXPECTS A FUNCTION CODE OF 65 TO BE RETURNED!
                        ;*
                        ;*THE HOST WILL BE REQUESTED TO TRANSFER 100 BYTES (18 BIT) OF DATA USING THE
                        ;*10 DUMP MODE PACKING FORMAT (5 BYTES PER 10 CPU WORD). THE BYTE COUNT
                        ;*REQUESTED WILL APPEAR IN THE MASSBUS REGISTER 05 (BYTE COUNT REGISTER).
                        ;*
                        ;*MASSBUS REGISTER 02 WILL CONTAIN THE RECORD COUNT (SET TO 1) AND THE
                        ;*PACKING MODE TO USE (CODE =2 FOR 10 DUMP) - NO UNIT IS NEEDED FOR THE
                        ;*TEST.
                        ;*
                        ;*THIS TEST WILL ISSUE 40 "SCLK" CYCLES TO TRANSFER THE DATA FROM THE
                        ;*HOST CPU. DATA IS CHECKED AFTER EACH CLOCK CYCLE AND WILL CAUSE AN
                        ;*ERROR REQUEST TO THE HOST IF FOUND TO BE INCORRECT!
                        ;*
                        ;*DETECTION OF A MASSBUS DATA PARITY ERROR OR "OTHER" ERROR STATUS IN
                        ;*THE WMC ERROR STATUS WORD (100332) ALSO CAUSES AN ERROR REPORT.
                        ;*
                        ;*DATA ERRORS OR STATUS ERRORS WILL TERMINATE THE DATA TRANSFER IF HOST
                        ;*CPU HAS REQUESTED THE MICROCOMPUTER TO "LOOP-ON-ERROR". INTERNAL STATUS
                        ;*ERRORS WILL BE CLEARED IF REPORTED TO BE SET.
                        ;*
                        ;*MASSBUS DATA TEST - USE HOST CPU TO CHECK MBD MODULE OK FOR WRITE
                        ;*AND READ DATA TRANSFERS.
                        ;*WRITE XFR TEST
4464                    SP
                        ;*********************************************************************
                        ;*PROCEDURE
                        ;*---------
                        ;*BGNTST
                        ;*  INIT TEST - TESTX MACRO
                        ;*  SET WMC 'LEFT' MODE TO TEST EXTRA BITS
                        ;*  ISSUE WMC RESTART TO INIT WRITE PATH
                        ;*  IF WMC 'LEFT' SET
                        ;*  :  THEN-CONTINUE
                        ;*  :  ELSE-REPORT ERROR
                        ;*  ENDIF
```

```
                                ;*  ISSUE WMC RESTART TO INIT WRITE PATH
                                ;*  INIT ECODE COUNTER TO 0 AND SELECT WMC CONTROL TO 'WRITE'
                                ;*  SET DATA BUS CONTROL TO 0 + WAIT SMALL AMOUNT OF TIME
                                ;*  IF DBUS STATUS STILL=0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT ERROR - INCORRECT DBUS
                                ;*  :       STATUS - MAYBE CABLE FAILURE
                                ;*  ENDIF
                                ;*  INIT DDR REGS TO "IN" + REQUEST HOST CPU FOR A 'WRITE' GCR DATA XFR
                                ;*  CONTINUE WITH HOST RESPONSE - CODE 63 OR 67 (IF HOST NOT CAPABLE OF 18 BIT XFR)
                                ;*  DATA SENT BY HOST IS 'MBDPAR' IN 10 DUMP MODE FORMAT
                                ;*  INIT 'SCLK' COUNT MASSBUS CONTROL TO 'OCC'-'RUN'-'MBRDEN'
                                ;*  SET 'SCLK' + WAIT FOR HOST RESPONSE WITH 'WCLK'
                                ;*  IF 'WCLK' RESPONSE BY HOST
                                ;*  :  THEN-REPORT NO HOST WCLK' RESPONSE-OPI ERROR
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  FINISH 1ST 18 BIT 'SCLK' CYCLE BY CLEARING 'SCLK' BIT FROM DBUS CONTROL
                                ;*  IF 'SCLK' GO AWAY
                                ;*  :  THEN-REPORT INCORRECT DBUS STATUS
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  GET DATA IN DDR REGS 'A', 'B', + 'C' + CHECK IF DATA OK
                                ;*  IF DATA RECEIVED OK
                                ;*  :  THEN-REPORT READ FAILURE OF DATA FROM HOST CPU
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  BGNDO
                                ;*  :  DO UNTIL ALL DATA CLOCKED FROM HOST AND CHECKED
                                ;*  :  CLOCK NEXT 18 BITS ACROSS MASSBUS
                                ;*  :  IF DATA RECEIVED OK
                                ;*  :  :  THEN-CONTINUE
                                ;*  :  :  ELSE-REPORT DATA XFR ERROR + 'SCLK' COUNT
                                ;*  :  ENDIF
                                ;*  ENDDO
                                ;*  DROP MASSBUS SIGNALS 'OCC' + 'RUN" TO RELEASE THE HOST CPU
                                ;*ENDTST
4464                            SE
 (1)                            ; ********************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MBD1 MICRO TEST 02
                                ;*MBD1 MICRO ERROR 11
                                ;*MBD1-DATA WRITE TM78 TEST USING DIAG DATA REG.-"DDR" IN "LEFT" MODE
                                ;*M8956, M8959
                                ;*WMC FAILED TO SET "WMC LEFT" STATUS BIT FOR DDR CONTROL NOT SET
                                ;*ACTUAL = NNNN
                                ;*TEST ABORTED!
                                ;*
                                ;*MBD1 MICRO TEST 02
                                ;*MBD1 MICRO ERROR 12
                                ;*MBD1-DATA WRITE TM78 TEST USING DIAG DATA REG.-"DDR" IN "LEFT" MODE
                                ;*M8956, M8959
                                ;*OPI ERROR - HOST CPU FAILED TO SET "RUN" ON A WRITE COMMAND (6X CODE)
```

```
1759                                    ;*TEST ABORTED!
1760                                    ;*
1761                                    ;*MBD1 MICRO TEST 02
1762                                    ;*MBD1 MICRO ERROR 13
1763                                    ;*MBD1-DATA WRITE TM78 TEST USING DIAG DATA REG.-"DDR" IN "LEFT" MODE
1764                                    ;*M8956, M8959
1765                                    ;*MBD BITS 7-0 COMPARE ERROR ON THIS "SCLK" XFR
1766                                    ;*ACTUAL = NNNN
1767                                    ;*EXPECTED = NNNN
1768                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1769                                    ;*EXPECTED 18 BITS = E EEEEEE
1770                                    ;*ACTUAL 18 BITS   = A AAAAAA
1771                                    ;*
1772                                    ;*MBD1 MICRO TEST 02
1773                                    ;*MBD1 MICRO ERROR 14
1774                                    ;*MBD1-DATA WRITE TM78 TEST USING DIAG DATA REG.-"DDR" IN "LEFT" MODE
1775                                    ;*M8956, M8959
1776                                    ;*MBD BITS 15-8 COMPARE ERROR ON THIS "SCLK" XFR
1777                                    ;*ACTUAL = NNNN
1778                                    ;*EXPECTED = NNNN
1779                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1780                                    ;*EXPECTED 18 BITS = E EEEEEE
1781                                    ;*ACTUAL 18 BITS   = A AAAAAA
1782                                    ;*TEST ABORTED!
1783                                    ;*
1784                                    ;*MBD1 MICRO TEST 02
1785                                    ;*MBD1 MICRO ERROR 15
1786                                    ;*MBD1-DATA WRITE TM78 TEST USING DIAG DATA REG.-"DDR" IN "LEFT" MODE
1787                                    ;*M8956, M8959
1788                                    ;*MBD BITS 17-16 AND "P" INCORRECT OR STATUS ERROR DETECTED (BITS 7-3
1789                                    ;*OF WMC ERROR WORD - 100332).
1790                                    ;*ACTUAL = NNNN
1791                                    ;*EXPECTED = NNNN
1792                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1793                                    ;*EXPECTED 18 BITS = E EEEEEE
1794                                    ;*ACTUAL 18 BITS   = A AAAAAA
1795                                    ;*
1796                                    ;*MBD1 MICRO TEST 02
1797                                    ;*MBD1 MICRO ERROR 16
1798                                    ;*MBD1-DATA WRITE TM78 TEST USING DIAG DATA REG.-"DDR" IN "LEFT" MODE
1799                                    ;*M8956, M8959
1800                                    ;*MASSBUS PARITY ERROR DETECTED BY THE TM78
1801                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1802                                    ;*EXPECTED 18 BITS = E EEEEEE
1803                                    ;*ACTUAL 18 BITS   = A AAAAAA
1804                                    ;*TEST ABORTED!
1805  4464                              $
 (1)                                    ; ************************************************************************
1806  4464                              TEST2:  TESTX   @2
 (1)  4464  3E  02            7.0                 MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4466  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
1807                                    ;%MBD1-DATA WRITE TM78 TEST USING DIAG. DATA REG.-"DDR" IN "LEFT" MODE
1808                                    ;&M8956, M8959
1809
```

```
                                        ;NOW SET THE WMC "LEFT" BIT
                                        
4469  3E  88          7.0   TST2L:  MVI     A,@210          ;PUT DDR IN THE "IN" MODE
446B  D3  DB         10.0           OUT     DDRCTL
446D  3E  39          7.0           MVI     A,@71           ;PUT A FMT TYPE OF "10 DUMP" & SKP CNT=11
446F  D3  D0         10.0           OUT     DATACTL         ;IN CONTROL REG FOR WMC PGM
4471  3E  01          7.0           MVI     A,W.RST         ;RESTART THE WMC
4473  D3  D3         10.0           OUT     WMCCTL
                                        
                                        ;CLEAR THE ERROR CODE COUNTER
                                        
4475                                    CLRECT
 (1)  4475  AF          4.0                           XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  4476  D3  D6     10.0                           OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  4478  D3  D6     10.0                           OUT     ERRCNT          ;CLEAR BITS 15-8
447A  AF          4.0           XRA     A               ;FINISH THE RESTART CYCLE
447B  D3  D3         10.0           OUT     WMCCTL
447D  3E  80          7.0           MVI     A,W.ENAB        ;THEN TELL WMC TO SET "LEFT"
447F  D3  D3         10.0           OUT     WMCCTL
                                        
4481  3E  28          7.0           MVI     A,40
4483  3D             4.0   T2WS:   DCR     A               ;WAIT FOR ABOUT 100 MICRO-SEC TO SET
4484  C2  83  44     10.0           JNZ     T2WS            ;DELAY TILL TIMEOUT
                                        
                                        :NOW CHECK THAT THE WMC PGM REALLY SET "WMC LEFT" BIT SO LOADING DDRA-DDRC
                                        ;WILL BE DIRECT TRANSLATION TO MASSBUS BITS.
                                        
4487  DB  A1         10.0           IN      CBUSSTA         ;GET CBUS STATUS (WMC LEFT BIT)
4489                                  ROUT    ADATA           ;SAVE ACTUAL STATUS FOR ERROR REPORT
 (1)  4489  D3  94     10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  448B  7F         4.0                    MOV     A,A             ;RETRY LINK
448C  E6  04          7.0           ANI     W.LEFT          ;SAVE ONLY WMC LEFT BIT
448E  C2  9F  44     10.0           JNZ     TST2GO          ;CONTINUE IF ITS SET...
4491                                  ERRA    TST2L,TST2LC
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4491  CD  0F  28 18.0                    CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0009                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4494  09                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4495  00                                .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4496  CD  15  28 18.0            TST2LC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4499  DA  69  44 10.0                    JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>WMC FAILED TO SET "WMC LEFT" STATUS BIT FOR DDR CONTROL NOT SET
                                        ;<TEST ABORTED!
449C  C3  18  28     10.0           JMP     TSTEND          ;ABORT RIGHT NOW!
449F  3E  08          7.0   TST2GO: MVI     A,W.WRITE       ;ENABLE ONLY "WRITE" IN WMC CNTRL WORD
44A1  D3  D3         10.0           OUT     WMCCTL          ;SET WRITE STATUS
                                        
                                        ;PROCEED TO SETUP THE DDR REGISTERS (CLEAR THEM)
                                        
44A3  3E  9B          7.0   TST2B:  MVI     A,@233          ;SET DDR CNTRL WORD TO "IN"
44A5  D3  DB         10.0           OUT     DDRCTL
                                        ;MASSBUS REGISTERS 02 (RECORD COUNT, FMT TYPE, SKIP CNT) AND REGISTER
                                        ;05 (BYTE COUNT REG) WILL BE SET UP ON THE DIAGNOSTIC MONITOR CALL.
                                        
```

```
1852                                    ;REQUEST THE DIAG. MON. TO SIGNAL THE HOST CPU FOR A WRITE GCR DATA
1853                                    ;TRANSFER USING "MBDPAR" DATA PATTERN (2ND ARG IN CALL). BYTE COUNT
1854                                    ;IS 100. (40 SCLK * 2.5 BYTES PER WORD IN 10 DUMP MODE). REG 02 GETS
1855                                    ;LOADED TO 1 RECORD @ 10 DUMP FORMAT.
1856
1857  44A7                                      REQ     2,1,40,@30004
 (1)  44A7  CD  06  28        18.0                       CALL    REQST
 (1)  44AA  01                                           .BYTE   1               ;DATA PATTERN NUMBER
 (1)  44AB  00  28                                       .WORD   40              ;SYSTEM "40" COUNT
 (1)  44AD  30  04                                       .WORD   @30004          ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  44AF  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (1)  44B0  02                                           .BYTE   2               ;REQUEST CODE
1858
1859                                    ;INIT THE SCLK COUNTER TO 0 ... TEST WILL STOP WHEN COUNT = 41.
1860
1861  44B1  3E  01              7.0           MVI     A,1
1862  44B3  32  53  46        13.0           STA     CLKCNT          ;INIT THE CLOCK COUNTER
1863
1864                                    ;SET THE DBUS CNTRL WORD TO "OCC", AND 'SCLK OUT"
1865
1866  44B6  3E  22              7.0           MVI     A,M.OCC!T.SCLK
1867  44B8  D3  C0             10.0           OUT     DBUSCTL         ;SET THE STATUS - NOW HAVE MASSBUS
1868
1869                                    ;NOW WAIT FOR HOST CPU TO ASSERT "RUN" ON THE MASSBUS...COULD TAKE UP TO
1870                                    ;10 MILLISEC (MASSBUS SPEC.)
1871
1872  44BA  0E  14              7.0           MVI     C,20            ;LOOK FOR 100 M-SEC.
1873  44BC  DB  C0             10.0  T2W:     IN      DBUSSTA         ;GET DBUS STATUS - CHECK FOR "RUN"
1874  44BE  E6  04              7.0           ANI     M.RUN           ;SAVE ONLY "RUN" BIT
1875  44C0  C2  DE  44         10.0           JNZ     TST2C           ;JUMP IF RUN IS UP - OK TO PROCEED
1876  44C3  3E  FF              7.0  T2WL:    MVI     A,@377          ;GET A DELAY TIME
1877  44C5  3D                  4.0  T2WL1:   DCR     A               ;DECREMENT
1878  44C6  C2  C5  44         10.0           JNZ     T2WL1           ;STAY IN LOOP TILL =0
1879                                    ;DOWNCOUNT GROSS TIMER COUNT
1880
1881  44C9  0D                  4.0           DCR     C
1882  44CA  C2  BC  44         10.0           JNZ     T2W             ;THEN CHECK IF "RUN" IS UP
1883  44CD  CD  80  45         18.0           CALL    QUIT
1884  44D0                                    ERR     TSTEND,T2WLC
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44D0  CD  09  28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000A                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44D3  0A                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44D4  00                                          .BYTE
 (1)  44D5  CD  15  28        18.0           T2WLL:: CALL    CYLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44D8  DA  18  28        10.0                   JC      TSTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
1885                                    ;>OPI ERROR - HOST CPU FAILED TO SET "RUN" ON A WRITE COMMAND (6X CODE)
1886                                    ;<TEST ABORTED!
1887  44DB  C3  18  28         10.0           JMP     TSTEND          ;FATAL ERROR! - QUIT NOW
1888
1889                                    ;START THE CLOCK CYCLE (SCLK) BY SETTING THE SCLK BIT IN DBUS CNTRL
1890                                    ;WORD
1891
1892  44DE  CD  68  46         18.0  TST2C:  CALL    CLOCK0          ;CLOCK THE 1ST 18 BITS OF DATA TO TM78
```

```
1893                                    ;DATA NOW TRANSFERED IN AN 18 BIT CHUNK FROM THE HOST CPU.
1894                                    ;CHECK THE DDR REGISTERS FOR EXPECTED RESULTS. REG "B" CONTAINS
1895                                    ;THE "SCLK" COUNT -1 (USED FOR A TABLE POINTER)
1896
1897  44E1  AF                   4.0    T2CHK:  XRA     A               ;CLEAR CAS REG 05 HIGH BYTE
1898  44E2                               .       ROUT    R05H            ;FOR ERROR PRINTING
 (1)  44E2  D3  8B              10.0                    OUT     R05H            ;WRITE AC INTO R05H
 (1)  44E4  7F                   4.0                    MOV     A,A             ;RETRY LINK
1899  44E5  3A  53  46           13.0            LDA     CLKCNT          ;GET THE CLOCK COUNT
1900  44E8                               .       ROUT    R05L            ;SAVE IT IN REG 05 LOW BYTE FOR ERROR REPORT
 (1)  44E8  D3  8A              10.0                    OUT     R05L            ;WRITE AC INTO R05L
 (1)  44EA  7F                   4.0                    MOV     A,A             ;RETRY LINK
1901  44EB  CD  D4  45           18.0            CALL    LD18BIT         ;GET THE EXPECTED DATA
1902  44EE  CD  AC  45           18.0            CALL    DDRACK          ;CHECK BITS 7 TO 0 (REG A)
1903  44F1  CA  05  45           10.0            JZ      TST2D           ;JUMP IF NO ERROR DETECTED
1904  44F4  CD  80  45           18.0            CALL    QUIT            ;ABORT THE XFR NOW.
1905  44F7                               .       ERRB    TST2L,TST2CD,6
 (1)                                   .       ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44F7  CD  12  28           18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000B                                                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44FA  0B                                                   .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44FB  06                                                   .BYTE   6               ;PRINT ROUTINE NUMBER
 (1)  44FC  CD  15  28           18.0                    TST2CD::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44FF  DA  69  44           10.0                            JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
1906                                    ;>MBD BITS 7-0 COMPARE ERROR ON THIS "SCLK" XFR
1907  4502  C3  18  28           10.0            JMP     TSTEND
1908
1909                                    ;CHECK DATA IN BITS 15 TO 8 OF DDR (REG B)
1910
1911  4505  CD  C3  45           18.0    TST2D:  CALL    DDRBCK          ;CHECK BITS 15 TO 8
1912  4508  CA  1C  45           10.0            JZ      TST2E           ;JUMP IF ALL OK
1913  450B  CD  80  45           18.0            CALL    QUIT            ;ABORT THE XFR NOW
1914  450E                               .       ERRB    TST2L,TST2DC,6
 (1)                                   .       ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  450E  CD  12  28           18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000C                                                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4511  0C                                                   .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4512  06                                                   .BYTE   6               ;PRINT ROUTINE NUMBER
 (1)  4513  CD  15  28           18.0                    TST2DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4516  DA  69  44           10.0                            JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
1915                                    ;>MBD BITS 15-8 COMPARE ERROR ON THIS "SCLK" XFR
1916                                    ;<TEST ABORTED!
1917  4519  C3  18  28           10.0            JMP     TSTEND
1918                                    ;NOW TEST DATA IN DDR REG "C" - PARITY BIT AND MASSBUS DATA BITS 17-16
1919                                    ;AND THE PARITY ERROR BITS.
1920
1921  451C  CD  FD  45           18.0    TST2E:  CALL    DDRCCK          ;CHECK DDR REG C
1922  451F  CA  33  45           10.0            JZ      TST2F           ;JUMP IF NO ERROR DETECTED
1923  4522  CD  A1  45           18.0            CALL    QUIT1           ;ABORT WRITE XFR
```

```
1925  4525                                    ERRB    TST2L,TST2EC,6
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4525  CD  12  28        18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000D                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4528  0D                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4529  06                                        .BYTE   6               ;PRINT ROUTINE NUMBER
 (1)  452A  CD  15  28        18.0            TST2EC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  452D  DA  69  44        10.0                    JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
1926                                    ;>MBD BITS 17-16 AND "P" INCORRECT OR STATUS ERROR DETECTED (BITS 7-3
1927                                    ;>OF WMC ERROR WORD - 100332).
1928  4530  C3  18  28        10.0            JMP     TSTEND          ;QUIT!
1929
1930  4533  DB  DA              10.0  TST2F:  IN      WMCERR          ;GET THE WMC ERROR BYTE
1931  4535  E6  40               7.0          ANI     M.PE            ;MASS BUSS PARITY ERROR?
1932  4537  CA  4B  45          10.0          JZ      TST2G           ;NO - CONTINUE
1933  453A  CD  A1  45          18.0          CALL    QUIT1           ;TERMINATE THE MASS BUS TRANSFER
1934  453D                                    ERR     TST2L,TST2FC,6
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  453D  CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000E                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4540  0E                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4541  06                                        .BYTE   6
 (1)  4542  CD  15  28        18.0            TST2FC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4545  DA  69  44        10.0                    JC      TST2L           ;LOOP ADDRESS IF LOOP SPECIFIED
1935                                    ;>MASSBUS PARITY ERROR DETECTED BY THE TM78
1936                                    ;<TEST ABORTED!
1937  4548  C3  18  28        10.0            JMP     TSTEND          ;ABORT THE TEST SEQUENCE
1938                                    ;HERE WHEN TIME TO DO NEXT "SCLK" CYCLE
1939
1940  454B  3A  53  46        13.0  TST2G:  LDA     CLKCNT          ;GET CURRENT CLOCK COUNT
1941  454E  3C                 4.0          INR     A               ;UPDATE IT
1942  454F  32  53  46        13.0          STA     CLKCNT          ;SAVE THE UPDATED COUNTED
1943  4552  FE  29               7.0          CPI     41              ;DONE WITH THE TEST?
1944  4554  CA  5D  45          10.0          JZ      T2END           ;YES - TERMINATE TRANSFER AND EXIT
1945
1946  4557  CD  68  46          18.0          CALL    CLOCK0          ;NO - ISSUE NEXT "SCLK" CYCLE
1947  455A  C3  E1  44          10.0          JMP     T2CHK           ;AND CHECK THE DATA IN DDR REGISTERS
1948
1949                                    ;HERE WHEN THE TRANSFER IS COMPLETE - TERMINATE THE TRANSFER BY ISSUING
195C                                    ;MASSBUS "EBL"
1951
1952  455D  3E  32               7.0  T2END:  MVI     A,M.OCC!M.EBL!T.SCLK  ;GET "OCC" + "EBL" + "SCLK OUT"
1953  455F  D3  C0              10.0          OUT     DBUSCTL         ;SET THE STATUS - TERMINATE THE TRANSFER
1954  4561  3E  22               7.0          MVI     A,M.OCC!T.SCLK  ;DROP "EBL"
1955  4563  D3  C0              10.0          OUT     DBUSCTL
1956  4565  3E  00               7.0          MVI     A,@0            ;CLEAR DBUS CONTROL - ALL DONE WITH MASSBUS
1957  4567  D3  C0              10.0          OUT     DBUSCTL         ;RELEASE THE MASSBUS DATA LINES
```

```
                                        ENDTST  TST2L
                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 4569  CD  06  28       18.0                           CALL    REQST
 456C  00                                              .BYTE                   ;DATA PATTERN NUMBER
 456D  00  00                                          .WORD                   ;SYSTEM "" COUNT
 456F  00  00                                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 4571  00                                              .BYTE                   ;DATA COMPARE FLAG IF =1
 4572  07                                              .BYTE   7               ;REQUEST CODE
 4573  3A  9A  4F        13.0          LDA     ITERA           ;GET ITERATION COUNT
 4576  3D                 4.0          DCR     A               ;DOWNCOUNT
 4577  32  9A  4F        13.0          STA     ITERA           ;SAVE COUNT
 457A  F2  69  44        10.0          JP      TST2L           ;DO TEST UNTIL TILL = 0
 457D  C3  C1  47        10.0          JMP     TEST3           ;DO THE NEXT TEST
                                ;HERE IF A DATA COMPARE ERROR DETECTED ON THE DDR REGISTERS AND
                                ;HOST CPU REQUESTED TO "LOOP-ON-ERROR"

 4580  3E  0A             7.0  QUIT:   MVI     A,M.EXC!T.SCLK  ;SEND MASSBUS "EXC"
 4582  D3  C0            10.0          OUT     DBUSCTL         ;TO TERMINATE HOST XFR
 4584  3E  1A             7.0          MVI     A,M.EBL.M.EXC!T.SCLK  ;SET BOTH EXC AND EBL
 4586  D3  C0            10.0          OUT     DBUSCTL
 4588  3E  00             7.0          MVI     A,0
 458A  D3  C0            10.0          OUT     DBUSCTL         ;RELEASE THE MASSBUS
 458C  3A  54  46        13.0          LDA     SDDRA           ;GET ACTUAL 7-0
 458F                                  ROUT    R16L
 458F  D3  9C            10.0                  OUT     R16L            ;WRITE AC INTO R16L
 4591  7F                 4.0                  MOV     A,A             ;RETRY LINK
 4592  3A  55  46        13.0          LDA     SDDRB           ;GET ACTUAL 15-8
 4595                                  ROUT    R16H
 4595  D3  9D            10.0                  OUT     R16H            ;WRITE AC INTO R16H
 4597  7F                 4.0                  MOV     A,A             ;RETRY LINK
 4598  3A  56  46        13.0          LDA     SDDRC           ;GET ACTUAL 17-16
 459B  F6  80             7.0          ORI     BIT7
 459D                                  ROUT    R17L
 459D  D3  9E            10.0                  OUT     R17L            ;WRITE AC INTO R17L
 459F  7F                 4.0                  MOV     A,A             ;RETRY LINK
 45A0  C9                10.0          RET                     ;EXIT
                                ;HERE IS CODE NECESSARY TO CLEAR PARITY ERROR OR "OTHER" ERRORS DETECTED
                                ;IN WMC ERROR STATUS (ALSO DDR REG "C").

 45A1  3E  09             7.0  QUIT1:  MVI     A,W.WRITE!W.RST ;SET "WRITE" + "RESTART"
 45A3  D3  D3            10.0          OUT     WMCCTL          ;IN WMC CONTROL WORD
 45A5  3E  08             7.0          MVI     A,W.WRITE       ;RESET TO ORIG. STATUS OF "WRITE"
 45A7  D3  D3            10.0          OUT     WMCCTL
 45A9  C3  80  45        10.0          JMP     QUIT            ;NORMAL ERROR TERMINATION OF DATA TRANSFER
                                S
 45AC                           ; ************************************************************************
                                ;DDRACK -- ROUTINE TO CHECK THE CONTENTS OF DDR REG A (DATA BITS 7-0)
                                S
 45AC                           ; ************************************************************************

 45AC  C5                12.0  DDRACK: PUSH    B               ;SAVE B & C
 45AD  32  51  46        13.0          STA     ASAV            ;SAVE DATA IN REG A
 45B0  3A  5A  46        13.0          LDA     SDDRAE          ;GET EXPECTED DATA
```

```
1993  45B3                                    ROUT    EDATA
 (1)  45B3  D3  95            10.0                   OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  45B5  7F                 4.0                   MOV     A,A             ;RETRY LINK
1994  45B6  47                 4.0           MOV     B,A             ;TEMP SAVE IT
1995  45B7  3A  54  46         13.0           LDA     SDDRA           ;GET THE READ DDRA DATA
1996  45BA                              COMCK: ROUT    ADATA           ;SAVE IN CASE OF ERROR
 (1)  45BA  D3  94            10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  45BC  7F                 4.0                   MOV     A,A             ;RETRY LINK
1997  45BD  B8                 4.0           CMP     B               ;IF ERROR DETECTED, "Z" WILL =0
1998  45BE  C1                10.0           POP     B
1999  45BF  3A  51  46         13.0           LDA     ASAV            ;RESET REG A
2000  45C2  C9                10.0           RET
2001
2002  45C3                                  S
 (1)                                        ; *****************************************************************************
2003                                        ;DDRBCK -- ROUTINE TO CHECK THE CONTENTS OF DDR REG B (DATA BITS 15-8)
2004  45C3                                  S
 (1)                                        ; *****************************************************************************
2005
2006  45C3  C5                12.0  DDRBCK: PUSH    B               ;SAVE B & C
2007  45C4  32  51  46         13.0           STA     ASAV            ;SAVE REG A
2008  45C7  3A  5B  46         13.0           LDA     SDDRBE          ;GET EXPECTED DATA
2009  45CA                                   ROUT    EDATA           ;SAVE EXPECTED
 (1)  45CA  D3  95            10.0                   OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  45CC  7F                 4.0                   MOV     A,A             ;RETRY LINK
2010  45CD  47                 4.0           MOV     B,A             ;TEMP SAVE
2011  45CE  3A  55  46         13.0           LDA     SDDRB           ;GET THE "ADATA" FROM DDR REG B
2012  45D1  C3  BA  45         10.0           JMP     COMCK           ;COMMON EXIT
2013  45D4                                  S
 (1)                                        ; *****************************************************************************
2014                                        ;LD18BIT -- ROUTINE TO LOAD THE 'EXPECTED' DDR DATA STORAGE AREA FROM
2015                                        ;       THE CLOCK COUNT IN REG A.
2016  45D4                                  S
 (1)                                        ; *****************************************************************************
2017
2018  45D4  C5                12.0  LD18BIT: PUSH   B               ;SAVE B & C
2019  45D5  E5                12.0           PUSH    H               ;AN  H & L
2020  45D6  3D                 4.0           DCR     A
2021  45D7  47                 4.0           MOV     B,A             ·  Y THE COUNT
2022  45D8  87                 4.0           ADD     A               ,CAL . THE DATA TABLE OFFSET
2023  45D9  80                 4.0           ADD     B               ;REALLY = COUNT * 3
2024  45DA  4F                 4.0           MOV     C,A             ;PUT RESULT IN REG C
2025  45DB  06  00             7.0           MVI     B,0             ;B & C POINT TO DATA TABLE ENTRY
2026  45DD  21  49  47         10.0           LXI     H,DATATB        ;GET START OF TABLE ADDRESS
2027  45E0  09                 10.0           DAD     B               ;H & L NOW POINT TO ENTRY ADDRESS
2028  45E1  7E                  7.0           MOV     A,M             ;GET DDRA EXPECTED
2029  45E2  32  5A  46         13.0           STA     SDDRAE          ;SAVE IT
2030  45E5                                   ROUT    R14L            ;SAV  BITS 7-0 FOR PRINTING EXPECTED
 (1)  45E5  D3  98            10.0                   OUT     R14L            ;WRITE AC INTO R14L
 (1)  45E7  7F                 4.0                   MOV,    A,A             ;RETRY LINK
2031  45E8  23                 6.0           INX     H               ;POINT TO NEXT
2032  45E9  7E                  7.0           MOV     A,M             ;GET EXPECTED DDRB
2033  45EA  32  5B  46         13.0           STA     SDDRBE          ;SAVE IT
2034  45ED                                   ROUT    R14H            ;SAVE BITS 15-8 FOR PRINTING EXPECTED
```

```
 (1)  45ED  D3  99          10.0                OUT   R14H          ;WRITE AC INTO R14H
 (1)  45EF  7F               4.0                MOV   A,A           ;RETRY LINK
2035  45F0  23               6.0          INX   H                  ;POINT TO NEXT
2036  45F1  7E               7.0          MOV   A,M                ;GET EXPECTED DDRC
2037  45F2  32  5C  46       13.0         STA   SDDRCE             ;SAVE IT
2038  45F5  F6  80            7.0         ORI   BIT7               ;SET THE SIGN BIT
2039  45F7                               ROUT  R15L                ;SAVE BITS 17-16 FOR PRINTING EXPECTED
 (1)  45F7  D3  9A          10.0                OUT   R15L          ;WRITE AC INTO R15L
 (1)  45F9  7F               4.0                MOV   A,A           ;RETRY LINK
2040  45FA  E1               10.0         POP   H                  ;RESET REGS
2041  45FB  C1               10.0         POP   B
2042  45FC  C9               10.0         RET                      ;EXIT
2043  45FD                           S
 (1)                               ;*****************************************************************************
2044                               ;DDRCCK -- ROUTINE TO CHECK THE CONTENTS OF DDR REG C (DATA BITS 17-16)
2045                               ;       AND THE PARITY BIT "P" WITH THE MASSBUS DATA PARITY ERROR FLAG
2046                               ;       AND "OTHER" PARITY ERROR STATUS.
2047  45FD                           S
 (1)                               ;*****************************************************************************
2048
2049  45FD  C5               12.0  DDRCCK: PUSH  B                  ;SAVE REG B & C
2050  45FE  32  51  46       13.0         STA   ASAV               ;SAVE REG A
2051  4601  3A  5C  46       13.0         LDA   SDDRCE             ;GET EXPECTED DDRC DATA
2052  4604                               ROUT  EDATA               ;SAVE IN CASE ERROR DETECTED
 (1)  4604  D3  95          10.0                OUT   EDATA         ;WRITE AC INTO EDATA
 (1)  4606  7F               4.0                MOV   A,A           ;RETRY LINK
2053  4607  47               4.0          MOV   B,A                ;TEMP SAVE
2054  4608  3A  56  46       13.0         LDA   SDDRC              ;GET REG C DATA
2055  460B  E6  40            7.0         ANI   @100               ;SEE IF ANY DATA PARITY ERROR STATUS DETECTED
2056  460D  C2  5D  46       10.0         JNZ   DDRCER             ;ERROR DETECTED!!!
2057  4610  3A  56  46       13.0         LDA   SDDRC              ;GET DDR REG C AGAIN
2058  4613  E6  07            7.0         ANI   7                  ;SAVE ONLY "P" AND 17-16
2059  4615  32  56  46       13.0         STA   SDDRC              ;SAVE ONLY DATA IN DDRC AREA
2060  4618  B8               4.0          CMP   B                  ;SAME AS EXPECTED?
2061  4619  C2  21  46       10.0         JNZ   DDRCED             ;NO - POSSIBLE HOST NOT CAPABLE...
2062
2063                               ;HERE TO EXIT DDR REG C CHECK
2064
2065  461C  C1               10.0  DDRCEX: POP   B                  ;RESET REGS
2066  461D  3A  51  46       13.0         LDA   ASAV
2067  4620  C9               10.0         RET                      ;EXIT
2068                               ;HERE IF REG C DATA ERROR DETECTED OR HOST NOT CAPABLE ERROR
2069
2070  4621  3A  98  4F       13.0  DDRCED: LDA   VALFC              ;GET THE FUNCTION CODE RETURNED FROM REQ
2071  4624  FE  35            7.0         CPI   @65                ;SEE IF HOST "NOT CAPABLE" IS SET
2072  4626  C2  1C  46       10.0         JNZ   DDRCEX             ;NOT SET - MUST BE REAL ERROR!
2073
2074  4629  78               4.0          MOV   A,B                ;GET EXPECTED DATA FOR DDRC AGAIN
2075  462A  E6  04            7.0         ANI   4                  ;SAVE ONLY "P"
2076  462C  32  52  46       13.0         STA   PBIT               ;SAVE THE CONDITION OF "P" DATA
2077  462F  78               4.0          MOV   A,B                ;GET DATA AGAIN
2078  4630  E6  03            7.0         ANI   3                  ;SAVE ONLY 17 & 16 DATA BITS
2079  4632  CA  43  46       10.0         JZ    PNMOD              ;NO MODIFICATION TO "P" BIT
2080  4635  FE  03            7.0         CPI   3                  ;IS IT =3 (NO MOD IF YES)
```

```
2081  4637  CA  43  46   10.0          JZ      PNMOD
2082  463A  3A  52  46   13.0          LDA     PBIT            ;GET THE CURRENT PBIT CONDITION
2083  463D  2F               4.0       CMA                     ;COMPLIMENT IT
2084  463E  E6  04           7.0       ANI     @4              ;SAVE ONLY THE P BIT AGAIN
2085  4640  32  52  46   13.0          STA     PBIT            ;SAVE
2086
2087  4643  3A  52  46   13.0  PNMOD:  LDA     PBIT            ;GET IT AGAIN
2088  4646                           ROUT    EDATA           ;UPDATE "EDATA" STORED
 (1)  4646  D3  95       10.0                   OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4648  7F           4.0                    MOV     A,A             ;RETRY LINK
2089  4649  47           4.0           MOV     B,A             ;SAVE NEW VALUE OF EXPECTED DDRC DATA
2090  464A  3A  56  46   13.0          LDA     SDDRC           ;GET DDR REG C AGAIN
2091  464D  B8           4.0           CMP     B               ;OK NOW?
2092  464E  C3  1C  46   10.0          JMP     DDRCEX          ;EXIT - "Z" SET IF OK
2093
2094  4651  00                 ASAV:   .BYTE   0               ;STORAGE FOR REG A
2095  4652  00                 PBIT:   .BYTE   0               ;STORAGE FOR MODIFIED DDR REG "C"
2096  4653  00                 CLKCNT: .BYTE   0               ;COPY OF THE SCLK COUNT IN PROGRESS
2097  4654  00                 SDDRA:  .BYTE   0               ;STORAGE FOR DDRA DATA
2098  4655  00                 SDDRB:  .BYTE   0               ;STORAGE FOR DDRB DATA
2099  4556  00                 SDDRC:  .BYTE   0               ;STORAGE FOR DDRC DATA
2100  4657  00                 SDDRAT: .BYTE   0               ;TEMP STORE DDRA (REAL DDRA)
2101  4658  00                 SDDRBT: .BYTE   0               ;TEMP STORE DDRB (REAL DDRB)
2102  4659  00                 SDDRCT: .BYTE   0               ;TEMP STORE DDRC (REAL DDRC)
2103  465A  00                 SDDRAE: .BYTE   0               ;EXPECTED DDRA DATA STORAGE
2104  465B  00                 SDDRBE: .BYTE   0               ;EXPECTED DDRB
2105  465C  00                 SDDRCE: .BYTE   0               ;EXPECTED DDRC
2106                           ;HERE IF PARITY ERROR OR OTHER STATUS ERROR DETECTED IN WMC ERROR WORD
2107
2108  465D  3A  56  46   13.0  DDRCER: LDA     SDDRC           ;GET DDR REG C AGAIN
2109  4660                           ROUT    ADATA           ;SAVE AS "ADATA" FOR ERROR REPORT
 (1)  4660  D3  94       10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4662  7F           4.0                    MOV     A,A             ;RETRY LINK
2110  4663  E6  FF           7.0       ANI     @377
2111  4665  C3  1C  46   10.0          JMP     DDRCEX          ;THEN EXIT
2112
2113                           ;HERE TO DO A "SCLK" CYCLE
2114
2115  4668  F5           12.0  CLOCK0: PUSH    PSW             ;SAVE A
2116  4669  C5           12.0          PUSH    B               ;SAVE B & C
2117  466A  DB  C0        10.0          IN      DBUSSTA         ;GET CURRENT DBUS STATUS
2118  466C  F6  01         7.0          ORI     M.SCLK          ;ADD IN THE SCLK BIT
2119  466E  47             4.0          MOV     B,A             ;SAVE TEMP RESULT
2120  466F  E6  80         7.0          ANI     M.WREN          ;SEE IF WRITTING OR READING
2121  4671  78             4.0          MOV     A,B             ;RESET REG A
2122  4672  C2  77  46   10.0          JNZ     CLOCK1          ;JUMP OVER IF TM78 IS WRITTING TO HOST
2123  4675  F6  02         7.0          ORI     T.SCLK          ;SET TM78 READ ENABLE
2124  4677  D3  C0        10.0  CLOCK1: OUT     DBUSCTL         ;SET "SCLK"
2125  4679  DB  C0        10.0          IN      DBUSSTA         ;GET MASSBUS STATUS
2126  467B  E6  80         7.0          ANI     M.WREN          ;SAVE ONLY WRITE ENABLE BIT
2127  467D  C2  83  46   10.0          JNZ     CLKSKP          ;SKIP OVER DDR READ CODE IF SET
2128  4680  CD  95  46   18.0          CALL    GETDDR          ;GET DATA TO DDR STORAGE AREA
2129  4683  DB  C0        10.0  CLKSKP: IN      DBUSSTA         ;GET STATUS AGAIN
2130  4685  E6  B0         7.0          ANI     @260            ;SAVE ALL EXCEPT "SCLK"
```

```
2131  4687  47                4.0          MOV     B,A             ;SAVE TEMP RESULT
2132  4688  E6  80             7.0          ANI     M.WREN
2133  468A  78                4.0          MOV     A,B             ;RESET REG A
2134  468B  C2  90  46        10.0          JNZ     CLOCK2          ;JUMP OVER IF WRITTONG TO HOST
2135  468E  F6  02             7.0          ORI     T.SCLK          ;SET READ ENABLE BIT
2136  4690  D3  C0            10.0  CLOCK2: OUT     DBUSCTL         ;CLEAR "SCLK" BIT IN DBUS CNTRL WORD
2137  4692  C1                10.0          POP     B               ;RESET THE SAVED REGS
2138  4693  F1                10.0          POP     PSW
2139  4694  C9                10.0          RET                     ;EXIT
2140  4695                                  $
 (1)                                      ; *****************************************************************************
2141                                       ;GETDDR -- ROUTINE TO RETRIEVE DATA FROM MASSBUS AND SAVE IT IN THE DDR
2142                                       ;       STORAGE AREA.
2143  4695                                 $
 (1)                                      ; *****************************************************************************
2144
2145  4695  DB  D8            10.0  GETDDR: IN      DDRA            ;GET DDRA DATA
2146  4697  32  57  46        13.0          STA     SDDRAT          ;SAVE IT
2147  469A  32  54  46        13.0          STA     SDDRA
2148  469D  DB  D9            10.0          IN      DDRB            ;GET DDRB DATA
2149  469F  32  58  46        13.0          STA     SDDRBT          ;SAVE IT
2150  46A2  32  55  46        13.0          STA     SDDRB
2151  46A5  DB  DA            10.0          IN      DDRC            ;GET DDRC DATA
2152  46A7  32  59  46        13.0          STA     SDDRCT          ;SAVE IT
2153  46AA  32  56  46        13.0          STA     SDDRC
2154  46AD  DB  A1            10.0          IN      CBUSSTA         ;SEE IF "LEFT" IS SET
2155  46AF  E6  04             7.0          ANI     W.LEFT          ;SEE IF WMC LEFT IS UP
2156  46B1  C0                12.0          RNZ                     ;RETURN IF SET - NO MODIFICATION NEEDED
2157
2158                                       ;WMC LEFT IS NOT SET (IN RIGHT MODE) MODIFY THE DDR DATA...
2159
2160  46B2  3A  57  46        13.0          LDA     SDDRAT          ;GET TEMP A
2161  46B5  E6  FC             7.0          ANI     $FC             ;STRIP LOW BITS
2162  46B7  0F                 4.0          RRC
2163  46B8  0F                 4.0          RRC                     ;POSITION TO RIGHT
2164  46B9  47                 4.0          MOV     B,A             ;STORE IN REG B
2165  46BA  3A  58  46        13.0          LDA     SDDRBT          ;GET TEMP B
2166  46BD  E6  03             7.0          ANI     $03             ;SAVE ONLY LOW BITS
2167  46BF  0F                 4.0          RRC
2168  46C0  0F                 4.0          RRC                     ;LEFT JUSTIFY BITS
2169  46C1  B0                 4.0          ORA     B               ;COMBINE THE TWO
2170  46C2  32  54  46        13.0          STA     SDDRA           ;SAVE THE DDRA DESIRED
2171
2172  46C5  3A  58  46        13.0          LDA     SDDRBT          ;GET TEMP B AGAIN
2173  46C8  E6  FC             7.0          ANI     $FC             ;SAVE ONLY TOP BITS
2174  46CA  0F                 4.0          RRC
2175  46CB  0F                 4.0          RRC                     ;RIGHT JUSTIFY
2176  46CC  47                 4.0          MOV     B,A             ;TEMP STORE
2177  46CD  3A  59  46        13.0          LDA     SDDRCT          ;GET TEMP DDRC DATA
2178  46D0  E6  03             7.0          ANI     $03             ;SAVE LOW BITS
2179  46D2  0F                 4.0          RRC
2180  46D3  0F                 4.0          RRC                     ;LEFT JUSTIFY
2181  46D4  B0                 4.0          ORA     B               ;COMBINE DATA
2182  46D5  32  55  46        13.0          STA     SDDRB           ;SAVE DESIRED DDRB DATA
```

```
2183
2184  46D8  3A 57 46   13.0          LDA     SDDRAT          ;GET DDRA TEMP DATA
2185  46DB  E6 03       7.0          ANI     $03             ;SAVE LOW BITS
2186  46DD  47           4.0          MOV     B,A             ;TEMP SAVE
2187  46DE  3A 59 46   13.0          LDA     SDDRCT          ;GET DDRC TEMP DATA
2188  46E1  E6 04       7.0          ANI     $04             ;SAVE THE "P" BIT
2189  46E3  B0           4.0          ORA     B               ;COMBINE DATA
2190  46E4  32 56 46   13.0          STA     SDDRC           ;SAVE DESIRED DDRC DATA
2191  46E7  C9          10.0          RET                     ;EXIT
2192  46E8                       S
 (1)                               ;*****************************************************************************
2193                               ;SETDDR -- ROUTINE TO TRANSLATE THE DESIRED DATA FROM A 1-FOR-1 BIT
2194                               ;       POSITION TO THE DDR TO BE USED AS DATA TO EITHER THE TM78 OR
2195                               ;       THE MASSBUS. WMC "LEFT" SET INDICATES THAT THIS ROUTINE IS NOT
2196                               ;       NECESSARY, THE DATA IS IN THE CORRECT FORMAT ALREADY. IF WMC
2197                               ;       "LEFT" IS NOT SET, THE DATA MUST BE TRANSLATED TO A COMPATABLE
2198                               ;       DDR FORMAT.
2199  46E8                       S
 (1)                               ;*****************************************************************************
2200
2201  46E8  DB A1       10.0  SETDDR: IN      CBUSSTA         ;GET THE STATUS OF THE WMC "LEFT" BIT
2202  46EA  E6 04        7.0          ANI     W.LEFT          ;SEE IF THE BIT IS SET
2203  46EC  C2 27 47    10.0          JNZ     SLFTDDR         ;WMC SET TO "LEFT" MODE - NO TRANSLATION
2204
2205                               ;HERE TO TRANSLATE THE DATA INTO AN ACCEPTABLE FORMAT FOR DDR
2206
2207  46EF  3A 57 46   13.0          LDA     SDDRAT          ;GET DDRA TEMP
2208  46F2  E6 3F       7.0          ANI     @77             ;STRIP 2 BITS OFF
2209  46F4  07           4.0          RLC
2210  46F5  07           4.0          RLC                     ;LEFT JUSTIFY
2211  46F6  47           4.0          MOV     B,A             ;TEMP SAVE
2212  46F7  3A 59 46   13.0          LDA     SDDRCT          ;GET DDRC TEMP
2213  46FA  E6 03       7.0          ANI     @3              ;SAVE 2 BITS
2214  46FC  B0           4.0          ORA     B
2215  46FD  32 54 46   13.0          STA     SDDRA           ;SAVE DDRA DESIRED
2216
2217  4700  3A 58 46   13.0          LDA     SDDRBT          ;GET DDRB TEMP
2218  4703  E6 3F       7.0          ANI     @77
2219  4705  07           4.0          RLC
2220  4706  07           4.0          RLC                     ;LEFT JUSTIFY
2221  4707  47           4.0          MOV     B,A
2222  4708  3A 57 46   13.0          LDA     SDDRAT          ;GET DDRA AGAIN
2223  470B  E6 C0       7.0          ANI     @300            ;SAVE 2 BITS
2224  470D  07           4.0          RLC
2225  470E  07           4.0          RLC
2226  470F  B0           4.0          ORA     B
2227  4710  32 55 46   13.0          STA     SDDRB           ;SAVE DDRB DATA
2228
2229  4713  3A 59 46   13.0          LDA     SDDRCT          ;GET DDRC TEMP
2230  4716  E6 04       7.0          ANI     @4              ;SAVE THE PARITY BIT
2231  4718  47           4.0          MOV     B,A             ;TEMP STORE
2232  4719  3A 58 46   13.0          LDA     SDDRBT          ;GET DDRB AGAIN
2233  471C  E6 C0       7.0          ANI     @300            ;SAVE 2 BITS
2234  471E  07           4.0          RLC
```

```
2235  471F  07              4.0          RLC                    ;RIGHT JUSTIFY
2236  4720  B0              4.0          ORA     B
2237  4721  32  56  46     13.0          STA     SDDRC          ;SAVE DDRC DATA
2238  4724  C3  39  47     10.0          JMP     SDDCOM         ;COMMON EXIT
2239                                   ;HERE IF NO MODIFICATION OF DDR DATA NECESSARY
2240
2241  4727  3A  57  46     13.0  SLFTDDR: LDA    SDDRAT         ;GET DDRA DATA
2242  472A  32  54  46     13.0          STA     SDDRA          ;SAVE FOR OUTPUT
2243  472D  3A  58  46     13.0          LDA     SDDRBT         ;GET TEMP DDRB DATA
2244  4730  32  55  46     13.0          STA     SDDRB          ;SAVE IT
2245  4733  3A  59  46     13.0          LDA     SDDRCT         ;GET DDRC TEMP DATA
2246  4736  32  56  46     13.0          STA     SDDRC          ;SAVE FOR OUTPUT
2247
2248  4739  3A  54  46     13.0  SDDCOM: LDA     SDDRA          ;GET DATA TO OUTPUT
2249  473C  D3  D8          10.0          OUT     DDRA
2250  473E  3A  55  46     13.0          LDA     SDDRB
2251  4741  D3  D9          10.0          OUT     DDRB
2252  4743  3A  56  46     13.0          LDA     SDDRC
2253  4746  D3  DA          10.0          OUT     DDRC
2254  4748  C9              10.0          RET                    ;EXIT
2255                                   ;HERE IS THE TABLE OF EXPECTED DATA - 40 18 BIT WORDS
2256
2257  4749  01                           DATATB: .BYTE   1      ;FLOATING 1'S
2258  474A  00                                   .BYTE   0
2259  474B  00                                   .BYTE   0
2260  474C  02                                   .BYTE   2
2261  474D  00                                   .BYTE   0
2262  474E  00                                   .BYTE   0
2263  474F  04                                   .BYTE   4
2264  4750  00                                   .BYTE   0
2265  4751  00                                   .BYTE   0
2266  4752  08                                   .BYTE   @10
2267  4753  00                                   .BYTE   0
2268  4754  00                                   .BYTE   0
2269  4755  10                                   .BYTE   @20
2270  4756  00                                   .BYTE   0
2271  4757  00                                   .BYTE   0
2272  4758  20                                   .BYTE   @40
2273  4759  00                                   .BYTE   0
2274  475A  00                                   .BYTE   0
2275  475B  40                                   .BYTE   @100
2276  475C  00                                   .BYTE   0
2277  475D  00                                   .BYTE   0
2278  475E  80                                   .BYTE   @200
2279  475F  00                                   .BYTE   0
2280  4760  00                                   .BYTE   0
2281  4761  00                                   .BYTE   @0
2282  4762  01                                   .BYTE   1
2283  4763  00                                   .BYTE   0
2284  4764  00                                   .BYTE   0
2285  4765  02                                   .BYTE   2
2286  4766  00                                   .BYTE   0
2287  4767  00                                   .BYTE   0
2288  4768  04                                   .BYTE   4
```

```
2289  4769  00          .BYTE   0
2290  476A  00          .BYTE   0
2291  476B  08          .BYTE   @10
2292  476C  00          .BYTE   0
2293  476D  00          .BYTE   0
2294  476E  10          .BYTE   @20
2295  476F  00          .BYTE   0
2296  4770  00          .BYTE   0
2297  4771  20          .BYTE   @40
2298  4772  00          .BYTE   0
2299  4773  00          .BYTE   0
2300  4774  40          .BYTE   @100
2301  4775  00          .BYTE   0
2302  4776  00          .BYTE   0
2303  4777  80          .BYTE   @200
2304  4778  00          .BYTE   0
2305  4779  00          .BYTE   0
2306  477A  00          .BYTE   0
2307  477B  01          .BYTE   1
2308  477C  00          .BYTE   0
2309  477D  00          .BYTE   0
2310  477E  02          .BYTE   2
2311  477F  FF          .BYTE   @377            ;FLOATING 0'S
2312  4780  FF          .BYTE   @377
2313  4781  01          .BYTE   1
2314  4782  FF          .BYTE   @377
2315  4783  FF          .BYTE   @377
2316  4784  02          .BYTE   2
2317  4785  FF          .BYTE   @377
2318  4786  7F          .BYTE   @177
2319  4787  03          .BYTE   3
2320  4788  FF          .BYTE   @377
2321  4789  BF          .BYTE   @277
2322  478A  03          .BYTE   3
2323  478B  FF          .BYTE   @377
2324  478C  DF          .BYTE   @337
2325  478D  03          .BYTE   3
2326  478E  FF          .BYTE   @377
2327  478F  EF          .BYTE   @357
2328  4790  03          .BYTE   3
2329  4791  FF          .BYTE   @377
2330  4792  F7          .BYTE   @367
2331  4793  03          .BYTE   3
2332  4794  FF          .BYTE   @377
2333  4795  FB          .BYTE   @373
2334  4796  03          .BYTE   3
2335  4797  FF          .BYTE   @377
2336  4798  FD          .BYTE   @375
2337  4799  03          .BYTE   3
2338  479A  FF          .BYTE   @377
2339  479B  FE          .BYTE   @376
2340  479C  03          .BYTE   3
2341  479D  7F          .BYTE   @177
2342  479E  FF          .BYTE   @377
```

```
2343  479F  03                          .BYTE   3
2344  47A0  BF                          .BYTE   @277
2345  47A1  FF                          .BYTE   @377
2346  47A2  03                          .BYTE   3
2347  47A3  DF                          .BYTE   @337
2348  47A4  FF                          .BYTE   @377
2349  47A5  03                          .BYTE   3
2350  47A6  EF                          .BYTE   @357
2351  47A7  FF                          .BYTE   @377
2352  47A8  03                          .BYTE   3
2353  47A9  F7                          .BYTE   @367
2354  47AA  FF                          .BYTE   @377
2355  47AB  03                          .BYTE   3
2356  47AC  FB                          .BYTE   @373
2357  47AD  FF                          .BYTE   @377
2358  47AE  03                          .BYTE   3
2359  47AF  FD                          .BYTE   @375
2360  47B0  FF                          .BYTE   @377
2361  47B1  03                          .BYTE   3
2362  47B2  FE                          .BYTE   @376
2363  47B3  FF                          .BYTE   @377
2364  47B4  03                          .BYTE   3
2365  47B5  FF                          .BYTE   @377            ;ALL 1''S
2366  47B6  FF                          .BYTE   @377
2367  47B7  07                          .BYTE   7
2368  47B8  00                          .BYTE   0               ;ALL 0''S
2369  47B9  00                          .BYTE   0
2370  47BA  04                          .BYTE   4
2371  47BB  55                          .BYTE   @125            ;ALTERNATING BITS
2372  47BC  55                          .BYTE   @125
2373  47BD  01                          .BYTE   1
2374  47BE  AA                          .BYTE   @252            ;COMPLIMENT ALT. BIT DATA
2375  47BF  AA                          .BYTE   @252
2376  47C0  02                DTEND:    .BYTE   2
```

```
                                .SBTTL  TEST 3 - READ DATA TRANSFER USING THE "DDR" REGISTERS - DDR IN RIGHT MODE
47C1                            ST
 (1)                            ; ****************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*READ DATA TRANSFER USING THE "DDR" REGISTERS - DDR IN RIGHT MODE
47C1                            SM
 (1)                            ; ****************************************************************************
 (1)                            ;*MODULE(S) UNDER TEST
 (1)                            ;*--------------------
                                ;*M8956 & M8959
47C1                            SD
 (1)                            ; ****************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST REQUIRES THE HOST TO READ THE "MBDPAR" DATA PATTERN WHICH
                                ;*IS 40 18 BIT WORDS LONG. THE BYTE COUNT REGISTER (05) WILL BE LOADED
                                ;*TO 100 BYTES (10 DUMP MODE USED -- 5 BYTES PER WORD * 40 WORDS).
                                ;*
                                ;*EACH 18 BIT WORD WILL BE CHECKED TO THE HOST CPU OVEP THE MASSBUS DATA
                                ;*LINES WITH DIAGNOSTIC "SCLK" SIGNAL.
                                ;*
                                ;*ANY MICROCOMPUTER DETECTED ERROR DETECTED DURING THE DATA TRANSFEP WILL
                                ;*BE REPORTED TO THE HOST CPU.
                                ;*
                                ;*READ XFER TEST
47C1                            SP
 (1)                            ; ****************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  INIT TEST-TESTX MACRO
                                ;*  ISSUE WMC RESTART TO INIT SYSTEM
                                ;*  INIT ECODE COUNTER TO 0 + SELECT WMC CONTROL TO 'WRITE' TO HOST CPU
                                ;*  REQUEST HOST TO START A READ FORWARD COMMAND @ 10 DUMP MODE WITH
                                ;*   'MBDPAR' DATA EXPECTED-40 SCLKS OF DATA SENT TO HOST CPU
                                ;*  CONTINUE WHEN HOST RESPONDS WITH A 71 FUNCTION CODE
                                ;*  INIT 'SCLK' COUNT TO '0' + SET MASSBUS 'OCC' + 'RUN' + 'MBWRTEN'
                                ;*  SET 'SCLK' + WAIT SMALL AMOUNT OF TIME
                                ;*  IF HOST RESPOND WITH 'WCLK'?
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT MASSBUS HANDSHAKE FAILURE ON DATA BUS-OPI ERROR
                                ;*  ENDIF
                                ;*  FINISH CLOCK CYCLE BY CLEARING 'SCLK' IN DBUS CONTROL
                                ;*  IF 'SCLK' GO AWAY?
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT MASSBUS STATUS ERROR
                                ;*  BGNDO
                                ;*  :  DO UNTIL 40 'SCLKS' OF DATA SENT
                                ;*  :  IF XFR DONE-40 CLKS
                                ;*  :  :  THEN-CONTINUE
                                ;*  :  :  ELSE-
                                ;*  :  ENDIF
                                ;*  ENDDO
```

```
                                ;*  TERMINATE HOST READ SFR BY SETTING 'EBL'-THEN TERMINATE BY DROPPING
                                ;*  MASSBUS 'OCC' + 'RUN' BITS IN DBUS CONTROL
                                ;*  REQUEST HOST CPU TO CHECK THE TERMINATION STATUS FROM READ XFR
                                ;*  TERMINATE TEST IF HOST RESPONDS WITH A 31-CONTINUE CODE
                                ;*  OR REPEAT TEST IF HOST RESPONDS WITH A 33-LOOP ON ERROR CODE
                                ;*ENDTST
47C1                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MBD1 MICRO TEST 03
                                ;*MBD1 MICRO ERROR 17
                                ;*MBD1-DATA READ FROM TM78 TEST USING DIAG DATA REG.-"DDR" IN "RIGHT" MODE
                                ;*M8956, M8959
                                ;*OPI ERROR - HOST CPU FAILED TO SET "RUN" FOR A READ COMMAND  7X CODE)
                                ;*TEST ABORTED!
                                ;*
                                ;*MBD1 MICRO TEST 03
                                ;*MBD1 MICRO ERROR 20
                                ;*MBD1-DATA READ FROM TM78 TEST USING DIAG DATA REG.-"DDR" IN "RIGHT" MODE
                                ;*M8956, M8959
                                ;*"SCLK" STUCK ON ... DIDN'T CLEAR AFTER IT WAS SET FOR DATA TRANSFER
                                ;*TEST ABORTED'
47C1                            S
 (1)                            ; ************************************************************************

47C1                            TEST3:  TESTX   @3
47C1  3E  03          7.0                       MVI     A,@3            ;DEFINE THE TEST NUMBER
47C3  CD  03  28     18.0                       CALL    TSET            ;SETUP THE TEST
                                ;%MBD-DATA READ FROM TM78 TEST USING DIAG. DATA REG.-"DDR" IN "RIGHT" MODE
                                ;&M8956, M8959

47C6  3E  01          7.0   TST3L:  MVI     A,W.RST         ;RESTART THE WMC
47C8  D3  D3         10.0           OUT     WMCCTL

                                ;FINISH THE WMC RESTART TO CLEAR THE WMC "LEFT" BIT

47CA  AF              4.0           XRA     A               ;NOW DROP WMC "RST" BI
47CB  D3  D3         10.0           OUT     WMCCTL
47CD  3E  B0          7.0           MVI     A,@260          ;INIT THE "ECODE" COUNTER
47CF  D3  D7         10.0           OUT     CNTCTL          ;SELECT THE COUNTER
47D1                                CLRECT
47D1  AF              4.0                   XRA     A               ;CLEAR THE ACCUMULATOR
47D2  D3  D6         10.0                   OUT     ERRCNT          ;CLEAR BITS 7-0
47D4  D3  D6         10.0                   OUT     ERRCNT          ;CLEAR BITS 15-8

                                ;SET THE "DDR" CONTROL WORD TO "OUT" TRANSFER

47D6  3E  88          7.0           MVI     A,@210
47D8  D3  DB         10.0           OUT     DDRCTL          ;INIT THE DDR CONTROL WORD
                                ;REQUEST THE HOST TO SETUP FOR A 100 BYTE READ TRANSFER (40 18 BIT WORDS)
                                ;AND EXPECT FUNCTION CODE 71 RETURNED (READ FORWARD COMMAND). PACKING
                                ;FORMAT IS 10 DUMP MODE.
```

```
2465  47DA  3E 08        7.0  TST3GO: MVI   A,W.WRITE            ;NOW TELL THE WMC TO 'WRITE" SO
2466  47DC  D3 D3       10.0          OUT   WMCCTL               ;DATA REG (DR) WILL NOT BE ENABLED TO MASSBUS
2467  47DE                             REQ   3,1,40,@30004
 (1)  47DE  CD 06 28    18.0                CALL  REQST
 (1)  47E1  01                              .BYTE 1              ;DATA PATTERN NUMBER
 (1)  47E2  00 28                           .WORD 40             ;SYSTEM "40" COUNT
 (1)  47E4  30 04                           .WORD @30004         ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  47E6  00                              .BYTE                ;DATA COMPARE FLAG IF =1
 (1)  47E7  03                              .BYTE 3              ;REQUEST CODE
2468
2469                                ;INIT THE SCLK COUNTER TO 1 - TEST WILL QUIT WHEN COUNT IS 41.
2470
2471  47E8  3E 01        7.0          MVI   A,1
2472  47EA  32 53 46    13.0         STA   CLKCNT               ;INIT THE CLOCK COUNTER
2473
2474                                ;HOST IS READY FOR THE DATA TRANSFER...
2475                                ;SET MASSBUS OCCUPIED (OCC), AND "MB WR EN"
2476
2477  47ED  3E A0        7.0          MVI   A,M.WREN!M.OCC
2478  47EF  D3 C0       10.0          OUT   DBUSCTL              ;SET OCC + MB WR EN
2479
2480                                ;NOW WAIT FOR HOST CPU TO ASSERT "RUN" ON THE MASSBUS...COULD TAKE UP TO
2481                                ;10 MILLISEC (MASSBUS SPEC.)
2482
2483  47F1  0E 14        7.0          MVI   C,20                 ;LOOK FOR 100 MILLI-SEC.
2484  47F3  DB C0       10.0  T3W:    IN    DBUSSTA              ;GET DBUS STATUS - CHECK FOR "RUN"
2485  47F5  E6 04        7.0          ANI   M.RUN                ;SAVE ONLY "RUN" BIT
2486  47F7  C2 15 48    10.0          JNZ   TST3WC               ;JUMP IF RUN IS UP - OK TO PROCEED
2487  47FA  3E FF        7.0  T3WL:   MVI   A,@377               ;GET A DELAY TIME
2488  47FC  3D           4.0  T3WL1:  DCR   A                   ;DECREMENT
2489  47FD  C2 FC 47    10.0          JNZ   T3WL1                ;STAY IN LOOP TILL =0
2490
2491                                ;DOWNCOUNT GROSS TIMER COUNT
2492
2493  4800  0D           4.0          DCR   C
2494  4801  C2 F3 47    10.0          JNZ   T3W                  ;THEN CHECK IF "RUN" IS UP
2495  4804  CD A3 48    18.0          CALL  QUITW
2496  4807                             ERR   TSTEND,T3WLC
 (1)                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4807  CD 09 28    18.0                CALL  ERLP             ;PROCESS ERROR - DO 2.3
 (1)       000F                             MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  480A  0F                              .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  480B  00                              .BYTE
 (1)  480C  CD 15 28    18.0          T3WLC:: CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  480F  DA 18 28    10.0                JC    TSTEND           ;LOOP ADDRESS IF LOOP SPECIFIED
2497                                ;>OPI ERROR - HOST CPU FAILED TO SET "RUN" FOR A READ COMMAND (7X CODE)
2498                                ;<TEST ABORTED!
2499  4812  C3 18 28    10.0          JMP   TSTEND               ;FATAL ERROR! - QUIT NOW
2500                                ;POINT TO THE TABLE OF DATA TO TRANSFER
2501
2502  4815  21 49 47    10.0  TST3WC: LXI   H,DATATB
2503
2504  4818  7E           7.0  TST3A:  MOV   A,M                 ;GET A BYTE OF DATA
2505  4819  32 57 46    13.0          STA   SDDRAT               ;SAVE DDR REG A - BITS 7-0
```

```
2506  481C  23              6.0          INX     H               ;POINT TO NEXT DATA BYTE
2507  481D  7E              7.0          MOV     A,M             ;GET THE DATA
2508  481E  32  58  46     13.0          STA     SDDRBT          ;SAVE IN DDR REG B - BITE 15-8
2509  4821  23              6.0          INX     H               ;POINT TO THE NEXT DATA BYTE
2510  4822  7E              7.0          MOV     A,M             ;GET IT
2511  4823  32  59  46     13.0          STA     SDDRCT          ;SAVE IN DDR REG C - BITS "P" + 17-16
2512  4826  CD  E8  46     18.0          CALL    SETDDR          ;LOAD THE DDR REGS
2513                                ;START THE DATA TRANSFER BY 1ST SETTING "SCLK"
2514
2515  4829  3E  A1            7.0  TST3B:  MVI     A,M.WREN!M.OCC!M.SCLK  ;SET DBUS CNTRL WORD TO
2516  482B  D3  C0           10.0          OUT     DBUSCTL         ;"MB WR EN" + "OCC" + "SCLK"
2517
2518                                ;FINISH THE 1ST SCLK CYCLE BY DROPPING THE SCLK BIT IN DBUS CNTRL WORD
2519
2520  482D  3E  A0            7.0  TST3BB: MVI     A,M.WREN.M.OCC  ;SET DBUS CNTRL WORD TO
2521  482F  D3  C0           10.0          OUT     DBUSCTL         ;"MB WR EN" + "OCC"
2522  4831  00                4.0          NOP                     ;SMALL DELAY
2523  4832  DB  C0           10.0          IN      DBUSSTA         ;GET DBUS CONTROL STATUS
2524  4834  E6  01            7.0          ANI     M.SCLK          ;SAVE ONLY SCLK BIT
2525  4836  CA  4A  48       10.0          JZ      TST3C           ;JUMP OVER IF THE BIT CLEARED AS EXPECTED
2526  4839  CD  A3  48       18.0          CALL    QUITW           ;ABORT THE WRITE XFR
2527  483C                                ERR     TST3L,TST3BC
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  483C  CD  09  28       18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0010                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  483F  10                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4840  00                           .BYTE
 (1)  4841  CD  15  28       18.0  TST3BC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4844  DA  C6  47       10.0                  JC      TST3L           ;LOOP ADDRESS IF LOOP SPECIFIED
2528                                ;>"SCLK" STUCK ON ... DIDN'T CLEAR AFTER IT WAS SET FOR DATA TRANSFER
2529                                ;<TEST ABORTED!
2530  4847  C3  18  28       10.0          JMP     TSTEND          ;QUIT TESTING
2531
2532                                ;SETUP FOR THE NEXT DATA TRANSFER WORD
2533  484A  23              6.0  TST3C:  INX     H               ;POINT TO NEXT DATA BYTE WORD
2534  484B  7E              7.0          MOV     A,M             ;GET A BYTE OF DATA
2535  484C  32  57  46     13.0          STA     SDDRAT          ;SAVE DDR REG A - BITS 7-0
2536  484F  23              6.0          INX     H               ;POINT TO NEXT DATA BYTE
2537  4850  7E              7.0          MOV     A,M             ;GET THE DATA
2538  4851  32  58  46     13.0          STA     SDDRBT          ;SAVE IN DDR REG B - BITE 15-8
2539  4854  23              6.0          INX     H               ;POINT TO THE NEXT DATA BYTE
2540  4855  7E              7.0          MOV     A,M             ;GET IT
2541  4856  32  59  46     13.0          STA     SDDRCT          ;SAVE IN DDR REG C - BITS "P" + 17-16
2542  4859  CD  E8  46     18.0          CALL    SETDDR          ;LOAD THE DDR REGS
2543
2544                                ;COUNT THE SCLK CYCLE (18 BIT WORD TRANSFERED)
2545
2546  485C  3A  53  46     13.0          LDA     CLKCNT          ;GET CURRENT CLOCK COUNT
2547  485F  3C              4.0          INR     A               ;UPDATE IT
2548  4860  32  53  46     13.0          STA     CLKCNT          ;SAVE UPDATED COUNT
2549  4863  FE  29            7.0          CPI     41              ;DONE WITH TRANSFER?
2550  4865  CA  6E  48       10.0          JZ      T3END           ;YES
2551
2552  4868  CD  68  46       18.0          CALL    CLOCKO          ;NO - CLOCK THE DATA LOADED IN DDR TO HOST
```

```
2553  486B  C3  4A  48  10.0          JMP     TST3C           ;AND LOOP BACK FOR THE NEXT WORD
2554                                ;HERE WHEN THE TEST IS FINISHED WITH THE DATA TRANSFER - TERMINATE
2555                                ;THE TRANSFER AND GIVE UP THE MASSBUS DATA LINES
2556
2557  486E  3E  B0   7.0  T3END:  MVI     A,M.WREN!M.OCC.M.EBL  ;SET DBUS CNTRL WORD TO
2558  4870  D3  C0  10.0          OUT     DBUSCTL         ;"MB WR EN" + "OCC" + "EBL"
2559  4872  3E  A0   7.0          MVI     A,M.WREN!M.OCC  ;DROP "EBL" SIGNAL TO MASSBUS
2560  4874  D3  C0  10.0          OUT     DBUSCTL
2561  4876  3E  00   7.0          MVI     A,0
2562  4878  D3  C0  10.0          OUT     DBUSCTL         ;THEN RELEASE THE MASSBUS
2563
2564                                ;REQUEST THE HOST CPU TO CHECK THE DATA JUST TRANSFERED TO HOST MEMORY
2565                                ;AND TO CHECK THE TERMINATION STATUS OF THE DATA TRANSFER FROM THE HOST
2566                                ;CPU SIDE OF THE MASSBUS. HOST CPU WILL REPORT ANY DETECTED ERROR AND
2567                                ;RESPOND WITH CONTINUE CODE (31) TO TERMINATE THIS TEST OR WITH THE
2568                                ;LOOP-ON-ERROR CODE (33) TO ISSUE THE TEST AGAIN.
2569
2570  487A                          REQ     5,1,40,@30004,1
 (1)  487A  CD  06  28  18.0                    CALL    REQST
 (1)  487D  01                                  .BYTE   1               ;DATA PATTERN NUMBER
 (1)  487E  00  28                              .WORD   40              ;SYSTEM "40" COUNT
 (1)  4880  30  04                              .WORD   @30004          ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  4882  01                                  .BYTE   1               ;DATA COMPARE FLAG IF =1
 (1)  4883  05                                  .BYTE   5               ;REQUEST CODE
2571                                ;CHECK WHAT RESPONSE CODE WAS RECEIVED BY THE MICROCOMPUTER...
2572                                ;ISSUE THE TEST AGAIN IF CODE 33 RECEIVED.
2573
2574  4884  3A  98  4F  13.0          LDA     VALFC           ;GET THE FUNCTION CODE RECEIVED
2575  4887  FE  1B   7.0          CPI     @33             ;LOOP-ON-ERROR CODE?
2576  4889  CA  C6  47  10.0          JZ      TST3L           ;YES - DO TEST AGAIN
2577  488C                          ENDTST  TST3L
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  488C                          REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  488C  CD  06  28  18.0                    CALL    REQST
 (2)  488F  00                                  .BYTE                   ;DATA PATTERN NUMBER
 (2)  4890  00  00                              .WORD                   ;SYSTEM "" COUNT
 (2)  4892  00  00                              .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4894  00                                  .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4895  07                                  .BYTE   7               ;REQUEST CODE
 (1)  4896  3A  9A  4F  13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4899  3D   4.0          DCR     A               ;DOWNCOUNT
 (1)  489A  32  9A  4F  13.0          STA     ITERA           ;SAVE COUNT
 (1)  489D  F2  C6  47  10.0          JP      TST3L           ;DO TEST UNTIL TILL = 0
2578  48A0  C3  B0  48  10.0          JMP     TEST4           ;DO THE NEXT TEST
2579                                ;HERE IS AN ERROR WAS DETECTED ON THE SCLK CYCLE - TERMINATE THE DATA
2580                                ;TRANSFER BY SENDING MASSBUS AN EBL AND EXC SIGNAL
2581  48A3  3E  88   7.0  QUITW:  MVI     A,M.WREN!M.EXC  ;SET DBUS CNTRL WORD TO
2582  48A5  D3  C0  10.0          OUT     DBUSCTL         ;SET EXC
2583  48A7  3E  18   7.0          MVI     A,M.EBL!M.EXC   ;SET BOTH EXC AND EBL
2584  48A9  D3  C0  10.0          OUT     DBUSCTL
2585  48AB  3E  00   7.0          MVI     A,0
2586  48AD  D3  C0  10.0          OUT     DBUSCTL         ;RELEASE THE MASSBUS
2587  48AF  C9  10.0          RET                     ;EXIT
```

```
                                .SBTTL  TEST 4 - READ DATA TRANSFER USING THE "DDR" REGISTERS - DDR IN "LEFT" MODE
4880                            ST
 (1)                            ; ****************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*READ DATA TRANSFER USING THE "DDR" REGISTERS - DDR IN "LEFT" MODE
4880                            SM
 (1)                            ; ****************************************************************************
 (1)                            ;*MODULE(S) UNDER TEST
 (1)                            ;*--------------------
                                ;*M8956 & M8959
4880                            SD
 (1)                            ; ****************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST REQUIRES THE HOST TO READ THE "MBDPAR" DATA PATTERN WHICH
                                ;*IS 40 18 BIT WORDS LONG. THE BYTE COUNT REGISTER (05) WILL BE LOADED
                                ;*TO 100 BYTES (10 DUMP MODE USED -- 5 BYTES PER WURD * 40 WORDS).
                                ;*
                                ;*EACH 18 BIT WORD WILL BE CHECKED TO THE HOST CPU OVER THE MASSBUS DATA
                                ;*LINES WITH DIAGNOSTIC "SCLK" SIGNAL.
                                ;*
                                ;*ANY MICROCOMPUTER DETECTED ERROR DETECTED DURING THE DATA TRANSFER WILL
                                ;*BE REPORTED TO THE HOST CPU.
                                ;*
                                ;*READ XFER TEST
4880                            SP
 (1)                            ; ****************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  INIT TEST-TESTX MACRO
                                ;*  ISSUE WMC RESTART TO INIT SYSTEM
                                ;*  SET WMC IN 'LEFT' MODE
                                ;*  IF WMC 'LEFT' IS SET
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT ERROR
                                ;*  ENDIF
                                ;*  INIT ECODE COUNTER TO 0 + SELECT WMC CONTROL TO 'WRITE' TO HOST CPU
                                ;*  REQUEST HOST TO START A READ FORWARD COMMAND @ 10 DUMP MODE WITH
                                ;*   'MBDPAR' DATA EXPECTED-40 SCLKS OF DATA SENT TO HOST CPU
                                ;*  CONTINUE WHEN HOST RESPONDS WITH A 71 FUNCTION CODE
                                ;*  INIT 'SCLK' COUNT TO '0' + SET MASSBUS 'OCC' + 'RUN' + 'MBWRTEN'
                                ;*  SET 'SCLK' + WAIT SMALL AMOUNT OF TIME
                                ;*  IF HOST RESPOND WITH 'WCLK'?
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT MASSBUS HANDSHAKE FAILURE ON DATA BUS-OPI ERROR
                                ;*  ENDIF
                                ;*  FINISH CLOCK CYCLE BY CLEARING 'SCLK' IN DBUS CONTROL
                                ;*  IF 'SCLK' GO AWAY?
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-REPORT MASSBUS STATUS ERROR
                                ;*  BGNDO
                                ;*  :  DO UNTIL 40 'SCLKS' OF DATA SENT
```

```
2631                                    ;*  :  IF XFR DONE-40 CLKS
2632                                    ;*  :  :  THEN-CONTINUE
2633                                    ;*  :  :  ELSE-
2634                                    ;*  :  ENDIF
2635                                    ;*  ENDDO
2636                                    ;*  TERMINATE HOST READ SFR BY SETTING 'EBL'-THEN TERMINATE BY DROPPING
2637                                    ;*  MASSBUS 'OCC' + 'RUN' BITS IN DBUS CONTROL
2638                                    ;*  REQUEST HOST CPU TO CHECK THE TERMINATION STATUS FROM READ XFR
2639                                    ;*  TERMINATE TEST IF HOST RESPONDS WITH A 31-CONTINUE CODE
2640                                    ;*  OR REPEAT TEST IF HOST RESPONDS WITH A 33-LOOP ON ERROR CODE
2641                                    ;*ENDTST
2642  48B0                              SE
 (1)                                    ; ************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
2643                                    ;*MBD1 MICRO TEST 04
2644                                    ;*MBD1 MICRO ERROR 21
2645                                    ;*MBD1-READ FROM TM78 TEST USING DIAG DATA REG.-"DDR" IN "LEFT" MODE
2646                                    ;*M8956, M8959
2647                                    ;*WMC FAILED TO SET "WMC LEFT" STATUS BIT IN DDR CONTROL BYTE
2648                                    ;*ACTUAL = NNNN
2649                                    ;*TEST ABORTED!
2650                                    ;*
2651                                    ;*MBD1 MICRO TEST 04
2652                                    ;*MBD1 MICRO ERROR 22
2653                                    ;*MBD1-READ FROM TM78 TEST USING DIAG DATA REG.-"DDR" IN "LEFT" MODE
2654                                    ;*M8956, M8959
2655                                    ;*OPI ERROR - HOST CPU FAILED TO SET "RUN" FOR A READ COMMAND (7X CODE)
2656                                    ;*TEST ABORTED.
2657  48B0                              S
 (1)                                    ; ************************************************************************
2658
2659  48B0                              TEST4:  TESTX   @4
 (1)  48B0  3E  04          7.0                          MVI     A,@4            ;DEFINE THE TEST NUMBER
 (1)  48B2  CD  03  28      18.0                         CALL    TSET            ;SETUP THE TEST
2660                                    ;%MBD1-READ FROM TM78 TEST USING DIAG. DATA REG.-"DDR" IN "LEFT" MODE
2661                                    ;&M8956, M8959
2662
2663  48B5  3E  B0           7.0  TST4L: MVI     A,@260          ;INIT THE "ECODE" COUNTER
2664  48B7  D3  D7          10.0        OUT     CNTCTL          ;SELECT THE COUNTER
2665  48B9                              CLRECT
 (1)  48B9  AF               4.0                  XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  48BA  D3  D6          10.0                  OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  48BC  D3  D6          10.0                  OUT     ERRCNT          ;CLEAR BITS 15-8
2666
2667                                    ;SET THE "DDR" CONTROL WORD TO "OUT" TRANSFER
2668
2669  48BE  3E  88           7.0        MVI     A,@210
2670  48C0  D3  DB          10.0        OUT     DDRCTL          ;INIT THE DDR CONTROL WORD
2671
2672                                    ;TRY TO SET WRITE MICRO "WMC LEFT" BIT
2673
2674  48C2  3E  39           7.0        MVI     A,@71           ;FMT=3 (10 DUMP), SKIP CNT = 11 (SET WMC LEFT)
2675  48C4  D3  D0          10.0        OUT     DATACTL         ;GO....
```

```
2676  48C6  3E  01          7.0          MVI   A,W.RST          ;RESTART THE WMC
2677  48C8  D3  D3         10.0          OUT   WMCCTL
2678  48CA  AF               4.0          XRA   A               ;NOW DROP WMC "RST" BIT
2679  48CB  D3  D3         10.0          OUT   WMCCTL
2680  48CD  00               4.0          NOP                   ;SMALL DELAY
2681  48CE  3E  80           7.0          MVI   A,W.ENAB        ;TELL WMC TO SET THE LEFT BIT
2682  48D0  D3  D3         10.0          OUT   WMCCTL          ;TRY NOW!
2683
2684  48D2  3E  28           7.0          MVI   A,40            ;WAIT FOR ABOUT 200 MICRO-SECONDS
2685  48D4  3D               4.0  T4WS:   DCR   A               ;DECREMENT THE COUNT
2686  48D5  C2  D4  48      10.0          JNZ   T4WS            ;WAIT A WHILE...
2687                                    ;NOW CHECK THAT THE WMC PGM REALLY SET "WMC LEFT" BIT SO LOADING DDRA-DDRC
2688                                    ;WILL BE DIRECT TRANSLATION TO MASSBUS BITS.
2689
2690  48D8  DB  A1          10.0          IN    CBUSSTA         ;GET CBUS STATUS (WMC LEFT BIT)
2691  48DA                                ROUT  ADATA           ;SAVE ACTUAL STATUS FOR ERROR REPORT
 (1)  48DA  D3  94          10.0                OUT   ADATA             ;WRITE AC INTO ADATA
 (1)  48DC  7F               4.0                MOV   A,A               ;RETRY LINK
2692  48DD  E6  04           7.0          ANI   W.LEFT          ;SAVE ONLY WMC LEFT BIT
2693  48DF  C2  F0  48      10.0          JNZ   TST4GO          ;CONTINUE IF ITS SET...
2694  48E2                                ERRA  TST4L,TST4LC
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  48E2  CD  0F  28      18.0                CALL  ERLPA             ;PROCESS ERROR - DO 2.3
 (1)       0011                             MSGN  =     MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48E5  11                               .BYTE MSGN              ;MESSAGE NUMBER ID
 (1)  48E6  00                               .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  48E7  CD  15  28      18.0          TST4LC::    CALL  CKLOP             ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48EA  DA  B5  48      10.0                JC    TST4L             ;LOOP ADDRESS IF LOOP SPECIFIED
2695                                    ;>WMC FAILED TO SET "WMC LEFT" STATUS BIT IN DDR CONTROL BYTE
2696                                    ;<TEST ABORTED!
2697  48ED  C3  18  28      10.0          JMP   TSTEND          ;ABORT RIGHT NOW!
2698                                    ;REQUEST THE HOST TO SETUP FOR A 100 BYTE READ TRANSFER (40 18 BIT WORDS)
2699                                    ;AND EXPECT FUNCTION CODE 71 RETURNED (READ FORWARD COMMAND). PACKING
2700                                    ;FORMAT IS 10 DUMP MODE.
2701
2702  48F0  3E  08           7.0  TST4GO: MVI   A,W.WRITE               ;NOW TELL THE WMC TO "WRITE" SO
2703  48F2  D3  D3          10.0          OUT   WMCCTL                  ;DATA REG (DR) WILL NOT BE ENABLED TO MASSBUS
2704  48F4                                REQ   3,1,40,@30004
 (1)  48F4  CD  06  28      18.0                CALL  REQST
 (1)  48F7  01                               .BYTE 1                 ;DATA PATTERN NUMBER
 (1)  48F8  00  28                           .WORD 40                ;SYSTEM "40" COUNT
 (1)  48FA  30  04                           .WORD @30004            ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  48FC  00                               .BYTE                   ;DATA COMPARE FLAG IF 1
 (1)  48FD  03                               .BYTE 3                 ;REQUEST CODE
2705
2706                                    ;INIT THE SCLK COUNTER TO 0 - TEST WILL QUIT WHEN COUNT IS 41.
2707
2708  48FE  3E  00           7.0          MVI   A,0             ;SET CLOCK TO '0' TICK
2709  4900  32  53  46      13.0          STA   CLKCNT          ;INIT THE CLOCK COUNTER
2710
2711                                    ;HOST IS READY FOR THE DATA TRANSFER...
2712                                    ;SET MASSBUS OCCUPIED (OCC), MASSBUS RUN (RUN), AND "MB WR EN"
2713
2714  4903  3E  A0           7.0          MVI   A,M.WREN!M.OCC
```

```
2715  4905  D3  C0          10.0          OUT     DBUSCTL          ;SET OCC + RUN + MB WR EN
2716
2717                                      ;NOW WAIT FOR HOST CPU TO ASSERT "RUN" ON THE MASSBUS...COULD TAKE UP TO
2718                                      ;10 MILLISEC (MASSBUS SPEC.)
2719
2720  4907  0E  14            7.0          MVI     C,20             ;LOOK FOR 100 MILLI-SEC.
2721  4909  DB  C0           10.0  T4W:    IN      DBUSSTA          ;GET DBUS STATUS - CHECK FOR "RUN"
2722  490B  E6  04            7.0          ANI     M.RUN            ;SAVE ONLY "RUN" BIT
2723  490D  C2  2B  49        10.0          JNZ     TST4WC           ;JUMP IF RUN IS UP - OK TO PROCEED
2724  4910  3E  FF            7.0  T4WL:   MVI     A,@377           ;GET A DELAY TIME
2725  4912  3D                4.0  T4WL1:  DCR     A                ;DECREMENT
2726  4913  C2  12  49        10.0          JNZ     T4WL1            ;STAY IN LOOP TILL =0
2727
2728                                      ;DOWNCOUNT GROSS TIMER COUNT
2729
2730  4916  0D                4.0          DCR     C
2731  4917  C2  09  49        10.0          JNZ     T4W              ;THEN CHECK IF "RUN" IS UP
2732  491A  CD  A3  48        18.0          CALL    QUITW
2733  491D                                  ERR     TSTEND,T4WLC
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  491D  CD  09  28        18.0                    CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0012                          MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4920  12                            .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  4921  00                            .BYTE
 (1)  4922  CD  15  28        18.0  T4WLC:: CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4925  DA  18  28        10.0          JC      TSTEND           ;LOOP ADDRESS IF LOOP SPECIFIED
2734                                      ;>OPI ERROR - HOST CPU FAILED TO SET "RUN" FOR A READ COMMAND (7X CODE)
2735                                      ;<TEST ABORTED!
2736  4928  C3  18  28        10.0          JMP     TSTEND           ;FATAL ERROR. - QUIT NOW
2737                                      ;POINT TO THE TABLE OF DATA TO TRANSFER
2738
2739  492B  21  48  47        10.0  TST4WC: LXI     H,DATATB-1       ;SETUP THE POINTER
2740
2741                                      ;SETUP FOR THE NEXT DATA TRANSFER WORD
2742
2743  492E  23                  6.0  TST4C:  INX     H                ;POINT TO NEXT DATA BYTE WORD
2744  492F  7E                  7.0          MOV     A,M              ;GET A BYTE OF DATA
2745  4930  32  57  46         13.0          STA     SDDRAT           ;SAVE DDR REG A - BITS 7-0
2746  4933  23                  6.0          INX     H                ;POINT TO NEXT DATA BYTE
2747  4934  7E                  7.0          MOV     A,M              ;GET THE DATA
2748  4935  32  58  46         13.0          STA     SDDRBT           ;SAVE IN DDR REG B - BITE 15-8
2749  4938  23                  6.0          INX     H                ;POINT TO THE NEXT DATA BYTE
2750  4939  7E                  7.0          MOV     A,M              ;GET IT
2751  493A  32  59  46         13.0          STA     SDDRCT           ;SAVE IN DDR REG C - BITS "P" + 17-16
2752  493D  CD  E8  46         18.0          CALL    SETDDR           ;LOAD THE DDR REGS
2753
2754                                      ;COUNT THE SCLK CYCLE (18 BIT WORD TRANSFERED)
2755
2756  4940  3A  53  46         13.0          LDA     CLKCNT           ;GET CURRENT CLOCK COUNT
2757  4943  3C                  4.0          INR     A                ;UPDATE IT
2758  4944  32  53  46         13.0          STA     CLKCNT           ;SAVE UPDATED COUNT
2759  4947  FE  29              7.0          CPI     41               ;DONE WITH TRANSFER? (STOP AFTER 40 SCLKS)
2760  4949  CA  52  49        10.0          JZ      T4END            ;YES
2761
```

```
2762  494C  CD  68  46   18.0          CALL    CLOCK0          ;NO - CLOCK THE DATA LOADED IN DDR TO HOST
2763  494F  C3  2E  49   10.0          JMP     TST4C           ;AND LOOP BACK FOR THE NEXT WORD
2764                                ;HERE WHEN THE TEST IS FINISHED WITH THE DATA TRANSFER - TERMINATE
2765                                ;THE TRANSFER AND GIVE UP THE MASSBUS DATA LINES
2766
2767  4952  3E  B0        7.0     T4END:  MVI     A,M.WREN!M.OCC.M.EBL  ;SET DBUS CNTRL WORD TO
2768  4954  D3  C0       10.0             OUT     DBUSCTL         ;"MB WR EN" + "OCC" + "EBL"
2769  4956  3E  A0        7.0             MVI     A,M.WREN!M.OCC  ;DROP "EBL" SIGNAL TO MASSBUS
2770  4958  D3  C0       10.0             OUT     DBUSCTL
2771  495A  3E  00        7.0             MVI     A,0
2772  495C  D3  C0       10.0             OUT     DBUSCTL         ;THEN RELEASE THE MASSBUS
2773
2774                                ;REQUEST THE HOST CPU TO CHECK THE DATA JUST TRANSFERED TO HOST MEMORY
2775                                ;AND TO CHECK THE TERMINATION STATUS OF THE DATA TRANSFER FROM THE HOST
2776                                ;CPU SIDE OF THE MASSBUS. HOST CPU WILL REPORT ANY DETECTED ERROR AND
2777                                ;RESPOND WITH CONTINUE CODE (31) TO TERMINATE THIS TEST OR WITH THE
2778                                ;LOOP-ON-ERROR CODE (33) TO ISSUE THE TEST AGAIN.
2779
2780  495E                          REQ     5,1,40,@30004,1
 (1)  495E  CD  06  28   18.0                   CALL    REQST
 (1)  4961  01                                  .BYTE   1               ;DATA PATTERN NUMBER
 (1)  4962  00  28                              .WORD   40              ;SYSTEM "40" COUNT
 (1)  4964  30  04                              .WORD   @30004          ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  4966  01                                  .BYTE   1               ;DATA COMPARE FLAG IF =1
 (1)  4967  05                                  .BYTE   5               ;REQUEST CODE
2781
2782                                ;CHECK WHAT RESPONSE CODE WAS RECEIVED BY THE MICROCOMPUTER...
2783                                ;ISSUE THE TEST AGAIN IF CODE 33 RECEIVED.
2784
2785  4968  3A  98  4F   13.0          LDA     VALFC           ;GET THE FUNCTION CODE RECEIVED
2786  496B  FE  1B        7.0          CPI     @33             ;LOOP-ON-ERROR CODE?
2787  496D  CA  B5  48   10.0          JZ      TST4L           ;YES - DO TEST AGAIN
2788  4970                           ENDTST  TST4L
 (1)                            ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4970                      REQ     /                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4970  CD  06  28   18.0                   CALL    REQST
 (2)  4973  00                                  .BYTE                   ;DATA PATTERN NUMBER
 (2)  4974  00  00                              .WORD                   ;SYSTEM "" COUNT
 (2)  4976  00  00                              .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4978  00                                  .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4979  07                                  .BYTE   7               ;REQUEST CODE
 (1)  497A  3A  9A  4F   13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  497D  3D        4.0          DCR     A               ;DOWNCOUNT
 (1)  497E  32  9A  4F   13.0          STA     ITERA           ;SAVE COUNT
 (1)  4981  F2  B5  48   10.0          JP      TST4L           ;DO TEST UNTIL TILL = 0
2789  4984  C3  18  28   10.0          JMP     TSTEND          ;EXIT - ALL DONE!
2790
2791        0000                    .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAV     4651          ASAVE    4F9B          ATTCD    4F97
AXNUM    4F91          B      =%0000          BADST  = 0090          BIT0   = 0001
BIT1   = 0002          BIT15  = 8000          BIT2   = 0004          BIT3   = 0008
BIT4   = 0010          BIT5   = 0020          BIT6   = 0040          BIT7   = 0080
BIT8   = 0100          BIT9   = 0200          BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60           BRKXCT= 4F00          BSAVE    4F9C          BYTCNT= 00D4
BYTEH    4F24          BYTEL    4F23          C      =%0001          CASCT    4F21
CASCTL= 00A0           CASSTA= 00A0          CATTH  = 0089          CATTL  = 0088
CBUSST= 00A1           CBYTH  = 008B          CBYTL  = 008A          CDG1H  = 0087
CDG1L  = 0086          CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095
CDG3L  = 0094          CDVTH  = 008D          CDVTL  = 008C          CHPTIE= 0028
CHOTIE= 0020           CH1TIE= 0021          CH2TIE= 0022          CH3TIE= 0023
CH4TIE= 0024           CH5TIE= 0025          CH6TIE= 0026          CH7TIE= 0027
CKLOP  = 2815          CLKCNT   4653         CLKCTL= 00F0          CLKSKP   4683
CLOCK    4F26          CLOCK0   4668         CLOCK1   4677         CLOCK2   4690
CMC0H  = 0099          CMC0L  = 0098          CMC1H  = 009B          CMC1L  = 009A
CMC2H  = 009D          CMC2L  = 009C          CMC3H  = 009F          CMC3L  = 009E
CMINH  = 0097          CMINL  = 0096          CNTCTL= 00D7          COMCK    45BA
CRCWRD= 0018           CSAVE    4F9D          CSRLH  = 0091          CSRLL  = 0090
CTCH   = 0085          CTCL   = 0084          CTSTH  = 008F          CTSTL  = 008E
CXCTH  = 0081          CXCTL  = 0080          CXINH  = 0083          CXINL  = 0082
C.     = 0001          C.AVAI= 0080          C.DP   = 0008          C.DSE  = 0010
C.DTU  = 0003          C.DVA  = 0008          C.FAIL= 00FC          C.FMT  = 0070
C.FNCT= 003E           C.GO   = 0001          C.INTC= 00FE          C.MAIN= 0020
C.NSA  = 0080          C.RCT  = 00FC          C.SER  = 0080          C.SHR  = 0040
C.SKPC= 000F           C.TAPE= 0040          C.WCS  = 0002          D      =%0002
DATACT= 00D0           DATATB   4749         DBUS     4F28          DBUSCT= 00C0
DBUSST= 00C0           DDRA   = 00D8          DDRACK   45AC         DDRAIN= 0010
DDRB   = 00D9          DDRBCK   45C3         DDRBIN= 0002          DDRC   = 00DA
DDRCCK   45FD          DDRCED   4621         DDRCER   465D         DDRCEX   461C
DDRCIN= 0001           DDRCO  = 0088          DDRCTL= 00DB          DIAFLG   4F22
DIAGPG= 4300           DIAGRM= 4F90          DIARD  = 000B          DONE1  = 0045
DONINT= 0010           DSAVE    4F9E          DSE    = 0006          DTEND    47C0
D.ATH0= 0001           D.ATH1= 0002          D.EOTD= 0010          D.LAGC= 0020
D.NOTW= 0040           D.NTHR= 0004          D.TACH= 0008          D.WR4  = 0080
E      =%0003          ECCBAD= 0042          ECCCOR= 0019          ECCOK  = 0041
ECCSTA= 001A           ECCTST= 000E          EDATA  = 0095          EOTCLR= 0003
ERFLG    4F93          ERLP   = 2809          ERLPA  = 280F          ERLPB  = 2812
ERLPE  = 280C          ERNUM    4F90         ERRCNT= 00D6          ESAVE    4F9F
E.ACRC= 0010           E.AMT  = 0020          E.CDP  = 0080          E.CRC  = 0080
E.PNTR= 0008           E.RPE  = 0040          E.STEC= 0001          E.TTEC= 0002
E.UNC  = 0004          FIFORD= 006A          FORMAT   4F25         FWDTST= 0061
GCRID  = 0089          GCRSET= 0002          GETDDR   4695         GOODTM= 0092
H      =%0004          HLSAVE   4FA0         IE     = 0008          INTSTA= 00E0
ITERA    4F9A          I.PWR  = 0020          I.RMPE= 0040          I5.5   = 0010
I6.5   = 0020          I7.5   = 0040          KCALL  = 005F         KCLR   = 007B
KDEP   = 003F          KENAB  = 0078          KEXAM  = 003E         KEYBRD= 00C8
KEY1   = 0078          KEY10  = 006D          KEY11  = 006E         KEY12  = 006F
KEY13  = 005C          KEY14  = 005D          KEY15  = 005E         KEY16  = 005F
KEY17  = 003C          KEY18  = 003D          KEY19  = 003E         KEY2   = 0079
KEY20  = 003F          KEY3   = 007A          KEY4   = 0077         KEY5   = 0074
KEY6   = 0075          KEY7   = 0076          KEY8   = 0077         KEY9   = 006C
KINTA  = 006F          KLDAD  = 003D          KNO    = 003C         KN1    = 005C
```

```
KN2    = 005D          KN3    = 005E          KN4    = 006C          KN5    = 006D
KN6    = 006E          KN7    = 0074          KN8    = 0075          KN9    = 0076
KU2    = 0079          KU3    = 007A          KU8    = 0077          L      =%0005
LPBLANK= 000F          LCE    = 000B          LCH    = 000C          LCL    = 000D
LC?    = 000E          LC0    = 0000          LC1    = 0001          LC2    = 0002
LC3    = 0003          LC4    = 0004          LC5    = 0005          LC6    = 0006
LC7    = 0007          LC8    = 0008          LC9    = 0009          LDLEDA= 00CA
LDLEDB= 00CB           LDLEDC= 00CC           LDLEDD= 00CD           LDLEDE= 00CE
LDLEDF= 00CF           LD18BI  45D4           LKDIAG= 2800           LKKBD  = 004C
LKKEY  = 0049          LKLWMG= 0058           LKLWMP= 0055           LKLWPG= 0052
LKLWPP= 004F           LKMOD7= 0046           LKOPR  = 0046          LPFLG   4F94
LPNUM   4F92           M      =%0006          MBSEL  = 00E0          MB.A   = 0008
MB.B   = 0004          MEMTOP= 4FFF           MINUS  = 000A          MM     = 8000
MSE    = 0008          MSGN   = 0012          MTACLR= 0000           MT.ARA= 0020
MT.CPE= 0080           MT.DSE= 0001           MT.FWD= 0040           MT.INH= 0008
MT.LWR= 0004           MT.MOT= 0002           MT.NWT= 0080           MT.PEC= 0040
MT.PSB= 0004           MT.PS0= 0001           MT.PS1= 0002           MT.REV= 0020
MT.WRT= 0010           MT.Z   = 0008          M.ATA  = 0080          M.CAPE= 0020
M.CONT= 0080           M.DEM  = 0020          M.EBL  = 0010          M.EXC  = 0008
M.FAIL= 0008           M.ILR  = 0010          M.INIT= 0010           M.OCC  = 0020
M.ONLI= 0001           M.PE   = 0040          M.PORT= 0080           M.RDEN= 0002
M.RDPE= 0008           M.RUN  = 0004          M.SCLK= 0001           M.TRA  = 0040
M.UNIT= 0007           M.WCLK= 0040           M.WCLN= 0080           M.WREN= 0080
M5.5   = 0001          M6.5   = 0002          M7.5   = 0004          NOTCAP= 0088
OKAY   = 00FF          OPRRAM= 4300           OPSTRT= 0058           OPVER  = 0040
PADCNT= 00D5           PADCRC= 0080           PBIT    4652           PDIAG  = 0048
PEID   = 008A          PENAB  = 004C          PESET  = 0001          PL     = 00B1
PNMOD   4643           PRDD   = 004C          PRENF  = 009C          PS     = 00B2
PSTAT  = 0048          PSW    =%0009          P.AMTP= 0001           P.BCTC= 0040
P.CMDP= 0020           P.INTE= 0080           P.LCS  = 0040          P.LWR  = 0020
P.RDP  = 0002          P.RPEN= 0004           P.RPST= 0002           P.RPOE= 0020
P.RP1E= 0010           P.RP2E= 0020           P.RP3E= 0010           P.SING- 0080
P.STAT= 0002           P.STPE= 0080           P.TACH= 0008           P.TUPR= 0010
P.WCSP= 0004           P.WDS  = 0040          P.WFLP= 0001           P.WPEN= 0010
P.WPOE= 0008           P.WP1E= 0004           P.WP2E= 0008           P.WP3E= 0004
P.SVOK= 0002           QUE    = 281B          QUEM   = 281E          QUIT    4580
QUITW   48A3           QUIT1   45A1           RAMT   = 0010          RARA   = 0006
RARAI  = 0004          RCHBD0= 0048           RCHBD1- 0047           RCHOK  = 0046
RCHST= 000C            RCLRT  = 000D          RCMD   = 000B          RCMLP  = 0003
RCONT  = 0080          RDATA  = 0017          RDCLK  = 0010          RDON   = 0011
READG  = 0007          REND   = 0014          REQST  = 2806          RESCHR= 00D1
REVTST= 0064           REWIND= 0004           RFIFOL= 0008           RGCLK  = 0002
RGCRI  = 0003          RIBG   = 0001          RILL   = 0012          RINST  = 000C
RMCTST= 0008           RMK2   = 0013          RNOP   = 0000          RPATH  = 0001
RPBAD  = 0044          RPCHI  = 0001          RPCLK  = 0003          RPCTL  = 0009
RPEI   = 0002          RPFAIL= 0000           RPF1   = 009D          RPF2   = 009E
RPOK   = 0043          RPOSTN= 0016           RPSTA  = 0015          RRCMT  = 000A
RSTAT  = 0002          RTIEB  = 000A          RTIER  = 0030          RTM    = 0005
RUNKI  = 0009          RUPTST= 005E           RWDUNL= 0005           R.AMT  = 0001
R.BOP  = 0008          R.DATA= 0040           R.DON  = 0002          R.DRDY= 0010
R.END  = 0010          R.ILL  = 0004          R.JVOK= 0004           R.MK2  = 0008
R.PLOD= 0008           R.PLO0= 0010           R.PLO1= 0020           R.POST= 0020
R.STNM= 0002           R.STOP= 0004           R.STPC= 0001           R.TBJN= 0080
R.TSTD= 0040           R.VOK  = 0080          ROOH   = 0081          ROOL   = 0080
```

```
R01H  = 0083          R01L  = 0082          R02H  = 0085          R02L  = 0084
R03H  = 0087          R03L  = 0086          R04H  = 0089          R04L  = 0088
R05H  = 008B          R05L  = 008A          R06H  = 008D          R06L  = 008C
R07H  = 008F          R07L  = 008E          R10H  = 0091          R10L  = 0090
R11H  = 0093          R11L  = 0092          R12H  = 0095          R12L  = 0094
R13H  = 0097          R13L  = 0096          R14H  = 0099          R14L  = 0098
R15H  = 009B          R15L  = 009A          R16H  = 009D          R16L  = 009C
R17H  = 009F          R17L  = 009E          R7.5  = 0010          SDDCOM  4739
SDDRA   4654          SDDRAE  465A          SDDRAT  4657          SDDRB   4655
SDDRBE  465B          SDDRBT  4658          SDDRC   4656          SDDRCE  465C
SDDRCT  4659          SELCLR= 0000          SETATA= 00A1          SETDDR  46E8
SID   = 0080          SLFTDD  4727          SOD   = 0080          SOE   = 0040
SP    =%0008          SSCLK = 0040          SSTEP = 0005          STACK = 4FFF
STATRM= 4F20          STPCT   4F20          STRSP = 5000          TADR00= 0080
TADR01= 0081          TADR02= 0082          TADR03= 0083          TADR04= 0084
TADR05= 0085          TADR06= 0086          TADR07= 0087          TADR10= 0088
TADR11= 0089          TADR12= 008A          TADR13= 008B          TAMT  = 0044
TASEL = 0080          TCMD  = 0040          TC.FWD= 0040          TC.INH= 0008
TC.LWR= 0004          TC.REV= 0020          TC.WRT= 0010          TEMP    4F99
TEST1   4300          TEST2   4464          TEST3   47C1          TEST4   48B0
TMF   = 0099          TMRDY = 0040          TRKENA= 00D2          TSET  = 2803
TSTEND= 2818          TSTS  = 0040          TST1AC  433A  G       TST1B   4340
TST1C   437B          TST1CA  4397  G       TST1CB  43A0          TST1CC  43BF  G
TST1CD  43E3  G       TST1D   43EC          TST1DC  43FA  G       TST1E   4403
TST1EC  4411  G       TST1F   441A          TST1FC  4429  G       TST1G   4432
TST1L   4305          TST2B   44A3          TST2C   44DE          TST2CD  44FC  G
TST2D   4505          TST2DC  4513  G       TST2E   451C          TST2EC  452A  G
TST2F   4533          TST2FC  4542  G       TST2G   454B          TST2GO  449F
TST2L   4469          TST2LC  4496  G       TST3A   4818          TST3B   4829
TST3BB  482D          TST3BC  4841  G       TST3C   484A          TST3GO  47DA
TST3L   47C6          TST3WC  4815          TST4C   492E          TST4GO  48F0
TST4L   48B5          TST4LC  48E7  G       TST4WC  492B          TUSEL0= 00D1
TUSEL1= 00D2          TU78  = 0010          T.ATH0= 0001          T.ATH1= 0002
T.BOT = 0004          T.EOT = 0002          T.FPT = 0001          T.NTHR= 0004
T.ONL = 0020          T.PES = 0008          T.PSBJ= 0020          T.PSOJ= 0008
T.PS1J= 0010          T.RDY = 0080          T.RDYO= 0040          T.RWD = 0010
T.SCLK= 0002          T1CHK   43C8          T1END   4444          T1W     4359
T1WL    4360          T1WLC   4372  G       T1WL1   4362          T2CHK   44E1
T2END   455D          T2W     44BC          T2WL    44C3          T2WLC   44D5  G
T2WL1   44C5          T2WS    4483          T3END   486E          T3W     47F3
T3WL    47FA          T3WLC   480C  G       T3WL1   47FC          T4END   4952
T4W     4909          T4WL    4910          T4WLC   4922  G       T4WL1   4912
T4WS    48D4          UIBG  = 00A1          VALFC   4F98          VALTB   4F95
VELTST= 005B          WDR.P = 0010          WMCCTL= 00D3          WMCERR= 00DA
WMCSTA= 00D0          WRTCLK= 0000          WRTDAT= 00D3          W.ACRC= 0004
W.CRC = 0008          W.DIAG= 0002          W.DONN= 0040          W.ECC = 0010
W.ENAB= 0080          W.ERR = 0020          W.FMT = 0070          W.GCR = 0010
W.LEFT= 0004          W.ONES= 0020          W.RESI= 00C2          W.REV = 0004
W.ROME= 0010          W.RST = 0001          W.SKIP= 000F          W.WRIT= 0008
W.XFER= 0020          X     =%000A          X.DONN= 0080          X.ENAB= 0040
X.PEPE= 0002          X.ROME= 0001          X.WCLK= 0001          Y     =%000B
.     = 4987
```

```
ERRORS DETECTED:  0

*MBD1.A78/PTP,MBD1=NLIST,PARAM,MACRO,LIST,MBD1
RUN-TIME: 5 7 0 SECONDS
CORE USED:  10K
```

3	"MEMORY MAP" EXTENDED ADDRESS BIT
19	READ MICRO CONTROLLER CONTROL & STATUS WORDS
139	TRANSPORT/PORT COMMAND + STATUS WORDS
380	COMMON ADDRESS SPACE REGISTERS
444	MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485	MASSBUS CONTROL WORD DEFINITIONS
557	WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709	KEYBOARD/DISPLAY CONTROL + STATUS WORD
797	MISCELLANEOUS VARIABLE DEFINITIONS
801	RAM ADDRESS DEFINITIONS
819	LINKAGE TABLE ADDRESSES
842	BIT DEFINITIONS
856	RIM AND SIM INSTRUCTION BIT DEFINITIONS
877	INTERRUPTABLE INSTRUCTION MACROS
924	DIAGNOSTIC VARIABLES
945	COMMON VARIABLE STORAGE
959	MACRO CALL DEFINITIONS
1047	TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101	8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1[illegible]	TEST 1 - WMC DATA FORMAT/SKIP COUNT ACCEPTANCE/REJECTION
1468	TEST 2 - WMC FORMAT/SKIP COUNT CHECKS
1640	SKIPSUB - SUBROUTINE FOR TEST 2

```
                ;*************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;                "BYTE/SCLK COUNT NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                MM = DATA FORMAT FROM CAS REGISTER 2
                ;                NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;                BYTE-SCLK COUNT = LLL
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                LLL = AS ABOVE
                ;                MM  = AS ABOVE
                ;                NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;                TRANSITION COUNT = LLL
                ;                WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;                EXPECTED 18 BITS =      E EEEEEE
                ;                ACTUAL 18 BITS =        A AAAAAA
                ;
                ;                WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;                BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;                OF CAS REG 15 LOW BYTE.
                ;
                ;                ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;                AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;                17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;                TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;                AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;                "SUBGROUP NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;*************************************************************************
```

```
1016                    ;*************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ;*************************************************************************
1035                    ;*************************************************************************
1036                    ;DATA PATTERN CODES
1037
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ;*************************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  WMC2 - WRITE MICROCONTROLLER PART #2
                                .SBTTL  TEST 1 - WMC DATA FORMAT/SKIP COUNT ACCEPTANCE/REJECTION
                                ;ID     WMC2-WRITE MICRO CONTROLLER PART #2
1335  4300                      ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1336                            ;*WMC FORMAT/SKIP COUNT TESTS
1337  4300                      SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1338                            ;*THIS TEST CHECKS THE OPERATION OF THE WRITE MICRO CONTROLLER IN
1339                            ;*RESPECT TO ACCEPTANCE OF THE SIX (6) DATA FORMATS.  THIS TEST ALSO
1340                            ;*VERIFIES THAT THE FORMATS ARE TAKEN WITHOUT REGARD FOR THE SKIP COUNT
1341                            ;*SPECIFIED.  THE FOLLOWING TABLE LISTS THE COMBINATIONS USED:
1342                            ;*
1343                            ;*      FORMAT          SKIP CNT            EXP. ERR.
1344                            ;*
1345                            ;*        0               0-17                NONE
1346                            ;*        1               0-17                NONE
1347                            ;*        2               0-17                NONE
1348                            ;*        3               0-17                NONE
1349                            ;*        4               0-17                NONE
1350                            ;*        5               0-17                NONE
1351  4300                      SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1352                            ;*BGNTST
1353                            ;*  INIT BYTE COUNTER, PAD COUNTER, & ERROR  CODE COUNTER CONTROL
1354                            ;*  SET SYSTEM CLOCK TO 'NORMAL'
1355                            ;*  CLEAR FORMAT COUNT INDEX
1356                            ;*  BGNDO
1357                            ;*  :  CLEAR SKIP COUNT INDEX
1358                            ;*  :  BGNDO
1359                            ;*  :  :  RESTART THE WMC + SYSTEM
1360                            ;*  :  :  WRITE CURRENT FORMAT 'COUNT' TO DATACTL WORD - SELECT PACKING MODE
1361                            ;*  :  :  WRITE REGISTER 02 WITH FORMAT NUMBER
1362                            ;*  :  :  CLEAR BYTE COUNTER, PAD COUNTER, AND ECODE COUNTER
1363                            ;*  :  :  CLEAR RESIDUAL CHARACTER WORD - 'RESCHR'
1364                            ;*  :  :  SET WMCCTL TO 'WRITE' AND 'ENABLE' TO START WMC ROM PROGRAM
1365                            ;*  :  :  SMALL DELAY TO WAIT FOR WMC ROM PGM TO FINISH
1366                            ;*  :  :  IF WMCSTA INDICATES 'DONE'
1367                            ;*  :  :  :  THEN-CONTINUE
1368                            ;*  :  :  :  ELSE-REPORT TIMEOUT ERROR
1369                            ;*  :  :  ENDIF
1370                            ;*  :  :  IF WMCERR HAS NO ERROR BITS SET
1371                            ;*  :  :  :  THEN-CONTINUE
1372                            ;*  :  :  :  ELSE-REPORT ERROR DETECTED AFTER A VALID WRITE COMMAND
1373                            ;*  :  :  ENDIF
1374                            ;*  :  :  DO UNTIL SKIP COUNT INDEX=17(8)
1375                            ;*  :  ENDDO
1376                            ;*  :  DO UNTIL FORMAT COUNT INDEX=6
```

```
                                        ;*  ENDDO
                                        ;*ENDTST
4300                                    SE
 (1)                                    ; *****************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
                                        ;*WMC2 MICRO TEST 01
                                        ;*WMC2 MICRO ERROR 01
                                        ;*WMC2-WMC DATA FORMAT/SKIP COUNT ACCEPTANCE
                                        ;*M8959
                                        ;*WMC NOT DONE AFTER A VALID WRITE 'PE' COMMAND ISSUED
                                        ;*DATA FORMAT = MM
                                        ;*SKIP COUNT  = NN
                                        ;*
                                        ;*WMC2 MICRO TEST 01
                                        ;*WMC2 MICRO ERROR 02
                                        ;*WMC2-WMC DATA FORMAT/SKIP COUNT ACCEPTANCE
                                        ;*M8959
                                        ;*WMC ERRORS BYTE NON ZERO AFTER A VALID WRITE PE COMMAND
                                        ;*ACTUAL = NNNN
                                        ;*DATA FORMAT = MM
                                        ;*SKIP COUNT  = NN
4300                                    S
 (1)                                    ; *****************************************************************************

4300                                    TEST1:  TESTX   @1
 (1) 4300  3E  01          7.0                   MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1) 4302  CD  03  28     18.0                   CALL    TSET            ;SETUP THE TEST
                                        ;%WMC2-WMC DATA FORMAT/SKIP COUNT ACCEPTANCE
                                        ;&M8959

4305  3E  38                7.0  TST1L:  MVI     A,@70           ;SET BYTE COUNTER FOR 0 LENGTH RECORD
4307  D3  D7               10.0          OUT     CNTCTL
4309  3E  70                7.0          MVI     A,@160          ;SELECT PAD COUNTER
430B  D3  D7               10.0          OUT     CNTCTL
430D  3E  B0                7.0          MVI     A,@260          ;SELECT ERROR CODE COUNTER
430F  D3  D7               10.0          OUT     CNTCTL

4311  AF                    4.0          XRA     A               ;SET THE SYSTEM CLOCK TO 'NORMAL'
4312  D3  F0               10.0          OUT     CLKCTL          ;WRITE THE CLOCK CONTROL WORD
4314  3E  9B                7.0          MVI     A,@233          ;SET THE DDR TO 'IN'
4316  D3  DB               10.0          OUT     DDRCTL

4318  0E  00                7.0          MVI     C,$0            ;CLEAR THE FORMAT COUNT
431A  06  00                7.0  T1LP0:  MVI     B,$0            ;CLEAR THE SKIP COUNT
431C  3E  01                7.0  T1LP1:  MVI     A,W.RST         ;LOAD THE WMC RESTART BIT
431E  D3  D3               10.0          OUT     WMCCTL          ;RESTART THE WMC
4320  AF                    4.0          XRA     A               ;CLEAR THE ACCUMULATOR
4321  D3  D3               10.0          OUT     WMCCTL          ;CLEAR THE RESTART BIT
4323  79                    4.0          MOV     A,C             ;COPY FORMAT COUNT
4324  07                    4.0          RLC                     ;SHIFT THE FORMAT OVER
4325  07                    4.0          RLC                     ;SHIFT THE FORMAT OVER
4326  07                    4.0          RLC                     ;3 BIT POSITIONS
4327  07                    4.0          RLC                     ;
```

```
1425  4328  E6  70          7.0          ANI   $70              ;REMOVE UNWANTED BITS
1426  432A  B0              4.0          ORA   B               ;OR IN THE SKIP COUNT
1427  432B  D3  D0         10.0          OUT   DATACTL         ;SET UP THE WRITE MICRO
1428                                     ;WITH FORMAT/SKIP INFO.
1429
1430  432D                               ROUT  R02H            ;LOAD INTO REG 02 HIGH BYTE
 (1)  432D  D3  85         10.0                OUT   R02H              ;WRITE AC INTO R02H
 (1)  432F  7F              4.0                MOV   A,A               ;RETRY LINK
1431  4330  AF              4.0          XRA   A               ;AND CLEAR REG 02 LOW BYTE
1432  4331                               ROUT  R02L
 (1)  4331  D3  84         10.0                OUT   R02L              ;WRITE AC INTO R02L
 (1)  4333  7F              4.0                MOV   A,A               ;RETRY LINK
1433  4334                               CLRECT                ;CLEAR THE ERROR COUNTER
 (1)  4334  AF              4.0                XRA   A                 ;CLEAR THE ACCUMULATOR
 (1)  4335  D3  D6         10.0                OUT   ERRCNT            ;CLEAR BITS 7-0
 (1)  4337  D3  D6         10.0                OUT   ERRCNT            ;CLEAR BITS 15-8
1434  4339                               CLRBCT                ;CLEAR THE BYTE COUNTER
 (1)  4339  AF              4.0                XRA   A                 ;CLEAR THE ACCUMULATOR
 (1)  433A  D3  D4         10.0                OUT   BYTCNT            ;CLEAR BITS 7-0
 (1)  433C  D3  D4         10.0                OUT   BYTCNT            ;CLEAR BITS 15-8
1435  433E                               CLRPCT                ;CLEAR THE PAD COUNTER
 (1)  433E  AF              4.0                XRA   A                 ;CLEAR THE ACCUMULATOR
 (1)  433F  D3  D5         10.0                OUT   PADCNT            ;CLEAR BITS 7-0
 (1)  4341  D3  D5         10.0                OUT   PADCNT            ;CLEAR BITS 15-8
1436  4343  AF              4.0          XRA   A               ;CLEAR THE ACCUMULATOR
1437  4344  D3  D1         10.0          OUT   RESCHR          ;CLEAR THE RESIDUAL CHARACTER
1438
1439  4346  3E  88          7.0          MVI   A,W.ENAB.W.WRITE
1440  4348  D3  D3         10.0          OUT   WMCCTL          ;ENABLE WMC-WRITE & PE
1441                                     ;
1442  434A  3E  32          7.0          MVI   A,50            ;SET THE TIMEOUT VALUE
1443  434C  3D              4.0  T1W:    DCR   A               ;DECREMENT THE COUNT
1444  434D  C2  4C  43      10.0          JNZ   T1W             ;CONTINUE UNTIL END OF COUNT
1445
1446  4350  DB  D0         10.0          IN    WMCSTA          ;GET THE WMC STATUS
1447  4352  E6  40          7.0          ANI   W.DONN          ;WMC NOT DONE BIT SET?
1448  4354  CA  5C  43      10.0          JZ    T1CO2           ;NO - CONTINUE
1449  4357                               ERR   T1LP1,T1CO2,@2
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4357  CD  09  28      18.0                CALL  ERLP              ;PROCESS ERROR - DO 2.3
 (1)        0001                              MSGN  =     MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  435A  01                                .BYTE MSGN              ;MESSAGE NUMBER ID
 (1)  435B  02                                .BYTE @2
 (1)  435C  CD  15  28      18.0          T1CO2:: CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  435F  DA  1C  43      10.0                JC    T1LP1           ;LOOP ADDRESS IF LOOP SPECIFIED
1450                           ;>WMC NOT DONE AFTER A VALID WRITE 'PE' COMMAND ISSUED
1451  4362  DB  DA         10.0          IN    WMCERR          ;GET THE WMC ERRORS BYTE
1452  4364  E6  F8          7.0          ANI   $F8             ;TEST FOR ERROR
1453  4366  CA  71  43      10.0          JZ    T1CN8           ;NONE-CONTINUE
1454  4369                               ROUT  ADATA           ;STORE THE ACTUAL WMC ERROR BYTE
 (1)  4369  D3  94         10.0                OUT   ADATA             ;WRITE AC INTO ADATA
 (1)  436B  7F              4.0                MOV   A,A               ;RETRY LINK
```

```
1456  436C                                        ERRA    T1LP1,T1CN8,@2
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  436C  CD  0F  28        18.0                        CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0002                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  436F  02                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4370  02                                            .BYTE   @2              ;PRINT ROUTINE NUMBER
 (1)  4371  CD  15  28        18.0                T1CN8:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4374  DA  1C  43        10.0                        JC      T1LP1           ;LOOP ADDRESS IF LOOP SPECIFIED
1457                                        ;>WMC ERRORS BYTE NON ZERO AFTER A VALID WRITE PE COMMAND
1458  4377  04                    4.0               INR     B                       ;UPDATE THE SKIP COUNT
1459  4378  78                    4.0               MOV     A,B                     ;
1460  4379  FE  10                7.0               CPI     @20                     ;DONE?
1461  437B  C2  1C  43        10.0                  JNZ     T1LP1                   ;NO - RUN NEXT SKIP COUNT TEST
1462  437E  0C                    4.0               INR     C                       ;INCREMENT THE FORMAT COUNT
1463  437F  79                    4.0               MOV     A,C                     ;
1464  4380  FE  06                7.0               CPI     $6                      ;DONE?
1465  4382  C2  1A  43        10.0                  JNZ     T1LP0                   ;NO-CONTINUE
1466  4385                                          ENDTST  TST1L
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4385                                          REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4385  CD  06  28        18.0                        CALL    REQST
 (2)  4388  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2)  4389  00  00                                        .WORD                   ;SYSTEM '*' COUNT
 (2)  438B  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  438D  00                                            .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  438E  07                                            .BYTE   7               ;REQUEST CODE
 (1)  438F  3A  9A  4F        13.0                        LDA     ITERA           ;GET ITERATION COUNT
 (1)  4392  3D                    4.0                     DCR     A               ;DOWNCOUNT
 (1)  4393  32  9A  4F        13.0                        STA     ITERA           ;SAVE COUNT
 (1)  4396  F2  05  43        10.0                        JP      TST1L           ;DO TEST UNTIL TILL = 0
```

```
                        .SBTTL  TEST 2 - WMC FORMAT/SKIP COUNT CHECKS
4399                    ST
                        ;*****************************************************************************
                        ;*TEST TITLE
                        ;*---- -----
                        ;*WMC FORMAT/SKIP COUNT ERROR CHECKING
4399                    SD
                        ;*****************************************************************************
                        ;*DESCRIPTION
                        ;*-----------
                        ;*THIS TEST CHECKS THE OPERATION OF THE WRITE MICRO CONTROLLER IN
                        ;*RESPECT TO ACCEPTANCE AND REJECTION OF VARIOUS DATA FORMAT, SKIP
                        ;*COUNT OPERATIONS IN THE READ MODE.  THE RESULTS EXPECTED ARE
                        ;*LISTED BELOW:
                        ;*
                        ;*      FORMAT                  SKIP CNT                EXP. ERR.
                        ;*       0                        0-1                     NONE
                        ;*       0                        2-17                     -2
                        ;*
                        ;*       1                        0-1                     NONE
                        ;*       1                        2-17                     -2
                        ;*
                        ;*       2                        0-4                     NONE
                        ;*       2                        5-17                     -1
                        ;*
                        ;*       3                        0-4                     NONE
                        ;*       3                        5-17                     -1
                        ;*
                        ;*       4                        0-10                    NONE
                        ;*       4                        11-17                    -1
                        ;*
                        ;*       5                        0-17                    NONE
                        ;*
                        ;*       6                        0-10                    NONE
                        ;*       6                        11-17                    -1
4399                    SP
                        ;*****************************************************************************
                        ;*PROCEDURE
                        ;*---------
                        ;*BGNTST
                        ;*  SET PACKING MODE NUMBER TO 0 (11-NORMAL)
                        ;*  SET SKIP COUNT ERROR LIMIT TO 2
                        ;*  SET THE ECODE EXPECTED TO -2
                        ;*  CALL SKIPSU3
                        ;*
                        ;*  SET PACKING MODE NUMBER TO 1 (15-NORMAL)
                        ;*  SET SKIP COUNT ERROR LIMIT TO 2
                        ;*  SET THE ECODE EXPECTED TO -2
                        ;*  CALL SKIPSUB
                        ;*
                        ;*  SET PACKING MODE NUMBER TO 2 (10-COMPAT)
                        ;*  SET SKIP COUNT ERROR LIMIT TO 4
                        ;*  SET THE ECODE EXPECTED TO -1
                        ;*  CALL SKIPSUB
```

```
                                ;*
                                ;*  SET THE PACKING MODE NUMBER TO 3 (10-DUMP)
                                ;*  SET SKIP COUNT ERROR LIMIT TO 5
                                ;*  SET THE ECODE EXPECTED TO -1
                                ;*  CALL SKIPSUB
                                ;*
                                ;*  SET PACKING MODE NUMBER TO 4 (10-HIGH DENSITY DUMP)
                                ;*  SET SKIP COUNT ERROR LIMIT TO 11(8)
                                ;*  SET THE ECODE EXPECTED TO -1
                                ;*  CALL SKIPSUB
                                ;*
                                ;*  SET PACKING MODE NUMBER TO 5 (IMAGE)
                                ;*  SET SKIP COUNT ERROR LIMIT TO 0 - SKIP COUNTS CAUSE NO ERROR!
                                ;*  CLEAR ECODE EXPECTED VALUE
                                ;*  CALL SKIPSUB
                                ;*
                                ;*  SET PACKING MODE NUMBER TO 6 (10-HIGH DENSITY COMPAT)
                                ;*  SET SKIP COUNT ERROR LIMIT TO 11(8)
                                ;*  SET ECODE EXPECTED TO -1
                                ;*  CALL SKIPSUB
                                ;*ENDTST
4399                            SE
 (1)                            ; **************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*WMC2 MICRO ERROR 02
                                ;*WMC2 MICRO ERROR 03
                                ;*WMC2-WMC FORMAT/SKIP-ACCEPTANCE/REJECTION
                                ;*M8959
                                ;*WMC ERRORS BYTE NON ZERO AFTER A VALID READ COMMAND
                                ;*ACTUAL = NNNN
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT  = NN
                                ;*
                                ;*WMC2 MICRO ERROR 02
                                ;*WMC2 MICRO ERROR 04
                                ;*WMC2-WMC FORMAT/SKIP-ACCEPTANCE/REJECTION
                                ;*M8959
                                ;*WMC - ERRORS BYTE INCORRECT AFTER INVALID READ COMMAND
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT  = NN
                                ;*
                                ;*WMC2 MICRO ERROR 02
                                ;*WMC2 MICRO ERROR 05
                                ;*WMC2-WMC FORMAT/SKIP-ACCEPTANCE/REJECTION
                                ;*M8959
                                ;*WMC ECODE COUNTER LOW BYTE NOT CORRECT AFTER INVALID READ COMMAND
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*DATA FORMAT - MM
                                ;*SKIP COUNT  = NN
                                ;*
```

```
1564                                    ;*WMC2 MICRO ERROR 02
1565                                    ;*WMC2 MICRO ERROR 06
1566                                    ;*WMC2-WMC FORMAT/SKIP-ACCEPTANCE/REJECTION
1567                                    ;*M8959
1568                                    ;*WMC ECODE COUNTER HIGH BYTE NOT CORRECT AFTER INVALID READ COMMAND
1569                                    ;*ACTUAL = NNNN
1570                                    ;*EXPECTED = NNNN
1571                                    ;*DATA FORMAT = MM
1572                                    ;*SKIP COUNT  = NN
1573  4399                              $
 (1)                                    ; ************************************************************************
1574
1575  4399                              TEST2:  TESTX   @2
 (1)  4399  3E  02          7.0                         MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  439B  CD  03  28     18.0                         CALL    TSET            ;SETUP THE TEST
1576                                    ;%WMC2-WMC FORMAT/SKIP-ACCEPTANLE/REJECTION
1577                                    ;&M8959
1578
1579  439E  3E  00          7.0  TST2L:  MVI     A,@0            ;LOAD PDP11 NORMAL FORMAT CODE
1580  43A0  32  1D  44     13.0          STA     FMTCOD          ;STORE FORMAT CODE
1581  43A3  3E  02          7.0          MVI     A,@2            ;LOAD THE FIRST FAIL SKIP COUNT
1582  43A5  32  1C  44     13.0          STA     SKIPFF          ;STORE SKIP COUNT
1583  43A8  21  FE  FF     10.0          LXI     H,@-2           ;LOAD THE SKIP ERROR CODE
1584  43AB  22  1E  44     16.0          SHLD    ERRCOD          ;STORE ERROR CODE
1585  43AE  CD  20  44     18.0          CALL    SKIPSUB         ;
1586                                                            ;
1587  43B1  3E  01          7.0          MVI     A,@1            ;LOAD PDP15 NORMAL FORMAT CODE
1588  43B3  32  1D  44     13.0          STA     FMTCOD          ;STORE FORMAT CODE
1589                                    ;FIRST FAIL SKIP COUNT IS 2, AND IS STILL SET UP FROM ABOVE
1590                                    ;ERROR CODE IS -2, AND IS STILL SET UP FROM ABOVE
1591  43B6  CD  20  44     18.0          CALL    SKIPSUB         ;
1592                                                            ;
1593  43B9  3E  02          7.0          MVI     A,@2            ;LOAD PDP10 COMPAT. FORMAT CODE
1594  43BB  32  1D  44     13.0          STA     FMTCOD          ;STORE FORMAT CODE
1595  43BE  3E  04          7.0          MVI     A,@4            ;LOAD THE FIRST FAIL SKIP COUNT
1596  43C0  32  1C  44     13.0          STA     SKIPFF          ;STORE SKIP COUNT
1597  43C3  21  FF  FF     10.0          LXI     H,@-1           ;LOAD THE SKIP FAIL ERROR CODE
1598  43C6  22  1E  44     16.0          SHLD    ERRCOD          ;STORE THE ERROR CODE
1599  43C9  CD  20  44     18.0          CALL    SKIPSUB         ;
1600                                                            ;
1601  43CC  3E  03          7.0          MVI     A,@3            ;LOAD PDP10 DUMP FORMAT CODE
1602  43CE  32  1D  44     13.0          STA     FMTCOD          ;STORE THE FORMAT CODE
1603  43D1  3E  05          7.0          MVI     A,@5            ;FIRST FAIL OF 5
1604  43D3  32  1C  44     13.0          STA     SKIPFF          ;FOR 10 DUMP MODE - SKIP COUNT
1605
1606                                                    ;ERROR CODE IS -1, AND IS STILL SET UP FROM ABOVE
1607
1608  43D6  CD  20  44     18.0          CALL    SKIPSUB         ;
1609                                                            ;
1610  43D9  3E  04          7.0          MVI     A,@4            ;LOAD PDP10 HIGH DEN. FORMAT CODE
1611  43DB  32  1D  44     13.0          STA     FMTCOD          ;STORE THE FORMAT CODE
1612  43DE  3E  09          7.0          MVI     A,@11           ;LOAD THE FIRST FAIL SKIP COUNT
1613  43E0  32  1C  44     13.0          STA     SKIPFF          ;STORE THE SKIP COUNT
1614
```

```
                                                ;ERROR CODE IS -1, AND IS STILL SET UP FROM ABOVE

4419 CD 20 44   18.0            CALL    SKIPSUB         ;
                                                        ;
4419 3E 05       7.0            MVI     A,@5            ;LOAD IMAGE FORMAT CODE
4419 32 1D 44   13.0            STA     FMTCOD          ;STORE THE FORMAT CODE
4419 AF          4.0            XRA     A               ;INDICATE NO FAILING SKIP COUNTS
4419 32 1C 44   13.0            STA     SKIPFF          ;STORE THE SKIP COUNT
4419 CD 20 44   18.0            CALL    SKIPSUB
4419 3E 06       7.0            MVI     A,@6
4419 32 1D 44   13.0            STA     FMTCOD
4419 3E 09       7.0            MVI     A,@11
4419 32 1C 44   13.0            STA     SKIPFF
4419 21 FF FF   10.0            LXI     H,@-1
4419 22 1E 44   16.0            SHLD    ERRCOD
4419 CD 20 44   18.0            CALL    SKIPSUB

4419                            ENDTST  TST2L
                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
4419                            REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
4419 CD 06 28   18.0                            CALL    REQST
4419 00                                         .BYTE                   ;DATA PATTERN NUMBER
4419 00 00                                      .WORD                   ;SYSTEM "" COUNT
4419 00 00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
4419 00                                         .BYTE                   ;DATA COMPARE FLAG IF -1
4419 07                                         .BYTE   7               ;REQUEST CODE
4419 3A 9A 4F   13.0            LDA     ITERA           ;GET ITERATION COUNT
4419 3D          4.0            DCR     A               ;DOWNCOUNT
4419 32 9A 4F   13.0            STA     ITERA           ;SAVE COUNT
4419 F2 9E 43   10.0            JP      TST2L           ;DO TEST UNTIL TILL = 0
4419                    S
                                ; ***************************************************************************
4419 C3 18 28   10.0            JMP     TSTEND
441C                    S
                                ; ***************************************************************************
441C 00                 SKIPFF: .BYTE   0               ;FIRST FAIL SKIP COUNT
441D 00                 FMTCOD: .BYTE   0               ;DATA FORMAT MODE UNDER TEST
441E 00 00              ERRCOD: .WORD   0               ;ERROR CODE EXPECTED
```

```
                        .SBTTL  SKIPSUB - SUBROUTINE FOR TEST 2
4420                    SSUB
                        ;*********************************************************************
                        ;*SUBROUTINE TITLE
                        ;*---------- -----
                        ;*SKIP TEST SUBROUTINE
4420                    SD
                        ;*********************************************************************
                        ;*DESCRIPTION
                        ;*-----------
                        ;*THIS SUBROUTINE PERFORMS ALL THE INPUT/OUT AND ERROR REPORTING FOR
                        ;*TEST 2.  THE SUBROUTINE LOADS THE DATA FORMAT SPECIFIED WITH SKIP
                        ;*COUNTS FROM 0-17(8) AND CHECKS THE RESPONSE OF THE WRITE MICRO
                        ;*CONTROLLER TO THE DATA FORMAT/SKIP COUNT BYTE, ANY DEVIATION FROM
                        ;*THE EXPECTED RESULTS CAUSES AN ERROR MESSAGE TO BE OUTPUT.
                        ;*
                        ;*      INPUT:
                        ;*
                        ;*      FMTCOD  THIS DATA BYTE CONTAINS THE NUMBER OF THE DATA FORMAT
                        ;*              TO BE TESTED AS FOLLOWS:
                        ;*                      0 = 11 NORMAL
                        ;*                      1 = 15 NORMAL
                        ;*                      2 = 10 COMPATABILITY
                        ;*                      3 = 10 CORE DUMP
                        ;*                      4 = 10 HIGH DENSITY
                        ;*                      5 = IMAGE
                        ;*
                        ;*      SKIPFF  THIS DATA BYTE CONTAINS THE FIRST FAIL SKIP COUNT, WHICH
                        ;*              IS THE FIRST ILLEGAL SKIP COUNT FOR THE DATA FORMAT.
                        ;*
                        ;*      ERRCOD  THIS DATA WORD (16 BITS) CONTAINS THE NUMBER THAT SHOULD
                        ;*              BE PRESENT IN THE WRITE MICRO CONTROLLER ECODE COUNTER FOR
                        ;*              THE ILLEGAL SKIP COUNTS FOR THE DATA FORMAT.
                        ;*
                        ;*      OUTPUT:
                        ;*
                        ;*              THIS SUBROUTINE HAS NO OUTPUT TO RETURN TO THE CALLING
                        ;*              MODULE.  HOWEVER, AS A RESULT OF ERROR CONDITIONS DETECTED
                        ;*              WITHIN THIS MODULE, ERROR MESSAGES HANDSHAKES WITH THE
                        ;*              HOST CAN TAKE PLACE.
4420                    S
                        ;*********************************************************************
4420                    SP
                        ;*********************************************************************
                        ;*PROCEDURE
                        ;*---------
                        ;*BGNSUB
                        ;*  INIT ALL COUNTER CONTROL-BYTE, PAD, + ECODE
                        ;*  SET SKIP COUNT INDEX TO 0
                        ;*  BGNDO
                        ;*  :  RESTART WMC
                        ;*  :  LOAD WMC PACKING MODE NUMBER INTO DATACTL
                        ;*  :  CLEAR ALL COUNTERS-BYTE, PAD, + ECODE
                        ;*  :  CLEAR RESCHR WORD
```

```
                                        ;*  :  ENABLE WMC SO ROM PGM WILL START CHECKING PARAMETERS
                                        ;*  :  SMALL DELAY FOR WMC ROM PGM TO CHECK PARAMETERS
                                        ;*  :  CLEAR WMC ENABLE BIT SO WMC 'DONE' WILL SET
                                        ;*  :  IF FAIL ON SKIP COUNT EXPECTED
                                        ;*  :  :  THEN-CHECK FAIL COUNT EXPECTED
                                        ;*  :  :  IF CURRENT SKIP COUNT > OR = FAIL COUNT EXPECTED
                                        ;*  :  :  :  THEN-CHECK FOR EXPECTED ERROR CONDIDION
                                        ;*  :  :  :  IF ERROR AS EXPECTED
                                        ;*  :  :  :  :  THEN-CONTINUE
                                        ;*  :  :  :  :  ELSE-REPORT INCORRECT ERROR ON INVALID READ COMMAND
                                        ;*  :  :  :  ENDIF
                                        ;*  :  :  :  ELSE-CONTINUE
                                        ;*  :  :  ENDIF
                                        ;*  :  :  ELSE-CHECK FOR NO ERROR
                                        ;*  :  :  IF WMC DETECTED ERROR
                                        ;*  :  :  :  THEN-REPORT ERROR AFTER A VALID READ COMMAND
                                        ;*  :  :  :  ELSE-CONTINUE
                                        ;*  :  :  ENDIF
                                        ;*  :  ENDIF
                                        ;*  :  DO UNTIL SKIP COUNT > 17(8)
                                        ;*  ENDDO
                                        ;*ENDSUB
1706  4420                              S
 (1)                                    ; ***************************************************************************
1707
1708  4420                              SKIPSUB: INICNT                 ;INITIALIZE THE COUNTERS
 (1)  4420  3E  30          7.0                          MVI     A,@060          ;INIT THE
 (1)  4422  D3  D7         10.0                          OUT     CNTCTL          ;BYTE COUNTER
 (1)  4424  3E  70          7.0                          MVI     A,@160          ;INIT THE
 (1)  4426  D3  D7         10.0                          OUT     CNTCTL          ;PAD COUNTER
 (1)  4428  3E  B0          7.0                          MVI     A,@260          ;INIT THE
 (1)  442A  D3  D7         10.0                          OUT     CNTCTL          ;ERROR COUNTER
1709  442C  06  00          7.0                  MVI     B,0                     ;CLEAR THE SKIP COUNT OFFSET POINTER
1710  442E  3E  01          7.0          SKLP0:  MVI     A,W.RST                 ;LOAD THE WMC RESTART COMMAND BIT
1711  4430  D3  D3         10.0                  OUT     WMCCTL                  ;RESTART THE WMC
1712  4432  AF              4.0                  XRA     A                       ;CLEAR THE ACCUMULATOR
1713  4433  D3  D3         10.0                  OUT     WMCCTL
1714  4435  3A  1D  44      13.0                  LDA     FMTCOD                  ;GET THE FORMAT CODE
1715  4438  07              4.0                  RLC                             ;SHIFT THE ACC. LEFT
1716  4439  07              4.0                  RLC                             ;3 BIT POSITIONS
1717  443A  07              4.0                  RLC                             ;3 BIT POSITIONS
1718  443B  07              4.0                  RLC
1719  443C  E6  70          7.0                  ANI     $70                     ;REMOVE UNWANTED BITS
1720  443E  B0              4.0                  ORA     B                       ;OR IN SKIP COUNT
1721  443F  D3  D0         10.0                  OUT     DATACTL                 ;SET UP THE WRITE MICRO
1722  4441                                       ROUT    RO2L                    ;WITH FORMAT/SKIP INFO.
 (1)  4441  D3  84         10.0                          OUT     RO2L                    ;WRITE AC INTO RO2L
 (1)  4443  7F              4.0                          MOV     A,A                     ;RETRY LINK
1723  4444                                       CLRECT                          ;CLEAR THE ECODE COUNT
 (1)  4444  AF              4.0                          XRA     A                       ;CLEAR THE ACCUMULATOR
 (1)  4445  D3  D6         10.0                          OUT     ERRCNT                  ;CLEAR BITS 7-0
 (1)  4447  D3  D6         10.0                          OUT     ERRCNT                  ;CLEAR BITS 15-8
1724  4449                                       CLRBCT                          ;CLEAR THE BYTE COUNT
 (1)  4449  AF              4.0                          XRA     A                       ;CLEAR THE ACCUMULATOR
```

```
 (1)  444A  D3  D4          10.0                    OUT     BYTCNT          ;CLEAR BITS 7-0
 (1)  444C  D3  D4          10.0                    OUT     BYTCNT          ;CLEAR BITS 15-8
1725  444E                                CLRPCT              ;CLEAR THE PAD COUNT
 (1)  444E  AF               4.0                     XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  444F  D3  D5          10.0                    OUT     PADCNT          ;CLEAR BITS 7-0
 (1)  4451  D3  D5          10.0                    OUT     PADCNT          ;CLEAR BITS 15-8
1726  4453  AF               4.0            XRA     A                       ;CLEAR THE ACCUMULATOR
1727  4454  D3  D1          10.0            OUT     RESCHR                  ;CLEAR THE RESIDUAL CHARACTER
1728                                                                        ;
1729  4456  3E  80           7.0            MVI     A,W.ENAB                ;
1730  4458  D3  D3          10.0            OUT     WMCCTL                  ;ENABLE WMC - READ PE
1731                                                                        ;
1732  445A  3E  1E           7.0            MVI     A,30                    ;LOAD THE TIMEOUT COUNT
1733  445C  3D               4.0    T2W:    DCR     A                       ;DECREMENT THE TIMEOUT COUNT
1734  445D  C2  5C  44       10.0            JNZ     T2W                     ;CONTINUE UNTIL TIMEOUT
1735  4460  AF               4.0            XRA     A
1736  4461  D3  D3          10.0            OUT     WMCCTL                  ;NOW DISABLE WMC SO 'DONE' WILL SET
1737  4463  3A  1C  44       13.0            LDA     SKIPFF                  ;GET THE FIRST FAIL SKIP FLAG
1738  4466  A7               4.0            ANA     A                       ;SET CONDITION CODE
1739  4467  CA  71  44       10.0            JZ      SKIP0                   ;NO ERROR EXPECTED-GO CHECK
1740  446A  B8               4.0            CMP     B                       ;COMPARE SKIP COUNT & FF SKIP COUNT
1741  446B  DA  89  44       10.0            JC      SKIP1                   ;SKIP COUNT > FF SKIP COUNT
1742  446E  CA  89  44       10.0            JZ      SKIP1                   ;SKIP COUNT IS EQUAL TO THE FIRST FAIL SKIP COUNT
1743  4471                            SKIP0:
1744  4471  DB  DA          10.0            IN      WMCERR                  ;GET THE WMC ERRORS BYTE
1745  4473  E6  70           7.0            ANI     $70                     ;TEST FOR ERROR
1746  4475  CA  80  44       10.0            JZ      SKCN3                   ;NONE-CONTINUE
1747  4478                                  ROUT    ADATA                   ;STORE THE ACTUAL WMC ERROR BYTE
 (1)  4478  D3  94          10.0                    OUT     ADATA                   ;WRITE AC INTO ADATA
 (1)  447A  7F               4.0                    MOV     A,A                     ;RETRY LINK
1748  447B                                ERRA    SKLP0,SKCN3,@2
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  447B  CD  0F  28      18.0                            CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0003                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  447E  03                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  447F  02                                              .BYTE   @2              ;PRINT ROUTINE NUMBER
 (1)  4480  CD  15  28      18.0                    SKCN3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4483  DA  2E  44      10.0                            JC      SKLP0           ;LOOP ADDRESS IF LOOP SPECIFIED
1749                            ;>WMC ERRORS BYTE NON ZERO AFTER A VALID READ COMMAND
1750  4486  C3  D8  44       10.0            JMP     SKINC
1751  4489                            SKIP1:
1752  4489  DB  DA          10.0            IN      WMCERR                  ;INPUT THE WMC ERRORS BYTE
1753  448B  E6  70           7.0            ANI     $70                     ;REMOVE UNWANTED BITS
1754  448D  FE  30           7.0            CPI     W.ROME!W.ERR            ;WMC ROM PARITY ERROR AND ERROR?
1755  448F  CA  9F  44       10.0            JZ      SKCN7                   ;YES-CONTINUE
1756  4492                                  ROUT    ADATA                   ;STORE ACTUAL ERROR BYTE
 (1)  4492  D3  94          10.0                    OUT     ADATA                   ;WRITE AC INTO ADATA
 (1)  4494  7F               4.0                    MOV     A,A                     ;RETRY LINK
1757  4495  3E  30           7.0            MVI     A,W.ROME!W.ERR          ;LOAD EXPECTED ERROR BYTE
1758  4497                                  ROUT    EDATA                   ;STORE IN CAS
 (1)  4497  D3  95          10.0                    OUT     EDATA                   ;WRITE AC INTO EDATA
 (1)  4499  7F               4.0                    MOV     A,A                     ;RETRY LINK
```

```
1760  449A                                        ERRB    SKLP0,SKCN7,@2
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  449A  CD  12  28        18.0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0004                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  449D  04                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  449E  02                                         .BYTE   @2              ;PRINT ROUTINE NUMBER
 (1)  449F  CD  15  28        18.0               SKCN7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44A2  DA  2E  44        10.0                       JC      SKLP0           ;LOOP ADDRESS IF LOOP SPECIFIED
1761                                        ;>WMC - ERRORS BYTE INCORRECT AFTER INVALID READ COMMAND
1762  44A5  2A  1E  44        16.0                LHLD    ERRCOD          ;EXPECTED ECODE COUNT=ACTUAL ECODE COUNT
1763  44A8  7D                  4.0                MOV     A,L             ;COPY L TO A
1764  44A9                                         ROUT    EDATA           ;STORE EXPECTED ECODE COUNT
 (1)  44A9  D3  95            10.0                       OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44AB  7F                 4.0                       MOV     A,A             ;RETRY LINK
1765  44AC  DB  D6            10.0                IN      ERRCNT          ;INPUT ECODE COUNT BITS 7:0
1766  44AE                                         ROUT    ADATA           ;STORE ACTUAL ECODE COUNT
 (1)  44AE  D3  94            10.0                       OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44B0  7F                 4.0                       MOV     A,A             ;RETRY LINK
1767  44B1  BD                  4.0                CMP     L               ;COMPARE ACTUAL AND EXPECTED
1768  44B2  CA  BA  44        10.0                 JZ      SKCN5           ;CONTINUE-IF EQUAL
1769  44B5                                         ERRB    SKLP0,SKCN5,@2
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44B5  CD  12  28        18.0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0005                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44B8  05                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44B9  02                                         .BYTE   @2              ;PRINT ROUTINE NUMBER
 (1)  44BA  CD  15  28        18.0               SKCN5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44BD  DA  2E  44        10.0                       JC      SKLP0           ;LOOP ADDRESS IF LOOP SPECIFIED
1770                                        ;>WMC ECODE COUNTER LOW BYTE NOT CORRECT AFTER INVALID READ COMMAND
1771  44C0  7C                  4.0                MOV     A,H             ;COPY H TO A
1772  44C1                                         ROUT    EDATA           ;STORE EXPECTED ECODE COUNT
 (1)  44C1  D3  95            10.0                       OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44C3  7F                 4.0                       MOV     A,A             ;RETRY LINK
1773  44C4  DB  D6            10.0                IN      ERRCNT          ;INPUT ECODE COUNT BITS 15:8
1774  44C6                                         ROUT    ADATA           ;STORE ACTUAL ECODE COUNT
 (1)  44C6  D3  94            10.0                       OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44C8  7F                 4.0                       MOV     A,A             ;RETRY LINK
1775  44C9  BC                  4.0                CMP     H               ;COMPARE ACTUAL WITH EXPECTED
1776  44CA  CA  D2  44        10.0                 JZ      SKCN6           ;CONTINUE IF EQUAL
1777  44CD                                         ERRB    SKLP0,SKCN6,@2
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44CD  CD  12  28        18.0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0006                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44D0  06                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44D1  02                                         .BYTE   @2              ;PRINT ROUTINE NUMBER
 (1)  44D2  CD  15  28        18.0               SKCN6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44D5  DA  2E  44        10.0                       JC      SKLP0           ;LOOP ADDRESS IF LOOP SPECIFIED
1778                                        ;>WMC ECODE COUNTER HIGH BYTE NOT CORRECT AFTER INVALID READ COMMAND
```

```
1780                                    ;HERE TO UPDATE THE SKIP COUNT FOR THE NEXT TEST
1781
1782  44D8  04                    4.0  SKINC:  INR     B               ;UPDATE THE SKIP COUNT
1783  44D9  78                    4.0          MOV     A,B             ;
1784  44DA  FE  10                7.0          CPI     @20             ;DONE?
1785  44DC  C2  2E  44           10.0          JNZ     SKLP0           ;NO - CONTINUE
1786  44DF  C9                   10.0          RET                     ;YES - RETURN
1787
1788        0000                            .END
```

```
A      =%0007   ADATA  = 0094   AMTIEP= 0001   AMTIE7= 0002
ARAIDF= 0098    ASAVE   4F9B    ATTCD   4F97    AXNUM   4F91
B      =%0000   BADST  = 0090   BIT0   = 0001   BIT1   = 0002
BIT15  = 8000   BIT2   = 0004   BIT3   = 0008   BIT4   = 0010
BIT5   = 0020   BIT6   = 0040   BIT7   = 0080   BIT8   = 0100
BIT9   = 0200   BRKPBC= 4F0A    BRKRAM= 4F10    BRKSTR= 4E60
BRKXCT= 4F00    BSAVE   4F9C    BYTCNT= 00D4    BYTEH   4F24
BYTEL   4F23    C      =%0001   CASCT   4F21    CASCTL= 00A0
CASSTA= 00A0    CATTH  = 0089   CATTL  = 0088   CBUSST= 00A1
CBYTH  = 008B   CBYTL  = 008A   CDG1H  = 0087   CDG1L  = 0086
CDG2H  = 0093   CDG2L  = 0092   CDG3H  = 0095   CDG3L  = 0094
CDVIH  = 008D   CDVTL  = 008C   CHPTIE= 0028   CHOTIE= 0020
CH1TIE= 0021    CH2TIE= 0022    CH3TIE= 0023   CH4TIE= 0024
CH5TIE= 0025    CH6TIE= 0026    CH7TIE= 0027   CKLOP  = 2815
CLKCTL= 00F0    CLOCK   4F26    CMCOH  = 0099   CMCOL  = 0098
CMC1H  = 009B   CMC1L  = 009A   CMC2H  = 009D   CMC2L  = 009C
CMC3H  = 009F   CMC3L  = 009E   CMINH  = 0097   CMINL  = 0096
CNTCTL= 00D7    CRCWRD= 0018    CSAVE   4F9D    CSRLH  = 0091
CSRLL  = 0090   CTCH   = 0085   CTCL   = 0084   CTSTH  = 008F
CTSTL  = 008E   CXCTH  = 0081   CXCTL  = 0080   CXINH  = 0083
CXINL  = 0082   C.     = 0001   C.AVAI= 0080   C.DP   = 0008
C.DSE  = 0010   C.DTU  = 0003   C.DVA  = 0008   C.FAIL= 00FC
C.FMT  = 0070   C.FNCT= 003E    C.GO   = 0001   C.INTC= 00FE
C.MAIN= 0020    C.NSA  = 0080   C.RCT  = 00FC   C.SER  = 0080
C.SHR  = 0040   C.SKPC= 000F    C.TAPE= 0040   C.WCS  = 0002
D      =%0002   DATACT= 00D0    DBUS    4F28    DBUSCT= 00C0
DBUSST= 00C0    DDRA   = 00D8   DDRAIN= 0010   DDRB   = 00D9
DDRBIN= 0002    DDRC   = 00DA   DDRCIN= 0001   DDRCO  = 0088
DDRCTL= 00DB    DIAFLG  4F22    DIAGPG= 4300   DIAGRM= 4F90
DIARD  = 000B   DONE1  = 0045   DONINT= 0010   DSAVE   4F9E
DSE    = 0006   D.ATH0= 0001   D.ATH1= 0002   D.EOTD= 0010
D.LAGC= 0020    D.NOTW= 0040    D.NTHR= 0004   D.TACH= 0008
D.WR4  = 0080   E      =%0003   ECCBAD= 0042   ECCCOR= 0019
ECCOK  = 0041   ECCSTA= 001A    ECCTST= 000E   EDATA  = 0095
EOTCLR= 0003    ERFLG   4F93    ERLP   = 2809   ERLPA  = 280E
ERLPB  = 2812   ERLPE  = 280C   ERNUM   4F90   ERRCNT= 00D6
ERRCOD  441E    ESAVE   4F9F    E.ACRC= 0010   E.AMT  = 0020
E.CDP  = 0080   E.CRC  = 0080   E.PNTR= 0008   E.RPE  = 0040
E.STEC= 0001    E.TTEC= 0002    E.UNC  = 0004   FIFORD= 006A
FMTCOD  441D    FORMAT  4F25    FWDTST= 0061   GCRID  = 0089
GCRSET= 0002    GOODTM= 0092    H      =%0004   HLSAVE  4FA0
IE     = 0008   INTSTA= 00E0   ITERA   4F9A   I.PWR  = 0020
I.RMPE= 0040    I5.5   = 0010   I6.5   = 0020   I7.5   = 0040
KCALL  = 005F   KCLR   = 007B   KDEP   = 003F   KENAB  = 0078
KEXAM  = 003E   KEYBRD= 00C8    KEY1   = 0078   KEY10  = 006D
KEY11  = 006E   KEY12  = 006F   KEY13  = 005C   KEY14  = 005D
KEY15  = 005E   KEY16  = 005F   KEY17  = 003C   KEY18  = 003D
KEY19  = 003E   KEY2   = 0079   KEY20  = 003F   KEY3   = 007A
KEY4   = 007B   KEY5   = 0074   KEY6   = 0075   KEY7   = 0076
KEY8   = 0077   KEY9   = 006C   KINTA  = 006F   KLDAD  = 003D
KN0    = 003C   KN1    = 005C   KN2    = 005D   KN3    = 005E
KN4    = 006C   KN5    = 006D   KN6    = 006E   KN7    = 0074
KN8    = 0075   KN9    = 0076   KU2    = 0079   KU3    = 007A
KU8    = 0077   L      =%0005   LBLANK= 000F   LCE    = 000B
```

```
LCH    = 000C          LCL    = 000D          LCP    = 000E          LC0    = 0000
LC1    = 0001          LC2    = 0002          LC3    = 0003          LC4    = 0004
LC5    = 0005          LC6    = 0006          LC7    = 0007          LC8    = 0008
LC9    = 0009          LDLEDA= 00CA           LDLEDB= 00CB           LDLEDC= 00CC
LDLEDD= 00CD           LDLEDE= 00CE           LDLEDF= 00CF           LKDIAG= 2800
LKKBD  = 004C          LKKEY  = 0049          LKLWMG= 0058           LKLWMP= 0055
LKLWPG= 0052           LKLWPP= 004F           LKMOD7= 0046           LKOPR  = 0046
LPFLG   4F94           LPNUM   4F92           M      =%0006          MBSEL  = 00E0
MB.A   = 0008          MB.B   = 0004          MEMTOP= 4FFF           MINUS  = 000A
MM     = 8000          MSE    = 0008          MSGN   = 0006          MTACLR= 0000
MT.ARA= 0020           MT.CPE= 0080           MT.DSE= 0001           MT.FWD= 0040
MT.INH= 0008           MT.LWR= 0004           MT.MOT= 0002           MT.NWT= 0080
MT.PEC= 0040           MT.PSB= 0004           MT.PS0= 0001           MT.PS1= 0002
MT.REV= 0020           MT.WRT= 0010           MT.Z   = 0008          M.ATA  = 0080
M.CAPE= 0020           M.CONT= 0080           M.DEM  = 0020          M.EBL  = 0010
M.EXC  = 0008          M.FAIL= 0008           M.ILR  = 0010          M.INIT= 0010
M.OCC  = 0020          M.ONLI= 0001           M.PE   = 0040          M.PORT= 0080
M.RDEN= 0002           M.RDPE= 0008           M.RUN  = 0004          M.SCLK= 0001
M.TRA  = 0040          M.UNIT= 0007           M.WCLK= 0040           M.WCLN= 0080
M.WREN= 0080           M5.5   = 0001          M6.5   = 0002          M7.5   = 0004
NOTCAP= 0088           OKAY   = 00FF          OPRRAM= 4300           OPSTRT= 0058
OPVER  = 0040          PADCNT= 00D5           PADCRC= 0080           PDIAG  = 0048
PEID   = 008A          PENAB  = 004C          PESET  = 0001          PL     = 00B1
PRDD   = 004C          PRENF  = 009C          PS     = 00B2          PSTAT  = 0048
PSW    =%0009          P.AMTP= 0001           P.BCTC= 0040           P.CMDP= 0020
P.INTE= 0080           P.LCS  = 0040          P.LWR  = 0020          P.RDP  = 0002
P.RPEN= 0004           P.RPST= 0002           P.RP0E= 0020           P.RP1E= 0010
P.RP2E= 0020           P.RP3E= 0010           P.SING= 0080           P.STAT= 0002
P.STPE= 0080           P.TACH= 0008           P.TUPR= 0010           P.WCSP= 0004
P.WDS  = 0040          P.WFLP= 0001           P.WPEN= 0010           P.WP0E= 0008
P.WP1E= 0004           P.WP2E= 0008           P.WP3E= 0004           P.5VOK= 0002
QUE    = 281B          QUEM   = 281E          RAMT   = 0010          RARA   = 0006
RARA1  = 0004          RCHBD0= 0048           RCHBD1= 0047           RCHOK  = 0046
RCHTST= 000C           RCLRT  = 000D          RCMD   = 000B          RCMLP  = 0003
RCONT  = 0080          RDATA  = 0017          RDCLK  = 0010          RDON   = 0011
READG  = 0007          REND   = 0014          REQST  = 2806          RESCHR= 00D1
REVTST= 0064           REWIND= 0004           RFIFOL= 0008           RGCLK  = 0002
RGCRI  = 0003          RIBG   = 0001          RILL   = 0012          RINST  = 000C
RMCTST= 0008           RMK2   = 0013          RNOP   = 0000          RPATH  = 0001
RPBAD  = 0044          RPCHI  = 0001          RPCLK  = 0003          RPCTL  = 0009
RPEI   = 0002          RPFAIL= 0000           RPF1   = 009D          RPF2   = 009E
RPOK   = 0043          RPOSTN= 0016           RPSTA  = 0015          RRCMT  = 000A
RSTAT  = 0002          RTIEB  = 000A          RTIER  = 0030          RTM    = 0005
RUNKI  = 0009          RUPTST= 005E           RWDUNL= 0005           R.AMT  = 0001
R.BOP  = 0008          R.DATA= 0040           R.DON  = 0002          R.DRDY= 0010
R.END  = 0010          R.ILL  = 0004          R.JVOK= 0004           R.MK2  = 0008
R.PLOD= 0008           R.PLO0= 0010           R.PLO1= 0020           R.POST= 0020
R.STNM= 0002           R.STOP= 0004           R.STPC= 0001           R.TBJN= 0080
R.TSTD= 0040           R.VOK  = 0080          R00H   = 0081          R00L   = 0080
R01H   = 0083          R01L   = 0082          R02H   = 0085          R02L   = 0084
R03H   = 0087          R03L   = 0086          R04H   = 0089          R04L   = 0088
R05H   = 008B          R05L   = 008A          R06H   = 008D          R06L   = 008C
R07H   = 008F          R07L   = 008E          R10H   = 0091          R10L   = 0090
R11H   = 0093          R11L   = 0092          R12H   = 0095          R12L   = 0094
```

```
R13H  = 0097          R13L  = 0096          R14H  = 0099          R14L  = 0098
R15H  = 009B          R15L  = 009A          R16H  = 009D          R16L  = 009C
R17H  = 009F          R17L  = 009E          R7.5  = 0010          SELCLR= 0000
SETATA= 00A1          SID   = 0080          SKCN3   4480  G       SKCN5   44BA  G
SKCN6   44D2  G       SKCN7   449F  G       SKINC   44D8          SKIPFF  441C
SKIPSU  4420          SKIP0   4471          SKIP1   4489          SKLP0   442E
SOD   = 0080          SOE   = 0040          SP    =%0008          SSCLK = 0040
SSTEP = 0005          STACK = 4FFF          STATRM= 4F20          STPCT   4F20
STRSP = 5000          TADR00= 0080          TADR01= 0081          TADR02= 0082
TADR03= 0083          TADR04= 0084          TADR05= 0085          TADR06= 0086
TADR07= 0087          TADR10= 0088          TADR11= 0089          TADR12= 008A
TADR13= 008B          TAMT  = 0044          TASEL = 0080          TCMD  = 0040
TC.FWD= 0040          TC.INH= 0008          TC.LWR= 0004          TC.REV= 0020
TC.WRT= 0010          TEMP    4F99          TEST1   4300          TEST2   4399
TMF   = 0099          TMRDY = 0040          TRKENA= 00D2          TSET  = 2803
TSTEND= 2818          TSTS  = 0040          TST1L   4305          TST2L   439E
TUSEL0= 00D1          TUSEL1= 00D2          TU78  = 0010          T.ATH0= 0001
T.ATH1= 0002          T.BOT = 0004          T.EOT = 0002          T.FPT = 0001
T.NTHR= 0004          T.ONL = 0020          T.PES = 0008          T.PSBJ= 0020
T.PSOJ= 0008          T.PS1J= 0010          T.RDY = 0080          T.RDYO= 0040
T.RWD = 0010          T.SCLK= 0002          T1CN8   4371  G       T1CO2   435C  G
T1LP0   431A          T1LP1   431C          T1W     434C          T2W     445C
UIBG  = 00A1          VALFC   4F98          VALTB   4F95          VELTST= 005B
WDR.P = 0010          WMCCTL= 00D3          WMCERR= 00DA          WMCSTA= 00D0
WRTCLK= 0000          WRTDAT= 00D3          W.ACRC= 0004          W.CRC = 0008
W.DIAG= 0002          W.DONN= 0040          W.ECC = 0010          W.ENAB= 0080
W.ERR = 0020          W.FMT = 0070          W.GCR = 0010          W.LEFT= 0004
W.ONES= 0020          W.RESI= 0002          W.REV = 0004          W.ROME= 0010
W.RST = 0001          W.SKIP= 000F          W.WRIT= 0008          W.XFER= 0020
X     =%000A          X.DONN= 0080          X.ENAB= 0040          X.PEPE= 0002
X.ROME= 0001          X.WCLK= 0001          Y     =%000B          .     = 44E0

ERRORS DETECTED:  0

*WMC2.A78/PTP,WMC2=NLIST,PARAM,MACRO,LIST,WMC2
RUN-TIME: 3 4 0 SECONDS
CORE USED:  10K
```

3 'MEMORY MAP'' EXTENDED ADDRESS BIT
19 READ MICRO CONTROLLER CONTROL & STATUS WORDS
139 TRANSPORT/PORT COMMAND + STATUS WORDS
380 COMMON ADDRESS SPACE REGISTERS
444 MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485 MASSBUS CONTROL WORD DEFINITIONS
557 WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709 KEYBOARD/DISPLAY CONTROL + STATUS WORD
797 MISCELLANEOUS VARIABLE DEFINITIONS
801 RAM ADDRESS DEFINITIONS
819 LINKAGE TABLE ADDRESSES
842 BIT DEFINITIONS
856 RIM AND SIM INSTRUCTION BIT DEFINITIONS
877 INTERRUPTABLE INSTRUCTION MACROS
924 DIAGNOSTIC VARIABLES
945 COMMON VARIABLE STORAGE
959 MACRO CALL DEFINITIONS
1047 TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101 SU80 MACRO CALLS - ERROR AND TEST INITIALIZATION
1333 TEST 1 - WRITE MICRO BYTE ASSY. TESTS FOR ALL FORMAT CODES

```
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
```

```
;*****************************************************************************
;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA', OR "ERRB"
;ERROR MACRO CALLS
;
;       1  -    REQUEST HOST CPU TO PRINT:
;               "BYTE/SCLK COUNT NUMBER = LLL"
;               WHERE:
;               LLL = THE VALUE STORED IN CAS REGISTER 5
;                     (THE BYTE COUNT REGISTER 16 BITS).
;
;       2  -    REQUEST HOST CPU TO PRINT:
;               DATA FORMAT = MM
;               SKIP COUNT  = NN
;               WHERE:
;               MM = DATA FORMAT FROM CAS REGISTER 2
;               NN = SKIP COUNT FROM CAS REGISTER 2
;
;       3  -    REQUEST HOST CPU TO PRINT:
;               BYTE-SCLK COUNT = LLL
;               DATA FORMAT = MM
;               SKIP COUNT  = NN
;               WHERE:
;               LLL = AS ABOVE
;               MM  = AS ABOVE
;               NN  = AS ABOVE
;
;       4  -    REQUEST HOST TO PRINT:
;               TRANSITION COUNT = LLL
;               WHERE: LLL = COUNT FROM CAS REGISTER 05
;
;       5  -    REQUEST HOST CPU TO PRINT:
;               EXPECTED 18 BITS =      E EEEEEE
;               ACTUAL 18 BITS =        A AAAAAA
;
;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
;               OF CAS REG 15 LOW BYTE.
;
;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
;               17 LOW BYTE SIGN BIT.
;
;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
;               AND/OR ACTUAL DATA.
;
;       7  -    REQUEST HOST CPU TO PRINT:
;               "SUBGROUP NUMBER = LLL"
;               WHERE:
;               LLL = THE VALUE STORED IN CAS REGISTER 5
;                     (THE BYTE COUNT REGISTER 16 BITS).
;
;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
;*****************************************************************************
```

```
1016                    ;*****************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ;*****************************************************************
1035                    ;*****************************************************************
1036                    ;DATA PATTERN CODES
1037
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ;*****************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  wMC3 - BYTE ASSEMBLY TESTS
                                .SBTTL  TEST 1 - WRITE MICRO BYTE ASSY. TESTS FOR ALL FORMAT CODES
                                ;ID     WMC3-WRITE MICRO CONTROLLER PART #3

1336  4300                      ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1337                            ;*BYTE ASSBMELY TESTS
1338  4300                      SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1339                            ;*THIS SERIES OF TESTS IS DESIGNED TO TEST THE WRITE MICROCONTROLLER BYTE
1340                            ;*ASSEMBLY LOGIC.  A KNOWN DATA PATTERN IS FED INTO BYTE ASSEMBLY LOGIC
1341                            ;*FROM THE DIAGNOSTIC DATA REGISTERS - "DDR" - TO PROVIDE AN 18 BIT WIDE
1342                            ;*REPLACEMENT FOR MASSBUS DATA.
1343                            ;*
1344                            ;*DEPENDING ON THE FORMAT CODE SELECTED, THE BYTE ASSEMBLY LOGIC WILL
1345                            ;*"BREAK UP" THE INPUT DATA (18 BITS) INTO THE DESIRED BYTE FORMAT TO BE
1346                            ;*WRITTEN ON TAPE.
1347                            ;*
1348                            ;*THIS TEST WILL ISSUE 4 "SCLKS" OF 18 BIT INPUT DATA (FOR EACH FORMAT
1349                            ;*SELECTED AND EACH DATA TEST PATTERN SELECTED) FOR 72 BITS OF INPUT DATA
1350                            ;*FROM "DDR" REGISTERS.  THE SELECTED FORMAT CODES WILL GENERATE THE FOLLOWING
1351                            ;*AMOUNT OF BAL OUTPUT BYTES FOR EACH 4 SCLKS GENERATED.  TOTAL AMOUNT OF
1352                            ;*SCLKS/FORMAT=60.
1353                            ;*
1354                            ;*FORMAT CODE   ORDER OF BYTES      AMOUNT OF BYTES/4 SCLK CYCLE
1355                            ;*0-11 NORMAL       B-A                          8
1356                            ;*1-15 NORMAL       A-B                          8
1357                            ;*2-10 COMPATABILITY A-B-C-D                     8
1358                            ;*3-10 DUMP         A-B-C-D-E                   10
1359                            ;*4-10 HIGH DEN COMP A-B-C-D-E-F-G-H-I           9
1360                            ;*5-IMAGE           A                            4
1361                            ;*4-10 HIGH DEN DUMP A-B-C-D-E-F-G-H-I           9
1362  4300                      S
 (1)                            ; ***************************************************************************
1363  4300                      SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1364                            ;*BGNTST
1365                            ;*  INIT THE TEST
1366                            ;*  CLEAR FORMAT INDEX NUMBER
1367                            ;*  BGNDO
1368                            ;*  :  INIT DATA PATTERN NUMBER FOR PASS
1369                            ;*  :  RESTART WMC & SET CLOCK CONTROL TO SINGLE STEP
1370                            ;*  :  INIT WMC BYTE COUNTER + ERROR CODE COUNTER CONTROL
1371                            ;*  :  CLEAR COUNTERS
1372                            ;*  :  CLEAR PROGRAM FLAGS AND STORAGE AREAS
1373                            ;*  :  SET DDR CONTROL TO 'OUT' SO DATA INJECTION POSSIBLE
1374                            ;*  :  GENERATE DATA TO DDR REGISTERS AND CHECK STORAGE
1375                            ;*  :  INIT SCLK COUNTER
```

```
1376                    ;*  :  LOAD DATACTL WITH CURRENT FORMAT NUMBER-ALSO SAVE COPY IN REG 02
1377                    ;*  :  CLOCK WMC THROUGH ITS RESTART AREA-WMC ROM PROGRAM
1378                    ;*  :  LOAD BYTE COUNTER
1379                    ;*  :  SET WMCCTL TO 'WRITE' & 'ENABLE'-START WMC ROM PGM
1380                    ;*  :  BGNDO
1381                    ;*  :  :  CLOCK SYSTEM
1382                    ;*  :  :  IF WMCSTA HAS NO 'XFER' BIT SET
1383                    ;*  :  :  :  THEN-UPDATE TIMEOUT COUNTERS
1384                    ;*  :  :  :  :  IF TIMEOUT DETECTED
1385                    ;*  :  :  :  :  :  THEN-REPORT LACK OF ACTIVITY IN SPECIFIED TIME
1386                    ;*  :  :  :  :  :  ELSE-CONTINUE
1387                    ;*  :  :  :  :  ENDIF
1388                    ;*  :  :  :  ELSE-GET WMCSTA WORD + SAVE FOR LATER CHECKING
1389                    ;*  :  :  :  :  BGNDO
1390                    ;*  :  :  :  :  :  CLOCK SYSTEM TO FINISH 1ST NIBBLE OF DATA
1391                    ;*  :  :  :  :  :  IF > 20 CLOCKS
1392                    ;*  :  :  :  :  :  :  THEN-REPORT FAILURE TO CLEAR 'XFER' BIT
1393                    ;*  :  :  :  :  :  :  ELSE-CONTINUE
1394                    ;*  :  :  :  :  :  ENDIF
1395                    ;*  :  :  :  :  :  DO UNTIL 'XFER' CLEARS OR TIMEOUT
1396                    ;*  :  :  :  :  ENDDO
1397                    ;*  :  :  :  :  BGNDO
1398                    ;*  :  :  :  :  :  CLOCK SYSTEM TO BEGIN 2ND NIBBLE OF DATA
1399                    ;*  :  :  :  :  :  IF > 128 CLOCKS
1400                    ;*  :  :  :  :  :  :  THEN-REPORT FAILURE TO SET 'XFER' FOR 2ND NIBBLE
1401                    ;*  :  :  :  :  :  :  ELSE-CONTINUE
1402                    ;*  :  :  :  :  :  ENDIF
1403                    ;*  :  :  :  :  :  DO UNTIL 'XFER' SETS OR TIMEOUT
1404                    ;*  :  :  :  :  ENDDO
1405                    ;*  :  :  :  :  GET PARITY BIT FROM INTSTA WORD FOR ASSEMBLED BYTE
1406                    ;*  :  :  :  :  BGNDO
1407                    ;*  :  :  :  :  :  CLOCK SYSTEM TO FINISH 2ND NIBBLE
1408                    ;*  :  :  :  :  :  IF > 20 CLOCKS
1409                    ;*  :  :  :  :  :  :  THEN-REPORT FAILURE TO CLEAR 'XFER' ON 2ND NIBBLE
1410                    ;*  :  :  :  :  :  :  ELSE-CONTINUE
1411                    ;*  :  :  :  :  :  ENDIF
1412                    ;*  :  :  :  :  :  DO UNTIL 'XFER' IS CLEAR OR TIMEOUT
1413                    ;*  :  :  :  :  ENDDO
1414                    ;*  :  :  :  :  CLOCK SYSTEM TWICE MORE TO 'SET' INTERNAL STATUS
1415                    ;*  :  :  :  :  IF 8TH ASSEMBLED WRTDAT BYTE-ECC CHAR.
1416                    ;*  :  :  :  :  :  THEN-CLEAR CHARACTER COUNTER FOR ECC DETECTION
1417                    ;*  :  :  :  :  :  :  GET STORED WMCSTA FROM 1ST NIBBLE TIME
1418                    ;*  :  :  :  :  :  :  IF WMCSTA INDICATES 'ECC' BIT SET
1419                    ;*  :  :  :  :  :  :  :  THEN-EXIT 'DO' LOOP
1420                    ;*  :  :  :  :  :  :  :  ELSE-REPORT ASSEM. BYTE FROM WMC WAS INCORRECT
1421                    ;*  :  :  :  :  :  :  ENDIF
1422                    ;*  :  :  :  :  :  ELSE-COMPARE ACTUAL BYTE IN WRTDAT WITH EXPECTED BYTE
1423                    ;*  :  :  :  :  :  :  IF SAME
1424                    ;*  :  :  :  :  :  :  :  THEN-CONTINUE
1425                    ;*  :  :  :  :  :  :  :  ELSE-REPORT ASSEM. BYTE FROM WMC WAS INCORRECT
1426                    ;*  :  :  :  :  :  :  ENDIF
1427                    ;*  :  :  :  :  :  :  IF PARITY ON ASSEMBLED BYTE IS CORRECT
1428                    ;*  :  :  :  :  :  :  :  THEN-CONTINUE
1429                    ;*  :  :  :  :  :  :  :  ELSE-REPORT BYTE PARITY ERROR ON ASSEM. BYTE
```

```
1430                                    ;*  :  :  :  :  :  : ENDIF
1431                                    ;*  :  :  :  :  :  ENDIF
1432                                    ;*  :  :  :  :  ENDIF
1433                                    ;*  :  :  :  ENDIF
1434                                    ;*  :  :  ENDIF
1435                                    ;*  :  :  DO UNTIL ALL BYTES ASSEMBLED AND CHECKED FOR CURRENT PACKING MODE
1436                                    ;*  :  ENDDO
1437                                    ;*  :  CHECK BYTE COUNTER TO SEE IF=0
1438                                    ;*  :  IF BYTE COUNTER=0
1439                                    ;*  :  :  THEN-UPDATE FORMAT NUMBER FOR NEXT PASS
1440                                    ;*  :  :  ELSE-REPORT FAILURE OF BYTE COUNTER TO GO ZERO WHEN EXPECTED
1441                                    ;*  :  ENDIF
1442                                    ;*  :  DO UNTIL FORMAT NUMBER=7 (ILLEGAL #)
1443                                    ;*  ENDDO
1444                                    ;*ENDTST
1445  4300                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1446                                    ;*WMC3 MICRO TEST 01
1447                                    ;*WMC3 MICRO ERROR 01
1448                                    ;*WMC3-WMC BYTE ASSY TESTS
1449                                    ;*M8959
1450                                    ;*WMC ROM PROGRAM FAILED TO REQUEST NEXT 'SCLK' CYCLE IN 128 CLOCK CYCLES
1451                                    ;*BYTE-SCLK COUNT = LLL
1452                                    ;*DATA FORMAT = MM
1453                                    ;*SKIP COUNT  = NN
1454                                    ;*TEST ABORTED!
1455                                    ;*
1456                                    ;*WMC3 MICRO TEST 01
1457                                    ;*WMC3 MICRO ERROR 02
1458                                    ;*WMC3-WMC BYTE ASSY TESTS
1459                                    ;*M8959
1460                                    ;*WMC ROM PGM FAILED TO SET 'XFER' BIT IN 128 CLOCK CYCLES AFTER XFR STARTED
1461                                    ;*TEST ABORTED!
1462                                    ;*
1463                                    ;*WMC3 MICRO TEST 01
1464                                    ;*WMC3 MICRO ERROR 03
1465                                    ;*WMC3-WMC BYTE ASSY TESTS
1466                                    ;*M8959
1467                                    ;*WMC FAILED TO DROP 'SCLK' BIT AFTER SEVERAL SYSTEM CLOCKS
1468                                    ;*TEST ABORTED!
1469                                    ;*
1470                                    ;*WMC3 MICRO TEST 01
1471                                    ;*WMC3 MICRO ERROR 04
1472                                    ;*WMC3-WMC BYTE ASSY TESTS
1473                                    ;*M8959
1474                                    ;*WMC FAILED TO CLEAR 'XFER' BIT ON 1ST NIBBLE OF THIS BYTE ASSY
1475                                    ;*
1476                                    ;*WMC3 MICRO TEST 01
1477                                    ;*WMC3 MICRO ERROR 05
1478                                    ;*WMC3-WMC BYTE ASSY TESTS
1479                                    ;*M8959
1480                                    ;*WMC FAILED TO SET 'XFER' BIT FOR 2ND NIBBLE ON THIS BYTE ASSY
```

```
1481                    ;*BYTE-SCLK COUNT = LLL
1482                    ;*DATA FORMAT = MM
1483                    ;*SKIP COUNT  = NN
1484                    ;*TEST ABORTED!
1485                    ;*
1486                    ;*WMC3 MICRO TEST 01
1487                    ;*WMC3 MICRO ERROR 06
1488                    ;*WMC3-WMC BYTE ASSY TESTS
1489                    ;*M8959
1490                    ;*WMC FAILED TO CLEAR 'XFER' BIT FOR 2ND NIBBLE OF THIS BYTE ASSY
1491                    ;*BYTE-SCLK COUNT = LLL
1492                    ;*DATA FORMAT = MM
1493                    ;*SKIP COUNT  = NN
1494                    ;*TEST ABORTED!
1495                    ;*
1496                    ;*WMC3 MICRO TEST 01
1497                    ;*WMC3 MICRO ERROR 07
1498                    ;*WMC3-WMC BYTE ASSY TESTS
1499                    ;*M8959
1500                    ;*WMC BYTE ASSEMBLY FAULT - BYTE ASSEMBLED DOESN'T MATCH EXPECTED .ATA
1501                    ;*BYTE-SCLK COUNT = LLL
1502                    ;*DATA FORMAT = MM
1503                    ;*SKIP COUNT  = NN
1504                    ;*
1505                    ;*WMC3 MICRO TEST 01
1506                    ;*WMC3 MICRO ERROR 10
1507                    ;*WMC3-WMC BYTE ASSY TESTS
1508                    ;*M8959
1509                    ;*WMC PARITY BIT ON ASSEMBLED DATA SHOULD =0 (ODD PARITY) AND WAS =1
1510                    ;*
1511                    ;*WMC3 MICRO TEST 01
1512                    ;*WMC3 MICRO ERROR 11
1513                    ;*WMC3-WMC BYTE ASSY TESTS
1514                    ;*M8959
1515                    ;*WMC PARITY BIT ON ASSEMBLED DATA WAS -0 AND SHOULD BE -1
1516                    ;*
1517                    ;*WMC3 MICRO TEST 01
1518                    ;*WMC3 MICRO ERROR 12
1519                    ;*WMC3-WMC BYTE ASSY TESTS
1520                    ;*M8959
1521                    ;*WMC BYTE COUNTER EXPECTED TO BE =0 - ALL BYTES RECEIVED FROM WMC
1522                    ;*BYTE-SCLK COUNT = LLL
1523                    ;*DATA FORMAT = MM
1524                    ;*SKIP COUNT  = NN
1525                    ;*
1526                    ;*WMC3 MICRO TEST 01
1527                    ;*WMC3 MICRO ERROR 13
1528                    ;*WMC3-WMC BYTE ASSY TESTS
1529                    ;*M8959
1530                    ;*WMC FAILED TO SET 'ECC' IN WMC STATUS ON 8TH 'XFER' CYCLE
1531                    ;*ECC CHAR. NOT PRESENT ON WMC OUTPUT
1532                    ;*BYTE-SCLK COUNT = LLL
1533                    ;*DATA FORMAT = MM
1534                    ;*SKIP COUNT  = NN
```

```
                                        ;*
                                        ;*WMC3 MICRO TEST 01
                                        ;*WMC3 MICRO ERROR 14
                                        ;*WMC3-WMC BYTE ASSY TESTS
                                        ;*M8959
                                        ;*WMC FAILED TO SET 'WMC LEFT' BIT DURING 10 PACKING MODE
                                        ;*DATA FORMAT = MM
                                        ;*SKIP COUNT  = NN
4300                                    $
 (1)                                    ; ***************************************************************************
                                        TST1:   TESTX   @1
 (1) 4300 3E 01              7.0                        MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1) 4302 CD 03 28          18.0                        CALL    TSET            ;SETUP THE TEST
                                        ;%WMC3-WMC BYTE ASSY TESTS
                                        ;&M8959

                                        ;INIT FORMAT CODE COUNTER TO 0 - 1ST FORMAT CODE = 11 NORMAL

4305 AF                     4.0 TST1ST: XRA     A
4306 32 E4 45              13.0        STA     FORMT           ;SAVE FORMAT CODE
4309 32 E3 45              13.0        STA     EABCNT          ;CLEAR THE LOOP CONTROL
                                        ;INIT COUNTER CONTROL FOR BYTE COUNTER AND ECODE COUNTER

                                        ;RESTART WRITE MICROCONTROLLER ROM PROGRAM

430C 3E FF                  7.0 TST1L:  MVI     A,@377          ;-1 TO OFFSET NUMBER
430E 32 E5 45              13.0        STA     DPNUM           ;INIT DATA PATTERN OFFSET SO 1ST WILL BE "0"
4311 3E 01                  7.0        MVI     A,W.RST         ;GET 'RST' BIT
4313 D3 D3                 10.0        OUT     WMCCTL          ;RESTART WMC
4315 3E 05                  7.0        MVI     A,SSTEP         ;SET THE SYSTEM TO SINGLE STEP MODE
4317 32 DE 45              13.0        STA     CCTLWD          ;SAVE THE CONTROL WORD
431A D3 F0                 10.0        OUT     CLKCTL          ;WRITE THE CLOCK CONTROL WORD
431C CD 46 48              18.0        CALL    CLKSYS          ;CLOCK THE WRITE PATH
431F CD 46 48              18.0        CALL    CLKSYS          ;CLOCK THE WRITE PATH
4322 AF                     4.0        XRA     A               ;FINISH THE WMC RESTART
4323 D3 D3                 10.0        OUT     WMCCTL
4325 3E 30                  7.0        MVI     A,@60
4327 D3 D7                 10.0        OUT     CNTCTL          ;SELECT BYTE COUNTER
4329 3E B0                  7.0        MVI     A,@260
432B D3 D7                 10.0        OUT     CNTCTL          ;SELECT ECODE COUNTER

                                    ;CLEAR COUNTERS

432D                                    CLRBCT                  ;CLEAR BYTE COUNTER
 (1) 432D AF                 4.0                        XRA     A               ;CLEAR THE ACCUMULATOR
 (1) 432E D3 D4             10.0                        OUT     BYTCNT          ;CLEAR BITS 7-0
 (1) 4330 D3 D4             10.0                        OUT     BYTCNT          ;CLEAR BITS 15-8
4332                                    CLRECT                  ;CLEAR ECODE COUNTER
 (1) 4332 AF                 4.0                        XRA     A               ;CLEAR THE ACCUMULATOR
 (1) 4333 D3 D6             10.0                        OUT     ERRCNT          ;CLEAR BITS 7-0
 (1) 4335 D3 D6             10.0                        OUT     ERRCNT          ;CLEAR BITS 15-8

4337 32 E9 45              13.0        STA     ECCCNT          ;CLEAR 'ECC' CHECK COUNTER
```

```
1580  435A  32  EA  45   13.0          STA     CLKCNT          ;CLEAR THE 'SCLK' COUNTER
1581  435D  32  E1  45   13.0          STA     ABCNT           ;CLEAR THE ASSEM. BYTE COUNTER
1582  4340  32  E2  45   13.0          STA     ABCNTR
1583
1584                                ;SET DDR CONTROL TO "OUT" FOR DATA INJECTION INTO BAL LOGIC
1585
1586  4343  3E  88        7.0          MVI     A,@210          ;SET "OUT" BITS
1587  4345  D3  DB       10.0          OUT     DDRCTL          ;LOAD DDR CONTROL
1588
1589  4347  CD  F8  45   18.0          CALL    GENDAT          ;GENERATE THE 1ST SET OF DATA
1590  434A  CD  D3  46   18.0          CALL    GCDAT           ;AND THE 1ST SET OF BAL COMPARE BYTES
1591  434D  3E  04        7.0          MVI     A,4             ;GET THE SCLK COUNT FOR RELOAD
1592  434F  32  EB  45   13.0          STA     SCCNTR          ;SAVE IT
1593                              ;LOAD DATA CONTROL WITH SELECTED FORMAT CODE
1594
1595  4352  3A  E4  45   13.0          LDA     FORMT           ;GET THE FORMAT CODE
1596  4355  CD  03  47   18.0          CALL    SHL4            ;POSITION IT TO BITS 6-4
1597  4358  D3  D0       10.0          OUT     DATACTL         ;LOAD DATA CONTROL WORD
1598  435A                             ROUT    R02H            ;ALSO SAVE IN CAS REG 02
 (1)  435A  D3  85       10.0                  OUT     R02H            ;WRITE AC INTO R02H
 (1)  435C  7F            4.0                  MOV     A,A             ;RETRY LINK
1599  435D  3E  04        7.0          MVI     A,4             ;SET RECORD COUNT TO 1
1600  435F                             ROUT    R02L
 (1)  435F  D3  84       10.0                  OUT     R02L            ;WRITE AC INTO R02L
 (1)  4361  7F            4.0                  MOV     A,A             ;RETRY LINK
1601  4362  06  B4        7.0          MVI     B,180           ;CLOCK THE SYSTEM THRU THE RESTART AREA
1602  4364  CD  46  48   18.0  T1RST:  CALL    CLKSYS          ;CLOCK THE SYSTEM
1603  4367  05            4.0          DCR     B
1604  4368  C2  64  43   10.0          JNZ     T1RST           ;STAY HERE TILL DONE COUNTING
1605  436B  3A  DF  45   13.0          LDA     DCNTWD          ;GET COUNT OF BYTES/4 SCLK
1606  436E  CD  03  47   18.0          CALL    LDBCNT          ;LOAD THE BYTE COUNTER FOR THIS FMT MODE
1607
1608                              ;ENABLE WRITE MICROCONTROLLER - START OF DATA XFR - WRITE
1609
1610  4371  3E  88        7.0  TST1A:  MVI     A,W.ENAB!W.WRITE  ;SELECT 'WMC EN' + 'WRITE' (PE MODE)
1611  4373  D3  D3       10.0          OUT     WMCCTL          ;LOAD WRITE MICROCONTROLLER CONTROL
1612  4375  AF            4.0          XRA     A
1613  4376  32  E8  45   13.0          STA     CLKON           ;CLEAR 'SCLK' ON FLAG
1614
1615  4379  06  FF        7.0          MVI     B,@377          ;SET SINGLE STEP CLOCK 'TIMEOUT' LIMIT TO MAX POS. COUNT
1616  437B  48            4.0          MOV     C,B             ;COPY FOR THE 'XFER' TIMEOUT COUNTER
1617
1618  437C  CD  1A  44   18.0  TST1B:  CALL    CKSCLK          ;ISSUE SINGLE STEP CLOCK
1619  437F  DA  0C  43   10.0          JC      TST1L           ;JUMP BACK IF ERROR DETECTED
1620  4382  DB  D0       10.0          IN      WMCSTA          ;GET WMC STATUS WORD
1621  4384  E6  20        7.0          ANI     W.XFER          ;SAVE ONLY 'XFER' BIT
1622  4386  C2  2D  44   10.0          JNZ     XFRBIT          ;JUMP IF 'XFER' IS SET - DATA READY IN WRITE DATA WORD
1623  4389  3A  F6  45   13.0          LDA     SXFER           ;GET THE SOFTWARE XFER FLAG
1624  438C  A7            4.0          ANA     A
1625  438D  CA  9A  43   10.0          JZ      TST1C           ;JUMP IF XFER NOT SET
1626  4390  AF            4.0          XRA     A               ;CLEAR SOFTWARE FLAG
1627  4391  32  F6  45   13.0          STA     SXFER           ;FOR NEXT PASS
1628  4394  C5           12.0          PUSH    B
1629  4395  06  7F        7.0          MVI     B,@177          ;SETUP A WATCHDOG COUNT
```

```
1630  4397  C3  53  44   10.0          JMP     XFRB2           ;PROCESS THE 'XFER' BIT DETECTED
1631                              ;HERE TO SEE IF 'SCLK' TIMEOUT HAPPENED
1632
1633  439A  05                4.0  TST1C:  DCR     B               ;COUNT CLOCK CYCLE
1634  439B  C2  AC  43        10.0          JNZ     TST1D           ;TIMEOUT? - JUMP IF OK
1635
1636  439E                          ERR     TST1L,TST1BC,3
 (1)                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  439E  CD  09  28        18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0001                     MSGN    =       MSGN+1          ,UPDATE MESSAGE NUMBER FOR THIS
 (1)  43A1  01                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43A2  03                      .BYTE   3
 (1)  43A3  CD  15  28        18.0  TST1BC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43A6  DA  0C  43        10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1637                              ;>WMC ROM PROGRAM FAILED TO REQUEST NEXT 'SCLK' CYCLE IN 128 CLOCK CYCLES
1638                              ;<TEST ABORTED.
1639  43A9  C3  18  28        10.0          JMP     TSTEND          ;TERMINATE TESTING NOW! - FATAL ERROR
1640
1641                              ;HERE TO SEE IF 'XFER' BIT TIMEOUT HAPPENED
1642
1643  43AC  0D                4.0  TST1D:  DCR     C               ;DECREMENT 'XFER' WATCHDOG COUNTER
1644  43AD  C2  7C  43        10.0          JNZ     TST1B           ;BACK TO LOOP IF OK
1645
1646  43B0                          ERR     TST1L,TST1DC
 (1)                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  43B0  CD  09  28        18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43B3  02                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43B4  00                      .BYTE
 (1)  43B5  CD  15  28        18.0  TST1DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43B8  DA  0C  43        10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1647                              ;>WMC ROM PGM FAILED TO SET 'XFER' BIT IN 128 CLOCK CYCLES AFTER XFR STARTED
1648                              ;<TEST ABORTED.
1649  43BB  C3  18  28        10.0          JMP     TSTEND
```

```
1651  43BE                                  S
 (1)                                        ; ***************************************************************************
1652                                        ;SCLKBIT -- HERE WHEN A CLOCK CYCLE RESULTS IN 'SCLK' SETTING IN DBUS
1653                                        ;       CONTROL WORD.  LOAD NEXT DATA SET (18 BITS) INTO DDR REGS.
1654  43BE                                  S
 (1)                                        ; ***************************************************************************
1655
1656  43BE  3A  E8  45      13.0  SCLKBIT: LDA     CLKON           ;SEE IF 'SCLK' WAS ON
1657  43C1  A7                4.0          ANA     A               ;SET CONDITION BITS
1658  43C2  C0               12.0          RNZ                     ;EXIT IF CLOCK WAS SET-HAVE DATA ALREADY
1659  43C3  CD  27  46       18.0          CALL    SETDDR          ;THEN LOAD THE DATA INTO THE DDR REGS
1660  43C6  C5               12.0          PUSH    B               ;SAVE REG B & C
1661  43C7  06  14            7.0          MVI     B,20            ;GET A WATCHDOG COUNT
1662  43C9  CD  46  48       18.0  SCKWT:  CALL    CLKSYS          ;CLOCK THE SYSTEM - WAIT FOR 'SCLK' TO GO AWAY
1663  43CC  DB  D0           10.0          IN      WMCSTA          ;GET THE WMC STATUS WORD
1664  43CE  E6  20            7.0          ANI     W.XFER          ;DID 'XFER' BIT SET?
1665  43D0  CA  D6  43       10.0          JZ      SCKWT1          ;NO - JUMP OVER
1666  43D3  32  F6  45       13.0          STA     SXFER           ;YUP - SET THE FLAG
1667  43D6  DB  C0           10.0  SCKWT1: IN      DBUSSTA         ;GET THE DATA BUS STATUS INFO
1668  43D8  E6  02            7.0          ANI     T.SCLK          ;SEE IF SCLK IS GONE YET
1669  43DA  CA  EF  43       10.0          JZ      SCKGON          ;YUP - PROCEED
1670  43DD  05                4.0          DCR     B               ;NOPE! - DOWNCOUNT WATCHDOG
1671  43DE  C2  C9  43       10.0          JNZ     SCKWT           ;STAY IN LOOP TILL NO 'SCLK' OR DIE
1672  43E1                                  ERR     SCKWTE,SCKWTC
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  43E1  CD  09  28       18.0                   CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0003                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43E4  03                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43E5  00                          .BYTE
 (1)  43E6  CD  15  28       18.0  SCKWTC::         CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43E9  DA  19  44       10.0                   JC      SCKWTE          ;LOOP ADDRESS IF LOOP SPECIFIED
1673                                  ;>WMC FAILED TO DROP 'SCLK' BIT AFTER SEVERAL SYSTEM CLOCKS
1674                                  ;<TEST ABORTED.
1675  43EC  C3  18  28       10.0          JMP     TSTEND          ;EXIT IF NOT LOOPING ON ERROR
1676  43EF  C1               10.0  SCKGON: POP     B               ;ALL OK - NOW RESET REG B & C
1677  43F0  3A  FB  45       13.0          LDA     SCCNTR          ;GET COUNTDOWN TO RELOAD DDR DATA
1678  43F3  3D                4.0          DCR     A
1679  43F4  32  EB  45       13.0          STA     SCCNTR          ;SAVE THE UPDATED COUNT
1680  43F7  C2  02  44       10.0          JNZ     SCKB1           ;JMP IF NOT TIME TO RELOAD DDR
1681
1682  43FA  CD  F8  45       18.0          CALL    GENDAT          ;GENERATE NEW DDR DATA FOR INJECTION
1683  43FD  3E  04            7.0          MVI     A,4             ;RESET THE COUNTER
1684  43FF  32  EB  45       13.0          STA     SCCNTR          ;FOR THE NEXT SET OF 'SCLK' CYCLES
1685
1686  4402  06  7F            7.0  SCKB1:  MVI     B,@177          ;GET NEW TIMEOUT COUNT
1687  4404  78                4.0          MOV     A,B             ;COPY TO REG A
1688  4405  32  E8  45       13.0          STA     CLKON           ;SET SCLK "ON" FLAG
1689  4408  3A  EA  45       13.0          LDA     CLKCNT          ;GET THE CURRENT SCLK COUNT
1690  440B  3C                4.0          INR     A               ;UPDATE IT
1691  440C  32  EA  45       13.0          STA     CLKCNT          ;SAVE THE NEW COUNT
1692
1693  440F  CD  A4  45       18.0          CALL    LFTCHK          ;SEE IF 'LEFT' CHECK IS NECESSARY
1694  4412  3A  E6  45       13.0          LDA     LFTTST          ;SEE IF HAVE TO TEST FOR LOADING DDR IN 'LEFT' MODE
1695  4415  A7                4.0          ANA     A               ;SET THE CONDITION BITS
```

```
1696  4416  C4  C3  45      18.0          CNZ     LFTCMP          ;COMPLIMENT THE 'LFTFLG' FLAG
1697  4419  C9              10.0  SCKWTE: RET
1698  441A                        S
 (1)                                ; ***************************************************************************
1699                               ;CKSCLK -- ROUTINE TO CLOCK THE SYSTEM ONCE AND THEN CHECK FOR PRESENCE
1700                               ;       OF 'SCLK' FROM THE WMC. IF 'SCLK' IS ON, THEN LOAD NEW DATA INTO
1701                               ;       THE DDR.
1702  441A                         S
 (1)                                ; ***************************************************************************
1703
1704  441A  CD  46  48      18.0  CKSCLK: CALL    CLKSYS          ;CLOCK SYSTEM ONCE
1705  441D  DB  C0          10.0          IN      DBUSSTA         ;GET STATUS OF DATA BUS
1706  441F  E6  02           7.0          ANI     T.SCLK          ;SAVE ONLY THE 'SCLK' BIT
1707  4421  C2  29  44      10.0          JNZ     CKSC1           ;JUMP IF SCLK IS ON
1708  4424  AF               4.0          XRA     A               ;NO SCLK - CLEAR CLOCK ON FLAG
1709  4425  32  E8  45      13.0          STA     CLKON
1710  4428  C9              10.0          RET                     ;EXIT
1711
1712  4429  CD  BE  43      18.0  CKSC1:  CALL    SCLKBIT         ;LOAD NEW DATA INTO DDR REGS
1713  442C  C9              10.0          RET                     ;EXIT
```

```
1715  442D                                  S
 (1)                                        ; ***************************************************************************
1716                                        ;XFRBIT -- HERE WHEN WMC STATUS INDICATES 'XFER' BIT IS SET. ISSUE 2
1717                                        ;       MORE CLOCKS THEN CHECK WRITE DATA WORD - THIS IS OUTPUT OF BAL
1718                                        ;       AND SHOULD COMPARE TO BYTE STORED IN 'BALDAT' COMPARE BUFFER.
1719                                        ;       REG 'D' + 'E' CONTAIN POINTER TO EXPECTED DATA BYTE.  REG 'C'
1720                                        ;       CONTAINS THE NUMBER OF BYTES TO CHECK FOR EACH 4 SCLK CYCLE.
1721  442D                                  S
 (1)                                        ; ***************************************************************************
1722
1723  442D  C5              12.0  XFRBIT: PUSH    B               ;SAVE REG B
1724  442E  DB  D0          10.0          IN      WMCSTA          ;GET THE CURRENT WMC STATUS
1725  4430  32  F5  45      13.0          STA     SWMCSTA         ;SAVE IT FOR THE ECC CHECK (LATER)
1726  4433  06  14           7.0          MVI     B,20            ;ALLOW UP TO 20. WMC CLOCKS TO DROP XFER
1727  4435  CD  1A  44      18.0  XFRB1:  CALL    CKSCLK          ;NEED TWO MORE CLOCK CYCLES BEFORE
1728  4438  DA  0C  43      10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
1729  443B  DB  D0          10.0          IN      WMCSTA          ;GET WMC STATUS
1730  443D  E6  20           7.0          ANI     W.XFER          ;SAVE THE XFER BIT
1731  443F  CA  4B  44      10.0          JZ      XFRGON          ;OK TO CONTINUE IF XFER IS GONE
1732  4442  05               4.0          DCR     B               ;DECREMENT WATCHDOG COUNTER
1733  4443  C2  35  44      10.0          JNZ     XFRB1           ;LOOP TIL =0 OR XFER IS GONE
1734  4446                                ERR     TST1L,XFRGON
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4446  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0004                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4449  04                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  444A  00                               .BYTE
 (1)  444B  CD  15  28      18.0  XFRGON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  444E  DA  0C  43      10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1735                                        ;>WMC FAILED TO CLEAR 'XFER' BIT ON 1ST NIBBLE OF THIS BYTE ASSY
1736                                        ;HERE TO LOOK FOR 'XFER' TO SET FOR THE 2ND NIBBLE OF THIS BYTE
1737
1738  4451  06  7F           7.0          MVI     B,@177
1739  4453  CD  1A  44      18.0  XFRB2:  CALL    CKSCLK          ;SEE IF SCLK IS SET (ALSO CLOCK)
1740  4456  DA  0C  43      10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
1741  4459  DB  D0          10.0          IN      WMCSTA          ;GET WMC STATUS
1742  445B  E6  20           7.0          ANI     W.XFER          ;SAVE ONLY XFER BIT
1743  445D  C2  72  44      10.0          JNZ     XFRB3           ;JUMP IF XFER IS SET - OK
1744  4460  05               4.0          DCR     B               ;LOOK FOR TIMEOUT
1745  4461  C2  53  44      10.0          JNZ     XFRB2           ;GO BACK TO LOOP IF OK
1746  4464                                ERR     TST1L,XFRB2C,3
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4464  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0005                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4467  05                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4468  03                               .BYTE   3
 (1)  4469  CD  15  28      18.0  XFRB2C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  446C  DA  0C  43      10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1747                                        ;>WMC FAILED TO SET 'XFER' BIT FOR 2ND NIBBLE ON THIS BYTE ASSY
1748                                        ;<TEST ABORTED!
1749  446F  C3  18  28      10.0          JMP     TSTEND
1750
1751                                        ;HERE TO LOOK FOR 'XFER' BIT TO GO AWAY TO INDICATE FULL BYTE IS AVAIL.
1752                                        ;FROM THE 'WRTDAT' WORD
```

```
1753
1754  4472  06  14        7.0  XFRB3:  MVI   B,20
1755  4474  DB  E0       10.0          IN    INTSTA          ;GET THE DATA PARITY BIT
1756  4476  32  F7  45   13.0          STA   SINTSTA         ;SAVE FOR PARITY CHECK
1757  4479  CD  1A  44   18.0  XFRB3A: CALL  CKSCLK           ;CLOCK THE SYSTEM TILL XFER BIT GOES AWAY
1758  447C  DA  0C  43   10.0          JC    TST1L           ;JUMP IF ERROR DETECTED
1759  447F  DB  D0       10.0          IN    WMCSTA          ;GET THE WMC STATUS
1760  4481  E6  20        7.0          ANI   W.XFER          ;SAVE ONLY THE XFER BIT
1761  4483  CA  98  44   10.0          JZ    XFRB4           ;JUMP IF GONE - OK
1762  4486  05         4.0          DCR   B               ;DOWNCOUNT WATCHDOG
1763  4487  C2  79  44   10.0          JNZ   XFRB3A          ;BACK TO LOOP IF OK
1764  448A                               ERR   TST1L,XFRB3C,3
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  448A  CD  09  28   18.0                    CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0006                                 MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  448D  06                                   .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  448E  03                                   .BYTE 3
 (1)  448F  CD  15  28   18.0          XFRB3C::  CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4492  DA  0C  43   10.0                    JC    TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1765                               ;>WM'  AILED TO CLEAR 'XFER' BIT FOR 2ND NIBBLE OF THIS BYTE ASSY
1766                               ;<T'   ABORTED!
1767  4495  C3  18  28   10.0          JMP   TSTEND
1768                               ;HERE TO TEST THE ASSEMBLED BYTE DATA AGAINST THE EXPECTED BYTE DATA
1769
1770  4498  CD  46  48   18.0  XFRB4:  CALL  CLKSYS          ;ALLOW THE BYTE COUNTER TO TICK
1771  449B  CD  46  48   18.0          CALL  CLKSYS          ;NEED 2 CLOCKS FOR 1 WMC INSTR.
1772  449E  C1         10.0          POP   B               ;RESTORE REG B
1773  449F  3A  E9  45   13.0          LDA   ECCCNT          ;GET 'XFER' COUNT
1774  44A2  3C         4.0          INR   A               ;ADD 1
1775  44A3  32  E9  45   13.0          STA   ECCCNT          ;SAVE THE UPDATED COUNT
1776  44A6  FE  08        7.0          CPI   8               ;TIME TO CHECK 'ECC' BIT?
1777  44A8  CA  87  45   10.0          JZ    ECCCHK          ;TEST 'ECC' IF COUNT=8
1778
1779                               ;GET ASSEMBLED BAL DATA BYTE FROM WRITE DATA WORDS
1780
1781  44AB  3A  E1  45   13.0          LDA   ABCNT           ;GET THE BYTE # UNDER TEST
1782  44AE  3C         4.0          INR   A               ;ADD 1
1783  44AF  32  E1  45   13.0          STA   ABCNT           ;SAVE THE BYTE NUMBER - UPDATED
1784  44B2                               ROUT  R05L
 (1)  44B2  D3  8A       10.0                    OUT   R05L            ;WRITE AC INTO R05L
 (1)  44B4  7F         4.0                    MOV   A,A             ;RETRY LINK
1785  44B5  AF         4.0          XRA   A
1786  44B6                               ROUT  R05H
 (1)  44B6  D3  8B       10.0                    OUT   R05H            ;WRITE AC INTO R05H
 (1)  44B8  7F         4.0                    MOV   A,A             ;RETRY LINK
1787  44B9  EB         4.0          XCHG                        ;PUT 'EXPECTED' POINTER IN H & L
1788  44BA  7E         7.0          MOV   A,M                   ;GET CORRECT BAL BYTE
1789  44BB                               ROUT  EDATA                 ;SAVE 'EXPECTED' DATA
 (1)  44BB  D3  95       10.0                    OUT   EDATA           ;WRITE AC INTO EDATA
 (1)  44BD  7F         4.0                    MOV   A,A             ;RETRY LINK
1790  44BE  DB  D3       10.0          IN    WRTDAT          ;GET ASSEMBLED DATA BYTE
1791  44C0                               ROUT  ADATA               ;SAVE 'ACTUAL' DATA
 (1)  44C0  D3  94       10.0                    OUT   ADATA           ;WRITE AC INTO ADATA
 (1)  44C2  7F         4.0                    MOV   A,A             ;RETRY LINK
```

```
1792  44C3  BE              7.0          CMP     M               ;EXPECTED = ACTUAL?
1793  44C4  EB              4.0          XCHG                    ;RESET H & L REGS
1794  44C5  13              6.0          INX     D               ;POINT TO NEXT 'EXPECTED' ENTRY
1795  44C6  CA  D4  44      10.0         JZ      TST1XC          ;JUMP OVER ERROR IF DATA BYTE OK
1796  44C9  3A  E1  45      13.0         LDA     ABCNT           ;GET THE COUNT OF ASSEM. BYTES AT FAILURE
1797  44CC  32  E3  45      13.0         STA     EABCNT          ;SAVE FOR THE LOOP CONTROL
1798  44CF                               ERRB    DELOOP,TST1XC,3
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44CF  CD  12  28        18.0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0007                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44D2  07                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44D3  03                         .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  44D4  CD  15  28        18.0     TST1XC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44D7  DA  6C  45        10.0                    JC      DELOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1799                                ;>WMC BYTE ASSEMBLY FAULT - BYTE ASSEMBLED DOESN'T MATCH EXPECTED DATA
1800                                ;HERE TO CHECK THE PARITY BIT ON THE DATA ASSEMBLED
1801
1802  44DA  DB  D3          10.0   XFRB4P: IN      WRTDAT          ;GET THE DATA BYTE AGAIN
1803  44DC  A7              4.0            ANA     A               ;SET THE CONDITION BITS
1804  44DD  E4  EC  44      18.0           CPO     DATPO           ;CHECK FOR ODD PARITY
1805  44E0  DA  0C  43      10.0           JC      TST1L           ;JUMP IF ERROR DETECTED
1806  44E3  EC  02  45      18.0           CPE     DATPE           ;CHECK FOR PARITY EVEN - BIT -1
1807  44E6  DA  0C  43      10.0           JC      TST1L           ;JUMP IF ERROR DETECTED
1808  44E9  C3  14  45      10.0           JMP     XFRB5           ;ON TO THE NEXT SECTION....
1809
1810                                ;HERE TO CHECK FOR PARITY ODD - BIT -0
1811
1812  44EC  F5              12.0   DATPO:  PUSH    PSW             ;SAVE THE PSW STATUS
1813  44ED  3A  F7  45      13.0           LDA     SINTSTA         ;GET THE PARITY BIT (IN BIT 0)
1814  44F0  E6  10           7.0           ANI     WDR.P           ;SAVE JUST THE PARITY BIT
1815  44F2  CA  00  45      10.0           JZ      DATPOE          ;EXIT IF OK
1816  44F5                               ERR     DATPOE,DATPOC
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44F5  CD  09  28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0008                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44F8  08                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44F9  00                         .BYTE
 (1)  44FA  CD  15  28        18.0     DATPOC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44FD  DA  00  45        10.0                    JC      DATPOE          ;LOOP ADDRESS IF LOOP SPECIFIED
1817                                ;>WMC PARITY BIT ON ASSEMBLED DATA SHOULD =0 (ODD PARITY) AND WAS =1
1818
1819  4500  F1              10.0   DATPOE: POP     PSW             ;RESET THE PSW STATUS
1820  4501  C9              10.0           RET
1821
1822                                ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
1823
1824  4502  3A  F7  45      13.0   DATPE:  LDA     SINTSTA         ;GET THE BIT
1825  4505  E6  10           7.0           ANI     WDR.P           ;SAVE JUST THE PARITY DATA BIT
1826  4507  C0              12.0           RNZ
```

```
1828  4508                                        ERR     DATPEE,DATPEC
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4508  CD  09  28        18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0009                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  450B  09                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  450C  00                                           .BYTE
 (1)  450D  CD  15  28        18.0                DATPEC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4510  DA  13  45        10.0                        JC      DATPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
1829                                    ;>WMC PARITY BIT ON ASSEMBLED DATA WAS =0 AND SHOULD BE =1
1830  4513  C9                10.0    DATPEE: RET
1831  4514  3A  E2  45        13.0    XFRB5:  LDA     ABCNTR          ;GET COUNT OF BYTES LEFT THIS CYCLE
1832  4517  3D                 4.0            DCR     A               ;DECREMENT # OF ASSEMBLED BYTES - COUNT
1833  4518  32  E2  45        13.0            STA     ABCNTR          ;SAVE IT
1834  451B  CC  D3  46        18.0            CZ      GCDAT           ;GENERATE NEW 'EXPECTED' BUFFER BASED ON
1835                                                                  ;CURRENT DDR DATA
1836  451E  06  7F              7.0           MVI     B,@177
1837  4520  48                 4.0           MOV     C,B             ;RESET WATCHDOG FOR 'XFER' BIT
1838  4521  E5                12.0           PUSH    H
1839  4522  21  E0  45        10.0           LXI     H,IMBYTC        ;POINT TO THE AMT OF BYTES EXPECTED
1840  4525  3A  E1  45        13.0           LDA     ABCNT           ;THEN GET # BYTES ASSEMBLED BY THE WMC
1841  4528  BE                 7.0           CMP     M               ;SEE IF DONE
1842  4529  E1                10.0           POP     H               ;RESET REG H & L
1843  452A  C2  9A  43        10.0           JNZ     TST1C           ;BACK TO LOOP IF MORE TO DO
1844
1845                                    ·DUMMY READ OF BYTE COUNTER
1846  452D  DB  D4             10.0           IN      BYTCNT
1847  452F  DB  D4             10.0           IN      BYTCNT
1848
1849  4531  DB  D4             10.0           IN      BYTCNT          ;GET BYTE COUNTER 7-0
1850  4533  47                 4.0           MOV     B,A             ;TEMP STORE
1851  4534  DB  D4             10.0           IN      BYTCNT          ;GET BYTE CNT 15-8
1852  4536  78                 4.0           MOV     A,B
1853  4537  A7                 4.0           ANA     A
1854  4538  CA  43  45        10.0           JZ      XFRB5C          ;JUMP IF BYTE COUNT OK
1855  453B                                    ROUT    ADATA           ;STORE THE ACTUAL BYTE COUNTER 7-0
 (1)  453B  D3  94             10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  453D  7F                  4.0                   MOV     A,A             ;RETRY LINK
1856  453E                                    ERRA    TST1L,XFRB5C,3
 (1)                                            ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  453E  CD  0F  28        18.0                        CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       000A                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4541  0A                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4542  03                                           .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  4543  CD  15  28        18.0                XFRB5C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4546  DA  0C  43        10.0                        JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1857                                    ;>WMC BYTE COUNTER EXPECTED TO BE =0 - ALL BYTES RECEIVED FROM WMC
1858
1859                                    ;HERE TO UPDATE 'FORMAT' COUNT TO NEXT CODE NUMBER
1860
1861  4549  3A  E4  45        13.0            LDA     FORMT           ;GET CURRENT #
1862  454C  3C                 4.0            INR     A               ;INCREMENT
1863  454D  32  E4  45        13.0            STA     FORMT           ;SAVE THE NEW FORMAT CODE NUMBER
1864  4550  FE  07             7.0            CPI     @7              ;DONE WITH THIS TEST?
1865  4552  C2  0C  43        10.0            JNZ     TST1L           ;LOOP BACK TO TEST THIS FORMAT MODE
```

```
1867  4555                                    ENDTST  TST1ST
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4555                            REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4555  CD 06 28      18.0                        CALL    REQST
 (2)  4558  00                                        .BYTE                   ;DATA PATTERN NUMBER
 (2)  4559  00 00                                     .WORD                   ;SYSTEM '"' COUNT
 (2)  455B  00 00                                     .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  455D  00                                        .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  455E  07                                        .BYTE   7               ;REQUEST CODE
 (1)  455F  3A 9A 4F      13.0        LDA     ITERA           ;GET ITERATION COUNT
 (1)  4562  3D             4.0        DCR     A               ;DOWNCOUNT
 (1)  4563  32 9A 4F      13.0        STA     ITERA           ;SAVE COUNT
 (1)  4566  F2 05 43      10.0        JP      TST1ST          ;DO TEST UNTIL TILL = 0
1868  4569  C3 18 28      10.0        JMP     TSTEND          ;ALL DONE
1869                                 ;HERE IS THE DATA COMPARE ERROR LOOP CONTROL ROUTINE
1870
1871                                 ;WILL ENABLE THE TEST TO LOOP AT THE SPECIFIED BYTE IN ERROR
1872
1873  456C  3A 94 4F      13.0  DELOOP: LDA     LPFLG           ;SEE IF LOOP FLAG STILL SET
1874  456F  A7             4.0        ANA     A
1875  4570  C2 79 45      10.0        JNZ     DELP1           ;JUMP IF LOOPING ON ERROR
1876  4573  32 E3 45      13.0        STA     EABCNT          ;NOT LOOPING - CLEAR THE FLAG
1877  4576  C3 14 45      10.0        JMP     XFRB5           ;AND EXIT
1878
1879  4579  3A E1 45      13.0  DELP1:  LDA     ABCNT           ;GET THE CURRENT COUNT OF ASSEMBLED BYTES
1880  457C  47             4.0        MOV     B,A             ;TEMP STORE
1881  457D  3A E3 45      13.0        LDA     EABCNT          ;GET THE BYTE COUNT AT THE DETECTED ERROR
1882  4580  B8             4.0        CMP     B               ;SEE IF AT THE SAME BYTE COUNT
1883  4581  CA 0C 43      10.0        JZ      TST1L           ;THEN JUMP TO RESTART TEST IF SAME
1884  4584  C3 14 45      10.0        JMP     XFRB5           ;ELSE, CONTINUE WITH THE TEST
1885
1886                                 ;HERE TO THROW AWAY ECC CHARACTER GENERATED EVERY 8TH BYTE OUTPUT FROM
1887                                 ;WMC.
1888
1889  4587  AF             4.0  ECCCHK: XRA     A               ;CLEAR REG A
1890  4588  32 E9 45      13.0        STA     ECCCNT          ;ZERO COUNT OF 'XFER' CYCLES
1891  458B  3A F5 45      13.0        LDA     SWMCSTA         ;GET THE 'STORED' WMC STATUS
1892  458E                            ROUT    ADATA           ;SAVE FOR ERROR PRINTOUT
 (1)  458E  D3 94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4590  7F             4.0                 MOV     A,A             ;RETRY LINK
1893  4591  E6 10           7.0        ANI     W.ECC           ;SAVE 'ECC' BIT
1894  4593  C2 9A 43      10.0        JNZ     TST1C           ;JUMP BACK TO CLOCK LOOP IF OK ('ECC' BIT WAS ON)
1895  4596                            ERRA    ECCON1,ECCON,3
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4596  CD 0F 28      18.0                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000B                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4599  0B                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  459A  03                        .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  459B  CD 15 28      18.0        ECCON:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  459E  DA A1 45      10.0                JC      ECCON1          ;LOOP ADDRESS IF LOOP SPECIFIED
1896                                 ;>WMC FAILED TO SET 'ECC' IN WMC STATUS ON 8TH 'XFER' CYCLE
1897                                 ;>ECC CHAR. NOT PRESENT ON WMC OUTPUT
1898  45A1  C3 18 28      10.0  ECCON1: JMP     TSTEND
```

```
                                        S
                                        ; ***************************************************************************
                                        ;LFTCHK -- ROUTINE TO TEST 'WMC LEFT' WHEN IN ANY 10 PACKING MODE.  'WMC
                                        ;       LEFT' WILL BE 0 FOR 11 NORMAL, 15 NORMAL, AND IMAGE MODE.
                                        S
                                        ; ***************************************************************************

45A4  3A E6 45      13.0  LFTCHK: LDA     LFTTST          ;GET CHECK FLAG
45A7  A7             4.0          ANA     A               ;SET CONDITION BITS
45A8  C8            12.0          RZ                      ;EXIT IF NO TEST NEEDED NOW
45A9  3A E7 45      13.0          LDA     LFTFLG          ;GET 'WMC LEFT' ON/OFF FLAG
45AC  A7             4.0          ANA     A               ;CHECK TO SEE IF ITS OFF
45AD  CA CB 45      10.0          JZ      LFTOFF

                                  ;HERE TO SEE IF ANY 10 XFR MODE HAS SET WMC LEFT STATUS

45B0  DB A1         10.0  LFTON:  IN      CBUSSTA         ;GET STATUS
45B2  E6 04          7.0          ANI     W.LEFT          ;SAVE ONLY 'WMC LEFT' BIT
45B4  C2 BC 45      10.0          JNZ     LFTOK2          ;JUMP IF 'LEFT' IS ON (ITS EXPECTED)
45B7                              ERR     TST1L,LFTOK2,2
 (1)                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45B7  CD 09 28  18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000C                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45BA  0C                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45BB  02                                .BYTE   2
 (1)  45BC  CD 15 28  18.0            LFTOK2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45BF  DA 0C 43  10.0                    JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                  ;>WMC FAILED TO SET 'WMC LEFT' BIT DURING 10 PACKING MODE
45C2  C9            10.0          RET

                                  ;HERE TO COMPLEMENT THE STATE OF THE LEFT MODE CHECK FLAG

45C3  3A E7 45      13.0  LFTCMP: LDA     LFTFLG          ;GET CURRENT 'LEFT' FLAG
45C6  2F             4.0          CMA                     ;COMPLEMENT-NEXT SCLK 'LEFT' SHOULD BE OFF
45C7  32 E7 45      13.0          STA     LFTFLG          ;SAVE IT
45CA  C9            10.0          RET
                                  ;HERE TO CHECK THAT ANY 10 XFR MODE HAS CLEARED WMC LEFT FLAG

45CB  DB A1         10.0  LFTOFF: IN      CBUSSTA         ;GET CBUS STATUS INFO
45CD  E6 04          7.0          ANI     W.LEFT          ;SAVE ONLY THE WMC LEFT BIT
45CF  CA D7 45      10.0          JZ      LFTOK1          ;JUMP IF NOT ON - OK
45D2                              ERR     TST1L,LFTOK1,2
 (1)                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45D2  CD 09 28  18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000D                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45D5  0D                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45D6  02                                .BYTE   2
 (1)  45D7  CD 15 28  18.0            LFTOK1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45DA  DA 0C 43  10.0                    JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                  ;>WMC FAILED TO CLEAR 'WMC LEFT' BIT DURING 10 PACKING MODE
45DD  C9            10.0          RET                     ;EXIT
```

```
1936                                    ;HERE IS THE NECESSARY STORAGE FOR TEST PARAMETERS AND DDR DATA STORAGE
1937
1938  45DE  00                          CCTLWD: .BYTE   0               ;IMAGE OF CLOCK CONTROL WORD
1939  45DF  00                          DCNTWD: .BYTE   0               ;COUNTER FOR TOTAL BYTES ASSEMBLED 4 SCLKS
1940  45E0  00                          IMBYTC: .BYTE   0               ;IMAGE OF BYTE COUNTER 7-0 BITS
1941  45E1  00                          ABCNT:  .BYTE   0               ;ASSEMBLED BYTE COUNTER
1942  45E2  00                          ABCNTR: .BYTE   0
1943  45E3  00                          EABCNT: .BYTE   0               ;BYTE COUNT AT A DETECTED ERROR
1944  45E4  00                          FORMT:  .BYTE   0               ;FORMAT CODE SELECTED FOR TEST
1945  45E5  00                          DPNUM:  .BYTE   0               ;DATA PATTERN OFFSET NUMBER (0 TO 17)
1946  45E6  00                          LFTTST: .BYTE   0               ;'WMC LEFT' TEST NEEDED FLAG
1947  45E7  00                          LFTFLG: .BYTE   0               ;'WMC LEFT' ON/OFF FLAG
1948  45E8  00                          CLKON:  .BYTE   0               ;'SCLK' ON FLAG-USED TO TELL WHEN TO FILL DDR REGS
1949  45E9  00                          ECCCNT: .BYTE   0               ;COUNT BY 7 FOR ECC CHARACTER CHECK
1950  45EA  00                          CLKCNT: .BYTE   0               ;COUNT OF SCLKS GENERATED
1951  45EB  00                          SCCNTR: .BYTE   0               ;COUNT TO RELOAD THE DDR REGS
1952  45EC  00                          SDDRA:  .BYTE   0               ;DDRA DATA ACTUALLY LOADED
1953  45ED  00                          SDDRB:  .BYTE   0               ;DDRB DATA ACTUALLY LOADED
1954  45EE  00                          SDDRC:  .BYTE   0               ;DDRC DATA ACTUALLY LOADED
1955
1956  45EF  00                          SDDRAT: .BYTE   0               ;DDRA DATA FROM PGM TABLE
1957  45F0  00                          SDDRBT: .BYTE   0               ;DDRB DATA FROM PGM TABLE
1958  45F1  00                          SDDRCT: .BYTE   0               ;DDRC DATA FROM PGM TABLE
1959
1960  45F2  00                          SDDRAC: .BYTE   0               ;COPY OF DDRA 'TEMP' DATA
1961  45F3  00                          SDDRBC: .BYTE   0               ;COPY OF DDRB 'TEMP
1962  45F4  00                          SDDRCC: .BYTE   0               ;COPY OF DDRC 'TEMP
1963  45F5  00                          SWMCSTA: .BYTE  0               ;COPY OF THE WMC STATUS AT 'XFER' BIT TIME
1964  45F6  00                          SXFER:  .BYTE   0               ;SOFTWARE 'XFER' FLAG
1965  45F7  00                          SINTSTA: .BYTE  0               ;SOFTWARE COPY OF CBUSSTA WORD
```

```
1967  45F8                              S
 (1)                                    ; ***************************************************************************
1968                                    ;GENDAT -- ROUTINE TO GENERATE DDR REG DATA FOR NEXT "SCLK" CYCLE.
1969                                    ;       DATA IS COADED IN DDR REGS AND A COPY IS STORED IN 'SDDRAC', +
1970                                    ;       'SDDRCC' FOR REFERENCE.
1971  45F8                              S
 (1)                                    ; ***************************************************************************
1972
1973  45F8  F5              12.0  GENDAT: PUSH    PSW             ;SAVE PSW + A
1974  45F9  C5              12.0        PUSH    B               ;SAVE B + C
1975  45FA  E5              12.0        PUSH    H               ;SAVE H + L
1976  45FB  3A  E5  45      13.0        LDA     DPNUM           ;GET THE LAST PATTERN # USED
1977  45FE  3C               4.0        INR     A               ;POINT TO NEXT
1978  45FF  32  E5  45      13.0        STA     DPNUM           ;SAVE NUMBER
1979  4602  47               4.0        MOV     B,A
1980
1981  4603  87               4.0        ADD     A
1982  4604  80               4.0        ADD     B               ;MULT BY 3 FOR OFFSET
1983  4605  4F               4.0        MOV     C,A
1984  4606  06  00            7.0        MVI     B,0             ;B + C HAVE OFFSET
1985  4608  21  88  46       10.0        LXI     H,CDATBL        ;POINT TO DATA TABLE
1986  460B  09               10.0        DAD     B               ;ADD IN OFFSET TO H + L
1987
1988  460C  7E                7.0        MOV     A,M             ;GET DATA BYTE FOR DDR 'A'
1989  460D  32  EF  45       13.0        STA     SDDRAT          ;SAVE DDR 'A' IMAGE
1990  4610  32  F2  45       13.0        STA     SDDRAC          ;LOAD DDR 'A' COPY
1991  4613  23                6.0        INX     H               ;POINT TO NEXT DATA BYTE
1992  4614  7E                7.0        MOV     A,M             ;GET IT
1993  4615  32  F0  45       13.0        STA     SDDRBT          ;SAVE DDR 'B' IMAGE
1994  4618  32  F3  45       13.0        STA     SDDRBC          ;SAVE DDRB COPY
1995  461B  23                6.0        INX     H               ;POINT TO NEXT DATA BYTE
1996  461C  7E                7.0        MOV     A,M             ;GET IT
1997  461D  32  F1  45       13.0        STA     SDDRCT          ;SAVE DDR 'C' IMAGE
1998  4620  32  F4  45       13.0        STA     SDDRCC          ;SAVE DDRC COPY
1999
2000  4623  E1               10.0  GENOUT: POP     H               ;RESTORE REGS
2001  4624  C1               10.0        POP     B
2002  4625  F1               10.0        POP     PSW
2003  4626  C9               10.0        RET
```

```
2005  4627                              S
 (1)                                    ; ***************************************************************************
2006                                    ;SETDDR -- ROUTINE TO TRANSLATE THE DESIRED DATA FROM A 1-FOR-1 BIT
2007                                    ;       POSITION TO THE DDR TO BE USED AS DATA TO EITHER THE TM78 OR
2008                                    ;       THE MASSBUS. WMC "LEFT" FLAG SET INDICATES THAT THIS ROUTINE IS NOT
2009                                    ;       NECESSARY, THE DATA IS IN THE CORRECT FORMAT ALREADY. IF WMC
2010                                    ;       "LEFT" FLAG IS NOT SET, THE DATA MUST BE TRANSLATED TO A COMPATABLE
2011                                    ;       DDR FORMAT.
2012  4627                              S
 (1)                                    ; ***************************************************************************
2013
2014  4627  3A  E7  45      13.0  SETDDR: LDA     LFTFLG          ;GET THE SOFTWARE 'LEFT' MODE FLAG
2015  462A  A7                    4.0         ANA     A               ;SET THE CONDITION BITS
2016  462B  C2  66  46      10.0         JNZ     SLFTDDR         ;WMC SET TO "LEFT" MODE - NO TRANSLATION
2017
2018                                    ;HERE TO TRANSLATE THE DATA INTO AN ACCEPTABLE FORMAT FOR DDR
2019
2020  462E  3A  EF  45      13.0         LDA     SDDRAT          ;GET DDRA TEMP
2021  4631  E6  3F                7.0         ANI     @77             ;STRIP 2 BITS OFF
2022  4633  07                    4.0         RLC
2023  4634  07                    4.0         RLC                     ;LEFT JUSTIFY
2024  4635  47                    4.0         MOV     B,A             ;TEMP SAVE
2025  4636  3A  F1  45      13.0         LDA     SDDRCT          ;GET DDRC TEMP
2026  4639  E6  03                7.0         ANI     @3              ;SAVE 2 BITS
2027  463B  B0                    4.0         ORA     B
2028  463C  32  EC  45      13.0         STA     SDDRA           ;SAVE DDRA DESIRED
2029
2030  463F  3A  F0  45      13.0         LDA     SDDRBT          ;GET DDRB TEMP
2031  4642  E6  3F                7.0         ANI     @77
2032  4644  07                    4.0         RLC
2033  4645  07                    4.0         RLC                     ;LEFT JUSTIFY
2034  4646  47                    4.0         MOV     B,A
2035  4647  3A  EF  45      13.0         LDA     SDDRAT          ;GET DDRA AGAIN
2036  464A  E6  C0                7.0         ANI     @300            ;SAVE 2 BITS
2037  464C  07                    4.0         RLC
2038  464D  07                    4.0         RLC
2039  464E  B0                    4.0         ORA     B
2040  464F  32  ED  45      13.0         STA     SDDRB           ;SAVE DDRB DATA
2041
2042  4652  3A  F1  45      13.0         LDA     SDDRCT          ;GET DDRC TEMP
2043  4655  E6  04                7.0         ANI     @4              ;SAVE THE PARITY BIT
2044  4657  47                    4.0         MOV     B,A             ;TEMP STORE
2045  4658  3A  F0  45      13.0         LDA     SDDRBT          ;GET DDRB AGAIN
2046  465B  E6  C0                7.0         ANI     @300            ;SAVE 2 BITS
2047  465D  07                    4.0         RLC
2048  465E  07                    4.0         RLC                     ;RIGHT JUSTIFY
2049  465F  B0                    4.0         ORA     B
2050  4660  32  EE  45      13.0         STA     SDDRC           ;SAVE DDRC DATA
2051  4663  C3  78  46      10.0         JMP     SDDCOM          ;COMMON EXIT
2052                                    ;HERE IF NO MODIFICATION OF DDR DATA NECESSARY
2053
2054  4666  3A  EF  45      13.0  SLFTDDR: LDA     SDDRAT          ;GET DDRA DATA
2055  4669  32  EC  45      13.0         STA     SDDRA           ;SAVE FOR OUTPUT
2056  466C  3A  F0  45      13.0         LDA     SDDRBT          ;GET TEMP DDRB DATA
```

```
2057  466F  32  ED  45   13.0          STA   SDDRB          ;SAVE IT
2058  4672  3A  F1  45   13.0          LDA   SDDRCT         ;GET DDRC TEMP DATA
2059  4675  32  EE  45   13.0          STA   SDDRC          ;SAVE FOR OUTPUT
2060
2061  4678  3A  EC  45   13.0  SDDCOM: LDA   SDDRA          ;GET DATA TO OUTPUT
2062  467B  D3  D8        10.0          OUT   DDRA
2063  467D  3A  ED  45   13.0          LDA   SDDRB
2064  4680  D3  D9        10.0          OUT   DDRB
2065  4682  3A  EE  45   13.0          LDA   SDDRC
2066  4685  D3  DA        10.0          OUT   DDRC
2067  4687  C9              10.0          RET                  ;EXIT
2068                                ;TABLE OF DATA TO USE FOR 'DDR' LOADING + SCLK REQUESTS
2069                                ;DATA APPEARS IN THE TABLE FOR WMC 'LEFT' XFR MODE...IF WMC IS IN
2070                                ;'RIGHT' MODE, THEN THE DATA IS SHIFTED TO THE LEFT TWO PLACES.
2071
2072  4688  45                    CDATBL: .BYTE   $45          ;HEX 12345
2073  4689  23                            .BYTE   $23
2074  468A  01                            .BYTE   $1
2075  468B  56                            .BYTE   $56          ;HEX 23456
2076  468C  34                            .BYTE   $34
2077  468D  02                            .BYTE   $2
2078  468E  67                            .BYTE   $67          ;HEX 34567
2079  468F  45                            .BYTE   $45
2080  4690  03                            .BYTE   $3
2081  4691  78                            .BYTE   $78          ;HEX 05678
2082  4692  56                            .BYTE   $56
2083  4693  00                            .BYTE   $0
2084  4694  89                            .BYTE   $89          ;HEX 16789
2085  4695  67                            .BYTE   $67
2086  4696  01                            .BYTE   $1
2087  4697  9A                            .BYTE   $9A          ;HEX 2789A
2088  4698  78                            .BYTE   $78
2089  4699  02                            .BYTE   $2
2090  469A  AB                            .BYTE   $AB          ;HEX 389AB
2091  469B  89                            .BYTE   $89
2092  469C  03                            .BYTE   $3
2093  469D  BC                            .BYTE   $BC          ;HEX 90ABC
2094  469E  9A                            .BYTE   $9A
2095  469F  00                            .BYTE   $0
2096  46A0  CD                            .BYTE   $CD          ;HEX 1ABCD
2097  46A1  AB                            .BYTE   $AB
2098  46A2  01                            .BYTE   $1
2099  46A3  DE                            .BYTE   $DE          ;HEX 2BCDE
2100  46A4  BC                            .BYTE   $BC
2101  46A5  02                            .BYTE   $2
2102  46A6  EF                            .BYTE   $EF          ;HEX 3CDEF
2103  46A7  CD                            .BYTE   $CD
2104  46A8  03                            .BYTE   $3
2105  46A9  F0                            .BYTE   $F0          ;HEX 0DEF0
2106  46AA  DE                            .BYTE   $DE
2107  46AB  00                            .BYTE   $0
2108  46AC  01                            .BYTE   $01          ;HEX 1GF01
2109  46AD  EF                            .BYTE   $EF
2110  46AE  01                            .BYTE   $1
```

```
2111  46AF  12                        .BYTE   $12             ;HEX 2F012
2112  46B0  F0                        .BYTE   $F0
2113  46B1  02                        .BYTE   $2
2114  46B2  23                        .BYTE   $23             ;HEX 30123
2115  46B3  01                        .BYTE   $01
2116  46B4  03                        .BYTE   $3
2117  46B5  34                        .BYTE   $34             ;HEX 01234
2118  46B6  12                        .BYTE   $12
2119  46B7  00                        .BYTE   $0
2120  46B8  00              DTEND:    .BYTE   $0              ;0 - END OF TABLE
2121  46B9  00                        .BYTE   $0
2122  46BA  00                        .BYTE   $0
2123
2124                        ;BAL 'EXPECTED' OUTPUT DATA
2125
2126  46BB  00              BALDAT:   .BYTE   0               ;TABLE OF EXPECTED DATA BYTES
2127  46BC  00                        .BYTE   0               ;OUTPUT FROM BYTE ASSEMBLY
2128  46BD  00                        .BYTE   0               ;LOGIC OF WMC
2129  46BE  00                        .BYTE   0
2130  46BF  00                        .BYTE   0
2131  46C0  00                        .BYTE   0
2132  46C1  00                        .BYTE   0
2133  46C2  00                        .BYTE   0
2134  46C3  00                        .BYTE   0
2135  46C4  00                        .BYTE   0
2136
2137                        ;TABLE OF EXPECTED NUMBER OF DATA BYTES OUTPUT FROM BAL FOR EVERY SCLKS
2138
2139  46C5  0C  47           FMTTBL:   .ADDR   F11N            ;0-11 NORMAL - 8 BYTES/4 CLOCKS
2140  46C7  14  47                    .ADDR   F15N            ;1-15 NORMAL - 8 BYTES/4 CLOCKS
2141  46C9  87  47                    .ADDR   F10C            ;2-10 COMPAT - 8 BYTES/4 CLOCKS
2142  46CB  76  47                    .ADDR   F10D            ;3-10 DUMP - 10 BYTES/4 CLOCKS
2143  46CD  F9  47                    .ADDR   F10HD           ;4-10 HIGH DEN COMPAT - 9 BYTES/4 CLOCKS
2144  46CF  34  47                    .ADDR   FIMAG           ;5-IMAGE MODE - 4 BYTES/4 CLOCKS
2145  46D1  F9  47                    .ADDR   F10HD           ;4-10 HIGH DEN DUMP - 9 BYTES/4 CLOCKS
```

```
2147  46D3                          S
 (1)                                ; ***************************************************************
2148                                ;GCDAT -- ROUTINE TO GENERATE THE EXPECTED DATA TABLE OF OUTPUT DATA
2149                                ;       FROM THE BAL.  THIS ROUTINE IS CALLED AFTER 4TH SCLK CYCLE COMPLETE.
2150  46D3                          S
 (1)                                ; ***************************************************************
2151
2152  46D3  F5              12.0  GCDAT:  PUSH   PSW          ;D + E NOW POINT TO STORAGE
2153  46D4  C5              12.0        PUSH   B
2154  46D5  E5              12.0        PUSH   H
2155  46D6  21  BB  46      10.0        LXI    H,BALDAT     ;POINT TO EXPECTED TABLE IN H + L
2156  46D9  EB               4.0        XCHG                ;D + E NOW USED AS POINTERS
2157  46DA  21  C5  46      10.0        LXI    H,FMTTBL     ;POINT TO START OF DATA GEN ROUTINES
2158  46DD  3A  E4  45      13.0        LDA    FORMT        ;GET FMT CODE #
2159  46E0  07               4.0        RLC                 ;MULT BY 2 TO GET OFFSET
2160  46E1  4F               4.0        MOV    C,A
2161  46E2  06  00           7.0        MVI    B,0
2162  46E4  09              10.0        DAD    B            ;H + L NOW POINT TO GEN ROUTINE
2163  46E5  D5              12.0        PUSH   D            ;SAVE POINTER TO TABLE
2164  46E6  5E               7.0        MOV    E,M          ;GET ROUTINE LOW ADDR
2165  46E7  23               6.0        INX    H            ;POINT TO HIGH BYTE ADDRESS OF ROUTINE
2166  46E8  56               7.0        MOV    D,M          ;GET THE HIGH BYTE ADDRESS
2167  46E9  EB               4.0        XCHG                ;MAKE H & L POINT TO ROUTINE
2168  46EA  D1              10.0        POP    D            ;RESET D & E
2169  46EB  E9               6.0        PCHL                ;GO TO PROPER GEN ROUTINE
2170
2171  46EC  21  BB  46      10.0  GCDATX: LXI    H,BALDAT     ;POINT TO START OF 'EXPECTED' BUFFER
2172  46EF  EB               4.0        XCHG                ;PUT POINTER IN REG 'D' & 'E'
2173  46F0  3A  DF  45      13.0        LDA    DCNTWD       ;GET # BYTES TO ASSEMBLE/4 SCLK CYCLE
2174  46F3  32  E2  45      13.0        STA    ABCNTR       ;SAVE FOR LATER TESTS
2175  46F6  E1              10.0        POP    H            ;RESET SAVED REGS
2176  46F7  C1              10.0        POP    B
2177  46F8  F1              10.0        POP    PSW
2178  46F9  C9              10.0        RET                 ;EXIT
2179                              ;ROUTINE TO SHIFT CONTENTS OF ACCUMULATOR RIGHT OR LEFT EITHER 2 OR 4 TIMES
2180
2181  46FA  E6  FC           7.0  SHR4:   ANI    @374         ;STRIP OFF 2 BITS
2182  46FC  0F               4.0        RRC                 ;SHIFT RIGHT 2 TIMES
2183  46FD  0F               4.0        RRC
2184  46FE  E6  FC           7.0  SHR2:   ANI    @374         ;STRIP OFF 2 BITS
2185  4700  0F               4.0        RRC                 ;SHIFT RIGHT 2 TIMES
2186  4701  0F               4.0        RRC
2187  4702  C9              10.0        RET
2188
2189  4703  E6  3F           7.0  SHL4:   ANI    @77          ;SHIFT LEFT TWICE
2190  4705  07               4.0        RLC
2191  4706  07               4.0        RLC
2192  4707  E6  3F           7.0  SHL2:   ANI    @77
2193  4709  07               4.0        RLC                 ;SHIFT LEFT TWICE
2194  470A  07               4.0        RLC
2195  470B  C9              10.0        RET
2196
2197  470C                          S
 (1)                                ; ***************************************************************
```

```
                                        ;F11N -- ROUTINE TO FILL 'EXPECTED' BAL OUTPUT TABLE WITH 8 BYTES OF
                                        ;       11 NORMAL FORMAT DATA BASED ON DATA IN DDR STORAGE TABLE.  REG-
                                        ;       ISTERS D & E POINT TO STORAGE TABLE
470C                                    $
 (1)                                    ; *****************************************************************************

                                F11N:   MVI     B,8             ;SETUP COUNTER
470C 06 08              7.0             CALL    FILTBL          ;FILL TABLE
470E CD 48 47          18.0             JMP     GCDATX          ;EXIT
4711 C3 EC 46          10.0

4714                                    $
 (1)                                    ; *****************************************************************************
                                        ;F15N -- ROUTINE TO FILL 'EXPECTED' DATA TABLE WITH 15 NORMAL DATA BYTES
                                        ;       BASED ON DATA IN DDR STORAGE TABLE.
4714                                    $
 (1)                                    ; *****************************************************************************

4714 06 08              7.0     F15N:   MVI     B,8             ;SET COUNTER TO 8 BYTES
4716 3A F2 45          13.0             LDA     SDDRAC          ;GET DDR REG A DATA - BITS 7-0
4719 32 F0 45          13.0             STA     SDDRBT          ;SAVE IN STORAGE DDR B - REVERSE A + B
471C 3A F3 45          13.0             LDA     SDDRBC          ;GET REAL DDR B
471F 32 EF 45          13.0             STA     SDDRAT          ;SAVE IN DDR A STORAGE
4722 CD 48 47          18.0             CALL    FILTBL          ;FILL COMPARE TALBE
4725 3A F2 45          13.0             LDA     SDDRAC          ;RESET DDR STORAGE FOR
4728 32 EF 45          13.0             STA     SDDRAT          ;REGS A + B
472B 3A F3 45          13.0             LDA     SDDRBC
472E 32 F0 45          13.0             STA     SDDRBT
4731 C3 EC 46          10.0             JMP     GCDATX          ;EXIT

4734                                    $
 (1)                                    ; *****************************************************************************
                                        ;FIMAG -- GENERATE COMPARE TABLE FOR BAL IMAGE MODE
4734                                    $
 (1)                                    ; *****************************************************************************

4734 06 04              7.0     FIMAG:  MVI     B,4             ;SET BYTE COUNT TO
4736 3A F2 45          13.0             LDA     SDDRAC          ;GET DDR REG A DATA BYTE
4739 32 F0 45          13.0             STA     SDDRBT          ;SAVE IN REG B STORAGE (ONLY NEG 8 BITS IN IMAGE)
473C CD 48 47          18.0             CALL    FILTBL          ;FILL COMPARE TABLE WITH EXPECTED IMAGE DATA
473F 3A F3 45          13.0             LDA     SDDRBC          ;RESET DDR REG 'B' DATA BYTE
4742 32 F0 45          13.0             STA     SDDRBT
4745 C3 EC 46          10.0             JMP     GCDATX          ;EXIT

4748                                    $
 (1)                                    ; *****************************************************************************
                                        ;FILTBL -- ROUTINE COMMON TO 11/15 NORMAL, AND IMAGE MODE TO FILL THE
                                        ;       'EXPECTED' BAL OUTPUT BYTE DATA.  STORAGE BUFFER POINTED TO BY
                                        ;       D + E.
4748                                    $
 (1)                                    ; *****************************************************************************

4748 78                 4.0     FILTBL: MOV     A,B             ;PUT THE COUNT IN A
4749 32 DF 45          13.0             STA     DCNTWD          ;SAVE IT
474C 3A EF 45          13.0     FILTBX: LDA     SDDRAT          ;GET DDR REG 'A' - BITS 7-0 DATA
```

```
2245  474F  12                7.0          STAX    D               ;SAVE IN TABLE
2246  4750  13                6.0          INX     D               ;POINT TO NEXT TABLE LOCATION
2247  4751  05                4.0          DCR     B               ;DECREMENT BYTE COUNT
2248  4752  3A  F0  45       13.0          LDA     SDDRBT          ;GET DDR REG 'B' - BITS 15-8 DATA
2249  4755  12                7.0          STAX    D               ;SAVE IN TABLE
2250  4756  13                6.0          INX     D               ;POINT TO NEXT TABLE LOCATION
2251  4757  05                4.0          DCR     B               ;DECREMENT BYTE COUNT
2252  4758  C2  4C  47       10.0          JNZ     FILTBX          ;DONE? JUMP BACK IF MORE TO DO.
2253  475B  AF                4.0          XRA     A
2254  475C  32  E6  45       13.0          STA     LFTTST          ;CLEAR CHECK FLAG FOR 'WMC LEFT' TEST
2255  475F  32  E7  45       13.0          STA     LFTFLG
2256  4762  C9                10.0          RET                     ;YES - EXIT.
2257  4763                                 $
 (1)                                       ; *****************************************************************************
2258                                       ;LDBCNT -- ROUTINE TO LOAD THE BYTE COUNTER WITH NUMBER OF EXPECTED
2259                                       ;       BYTES BASED ON 60 SCLKS OF DATA AND THE SELECTED FORMAT CODE.
2260  4763                                 $
 (1)                                       ; *****************************************************************************
2261
2262  4763  0E  10              7.0  LDBCNT: MVI     C,DTEND-CDATBL/3        ;GET AMOUNT OF DATA PATTERNS
2263  4765  47                4.0          MOV     B,A             ;PUT THE COUNT IN REG B
2264  4766  AF                4.0          XRA     A               ;CLEAR ACCUMULATOR
2265  4767  80                4.0  LDBC1:  ADD     B               ;ADD IN BYTE COUNT (MULTIPLY)
2266  4768  0D                4.0          DCR     C               ;DOWNCOUNT
2267  4769  C2  67  47       10.0          JNZ     LDBC1           ;JUMP IF LOOP NOT DONE
2268  476C  32  E0  45       13.0          STA     IMBYTC          ;AND SAVE IN IMAGE STORAGE FRO BITS 7-0
2269  476F  3C                4.0          INR     A               ;FIX BYTE COUNTER FOR INTEL 8253 CHIP
2270  4770  D3  D4            10.0          OUT     BYTCNT          ;SAVE IN BYTE COUNTER
2271  4772  AF                4.0          XRA     A               ;CLEAR BITS 15-8 OF COUNTER
2272  4773  D3  D4            10.0          OUT     BYTCNT          ;LOAD BITS 15-8
2273  4775  C9                10.0          RET                     ;EXIT
2274
2275  4776                                 $
 (1)                                       ; *****************************************************************************
2276                                       ;F10D -- ROUTINE TO CALCULATE THE 10 BYTES EXPECTED FROM BAL WHEN IN 10
2277                                       ;       DUMP MODE AND HAVE 4 SCLKS (2 36 BIT WORDS OR 4 18 BIT WORDS).
2278  4776                                 $
 (1)                                       ; *****************************************************************************
2279
2280  4776  3E  0A              7.0  F10D:   MVI     A,10            ;FILL 10 BYTES OF DATA IN TABLE
2281  4778  32  DF  45       13.0          STA     DCNTWD          ;SAVE THE COUNT
2282  477B  CD  95  47       18.0  F10DA:  CALL    BYTATD          ;GET BYTES 'A' TO 'D'
2283  477E  CD  EF  47       18.0          CALL    F10DE           ;GET BYTE 'E'
2284  4781  C2  7B  47       10.0          JNZ     F10DA           ;LOOP BACK IF MORE TO DO
2285  4784  C3  EC  46       10.0          JMP     GCDATX          ;THEN EXIT
2286
2287  4787                                 $
 (1)                                       ; *****************************************************************************
2288                                       ;F10C -- ROUTINE TO CALCULATE 10 COMPAT. MODE BYTES 'A' TO 'D'
2289  4787                                 $
 (1)                                       ; *****************************************************************************
2290
2291  4787  3E  08              7.0  F10C:   MVI     A,8
2292  4789  32  DF  45       13.0          STA     DCNTWD          ;SAVE THE BYTE COUNT/4 SCLK
```

```
2293  478C  CD  95  47   18.0  F10CA:  CALL   BYTATD          ;GET BYTES 'A' TO 'D'
2294  478F  C2  8C  47   10.0          JNZ    F10CA
2295  4792  C3  EC  46   10.0          JMP    GCDATX          ;THEN EXIT
2296                               ;CALC. 10 MODE BYTE 'A' - BITS 0 TO 7
2297
2298  4795  47                4.0  BYTATD: MOV    B,A             ;COPY THE COUNT
2299  4796  3A  F0  45  13.0          LDA    SDDRBT          ;GET DDR REG 'B' DATA
2300  4799  CD  FE  46  18.0          CALL   SHR2            ;SHIFT ACCUMULATOR RIGHT 2 PLACES
2301  479C  4F                4.0          MOV    C,A             ;SAVE RESULT
2302  479D  3A  F1  45  13.0          LDA    SDDRCT          ;GET BIT 17 + 16 DATA BITS
2303  47A0  E6  03             7.0          ANI    3
2304  47A2  0F                4.0          RRC
2305  47A3  0F                4.0          RRC                    ;PUT 16 + 17 DATA IN BIT 6 + 7 POSITION
2306  47A4  B1                4.0          ORA    C               ;COMBINE FOR BYTE 'A'
2307  47A5  12                7.0          STAX   D               ;SAVE IN TABLE
2308  47A6  13                6.0          INX    D               ;POINT TO NEXT ENTRY IN TABLE
2309  47A7  05                4.0          DCR    B               ;DECREMENT BYTE COUNT
2310
2311                               ;CALC. 10 MODE BYTE 'B' - BITS 8 TO 15
2312
2313  47A8  3A  F0  45  13.0  F10MB:  LDA    SDDRBT
2314  47AB  E6  03             7.0          ANI    3               ;SAVE BYTS ''17'' + ''16''
2315  47AD  0F                4.0          RRC
2316  47AE  0F                4.0          RRC                    ;POSITION
2317  47AF  4F                4.0          MOV    C,A             ;SAVE FOR LATER
2318  47B0  3A  EF  45  13.0          LDA    SDDRAT          ;GET DDR REG 'A' DATA
2319  47B3  CD  FE  46  18.0          CALL   SHR2            ;SHIFT RIGHT 2 PLACES
2320  47B6  B1                4.0          ORA    C               ;COMBINE TO MAKE BYTE 'B'
2321  47B7  12                7.0          STAX   D               ;SAVE BYTE 'B' IN TABLE
2322  47B8  13                6.0          INX    D               ;UPDATE TABLE POINTER
2323  47B9  05                4.0          DCR    B               ;DECREMENT BYTE COUNT
2324
2325                               ;CALC. 10 MODE BYTE 'C' - BITS 16 TO 23
2326
2327  47BA  3A  EF  45  13.0  F10MC:  LDA    SDDRAT          ;GET DDR REG 'A' BYTE
2328  47BD  E6  03             7.0          ANI    3               ;SAVE BITS ''16'' + ''17''
2329  47BF  0F                4.0          RRC
2330  47C0  0F                4.0          RRC
2331  47C1  4F                4.0          MOV    C,A             ;SAVE FOR LATER
2332  47C2  3A  F1  45  13.0          LDA    SDDRCT          ;GET DDR REG 'C' BYTE
2333  47C5  CD  03  47  18.0          CALL   SHL4            ;SHIFT LEFT 4 PLACES TO 5 + 4
2334  47C8  B1                4.0          ORA    C               ;COMBINE FOR BITS 16-19
2335  47C9  4F                4.0          MOV    C,A             ;SAVE FOR LATER
```

```
2337  47CA  3A  F0  45   13.0          LDA   SDDRBT          ;GET DDR REG 'B' BYTE
2338  47CD  CD  FA  46   18.0          CALL  SHR4            ;RIGHT JUSTIFY 4 PLACES
2339                                                        ;FOR BITS "20" - "23"
2340  47D0  B1                4.0      ORA   C               ;COMBINE TO MAKE BYTE 'C'
2341  47D1  12                7.0      STAX  D               ;SAVE 10 DUMP BYTE 'C' IN TABLE
2342  47D2  13                6.0      INX   D               ;POINT TO NEXT TABLE ENTRY
2343  47D3  05                4.0      DCR   B               ;DECREMENT BYTE COUNT
2344                                ;CALC. 10 MODE BYTE 'D' - BITS 24 TO 31
2345
2346  47D4  3A  F0  45   13.0  F10MD: LDA   SDDRBT          ;GET DDR REG 'B' DATA
2347  47D7  CD  03  47   18.0          CALL  SHL4            ;LEFT JUSTIFY FOR BITS 24 - 27
2348  47DA  4F                4.0      MOV   C,A             ;SAVE FOR LATER
2349  47DB  3A  EF  45   13.0          LDA   SDDRAT          ;GET DDR REG 'A' DATA
2350  47DE  CD  FA  46   18.0          CALL  SHR4            ;RIGHT JUSTIFY REMAINING
2351                                                        ;4 HIGH BITS TO BIT 0 FOR
2352                                                        ;BITS "28" TO "31"
2353  47E1  B1                4.0      ORA   C               ;COMBINE TO MAKE BYTE 'D'
2354  47E2  12                7.0      STAX  D               ;SAVE IN TABLE
2355  47E3  13                6.0      INX   D               ;POINT TO NEXT TABLE ENTRY
2356  47E4  3E  FF            7.0      MVI   A,@377
```

```
2358  47E6  32  E6  45   13.0          STA     LFTTST          ;SET TO TEST 'LEFT'
2359  47E9  32  E7  45   13.0          STA     LFTFLG          ;'WMC LEFT' SET
2360  47EC  05               4.0      DCR     B               ;DECREMENT COUNT OF BYTES/SCLK
2361  47ED  78               4.0      MOV     A,B             ;AND PUT COUNT IN REG A
2362  47EE  C9              10.0      RET
2363
2364                                ;CALC. REMAINING 10 DUMP BYTE 'E' - BITS 32 TO 35
2365
2366  47EF  3A  EF  45   13.0  F10DE:  LDA     SDDRAT          ;GET DDR REG 'A' DATA
2367  47F2  E6  0F           7.0      ANI     $F              ;SAVE 4 BITS
2368  47F4  12               7.0      STAX    D               ;SAVE 10 DUMP BYTE 'E'
2369  47F5  13               6.0      INX     D               ;POINT TO NEXT TABLE ENTRY
2370  47F6  05               4.0      DCR     B               ;DECREMENT BYTE COUNT
2371  47F7  78               4.0      MOV     A,B             ;COPY TO REG A
2372  47F8  C9              10.0      RET
2373  47F9                           S
 (1)                               ; ************************************************************************
2374                               ;F10HD -- ROUTINE TO CALCULATE THE 'EXPECTED' DATA BYTE OUTPUT FROM BAL
2375                               ;       WHEN IN 10 HIGH DENSITY MODE.  4 SCLKS GENERATE 9 BYTES OF OUT-
2376                               ;       PUT.  D & E POINT TO STORAGE TABLE.
2377  47F9                           S
 (1)                               ; ************************************************************************
2378
```

```
2380                                        ;CALC. 10 HD BYTE 'A' - BITS 0 TO 7
2381
2382  47F9  3E  09        7.0  F10HD:  MVI   A,9            ;GET BYTE COUNT/4 SCLKS
2383  47FB  32  DF  45     13.0         STA   DCNTWD         ;SAVE IT
2384  47FE  CD  95  47     18.0         CALL  BYTATD         ;GET BYTES 'A' TO 'D'
2385
2386                                        ;CALC. 10 HD BYTE 'E' - BITS 32 TO BITS 3 OF NEXT WORD
2387
2388  4801  3A  EF  45     13.0  F10HDE: LDA   SDDRAT        ;GET DDR 'A'
2389  4804  CD  03  47     18.0         CALL  SHL4
2390  4807  4F              4.0         MOV   C,A            ;SAVE
2391  4808  3A  F1  45     13.0         LDA   SDDRCT         ;GET DDR 'C'
2392  480B  CD  07  47     18.0         CALL  SHL2
2393  480E  B1              4.0         ORA   C              ;COMBINE FOR BITS 7 TO 2
2394  480F  4F              4.0         MOV   C,A            ;SAVE
2395  4810  3A  F0  45     13.0         LDA   SDDRBT         ;GET DDR 'B'
2396  4813  E6  C0          7.0         ANI   @300           ;SAVE 2 HIGH BITS
2397  4815  07              4.0         RLC                  ;POSITION TO BITS
2398  4816  07              4.0         RLC                  ;1 + 0
2399  4817  B1              4.0         ORA   C              ;COMBINE TO MAKE BYTE 'E'
2400  4818  12              7.0         STAX  D              ;SAVE 10 HD BYTE 'E' IN TABLE
```

```
2402  4819  13              6.0          INX     D               ;POINT TO NEXT TABLE ENTRY
2403
2404                                 ;CALC. 10 HD BYTE 'F' - BITS 4 TO 11 OR WORD 2
2405
2406  481A  3A  F0  45       13.0  F10HDF: LDA     SDDRBT          ;GET DDR 'B'
2407  481D  CD  07  47       18.0          CALL    SHL2
2408  4820  4F               4.0          MOV     C,A             ;SAVE
2409  4821  3A  EF  45       13.0          LDA     SDDRAT          ;GET DDR 'A'
2410  4824  E6  C0           7.0          ANI     @300            ;SAVE 2 BITS
2411  4826  07               4.0          RLC                     ;POSITION IN THE LEFT
2412  4827  07               4.0          RLC
```

```
2414  4828  B1                  4.0          ORA     C               ;COMBINE TO MAKE BYTE 'F'
2415  4829  12                  7.0          STAX    D               ;SAVE 10 HD BYTE 'F'
2416  482A  13                  6.0          INX     D               ;POINT TO NEXT TABLE ENTRY
2417
2418                                   ;CALC. 10 HD BYTE 'G' - BITS 12 TO 19 OF 2ND WORD
2419
2420  482B  3A  EF  45           13.0  F10HDG: LDA     SDDRAT          ;GET DDR 'A'
2421  482E  CD  07  47           18.0          CALL    SHL2
2422  4831  4F                   4.0          MOV     C,A             ;SAVE
2423  4832  3A  F1  45           13.0          LDA     SDDRCT          ;GET DDR 'C'.
2424  4835  E6  03                7.0          ANI     3               ;SAVE 2 BITS
2425  4837  B1                   4.0          ORA     C               ;COMBINE TO MAKE BYTE 'G'
2426  4838  12                   7.0          STAX    D               ;SAVE IN TABLE
2427  4839  13                   6.0          INX     D               ;POINT TO NEXT TABLE ENTRY
2428                                   ;CALC. 10 HD BYTE 'H' - BITS 20 - 27
2429
2430  483A  3A  F0  45           13.0  F10HDH: LDA     SDDRBT          ;GET DDR 'B'
2431  483D  12                   7.0          STAX    D               ;STORE IN TABLE AS BYTE 'H'
2432  483E  13                   6.0          INX     D               ;POINT TO NEXT BYTE ENTRY
2433
2434                                   ;GET + SAVE LAST BYTE OF 10 HD MODE - BITS 28 - 35 OF 2ND WORD
2435
2436  483F  3A  EF  45           13.0  F10HDI: LDA     SDDRAT          ;GET DDR 'A'
2437  4842  12                   7.0          STAX    D               ;STORE IN TABLE
2438  4843  C3  EC  46           10.0          JMP     GCDATX          ;GO BACK TO COMMON EXIT
2439
2440  4846                                   S
 (1)                               ; *****************************************************************************
2441                                   ;CLKSYS -- ROUTINE TO CLOCK THE SYSTEM 1 TIME BY WRITING CLOCK CONTROL
2442                                   ;       'CLK' BIT TO A '1' THEN A '0'.
2443  4846                                   S
 (1)                               ; *****************************************************************************
2444
2445  4846  3A  DE  45           13.0  CLKSYS: LDA     CCTLWD          ;GET SOFTWARE CLOCK CONTROL IMAGE
2446  4849  F6  40                7.0          ORI     SSCLK           ;ADD IN 'CLK' BIT
2447  484B  D3  F0               10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL
2448  484D  E6  3F                7.0          ANI     @77             ;STRIP OFF CLOCK BIT
2449  484F  D3  F0               10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL WORD
2450  4851  C9                   10.0          RET                     ;EXIT - CLOCK CYCLE COMPLETE
2451
2452        0000                           .END
```

```
A     =%0007    ABCNT   45E1      ABCNTR  45E2      ADATA = 0094
AMTIEP= 0001    AMTIE7= 0002      ARAIDF= 0098      ASAVE   4F9B
ATTCD   4F97    AXNUM   4F91      B     =%0000      BADST = 0090
BALDAT  46BB    BIT0  = 0001      BIT1  = 0002      BIT15 = 8000
BIT2  = 0004    BIT3  = 0008      BIT4  = 0010      BIT5  = 0020
BIT6  = 0040    BIT7  = 0080      BIT8  = 0100      BIT9  = 0200
BRKPBC= 4F0A    BRKRAM= 4F10      BRKSTR- 4E60      BRKXCT= 4F00
BSAVE   4F9C    BYTATD  4795      BYTCNT- 00D4      BYTEH   4F24
BYTEL   4F23    C     =%0001      CASCT   4F21      CASCTL= 00A0
CASSTA= 00A0    CATTH = 0089      CATTL = 0088      CBUSST= 00A1
CBYTH = 008B    CBYTL = 008A      CCTLWD  45DE      CDATBL  4688
CDG1H = 0087    CDG1L = 0086      CDG2H = 0093      CDG2L = 0092
CDG3H = 0095    CDG3L = 0094      CDVTH = 008D      CDVTL = 008C
CHPTIE= 0028    CHOTIE= 0020      CH1TIE= 0021      CH2TIE= 0022
CH3TIE= 0023    CH4TIE= 0024      CH5TIE= 0025      CH6TIE= 0026
CH7TIE= 0027    CKLOP = 2815      CKSCLK  441A      CKSC1   4429
CLKCNT  45EA    CLKCTL= 00F0      CLKON   45E8      CLKSYS  4846
CLOCK   4F26    CMC0H = 0099      CMC0L = 0098      CMC1H = 009B
CMC1L = 009A    CMC2H = 009D      CMC2L = 009C      CMC3H = 009F
CMC3L = 009E    CMINH = 0097      CMINL = 0096      CNTCTL= 00D7
CRCWRD= 0018    CSAVE   4F9D      CSRLH = 0091      CSRLL = 0090
CTCH  = 0085    CTCL  = 0084      CTSTH = 008F      CTSTL = 008E
CXCTH = 0081    CXCTL = 0080      CXINH = 0083      CXINL = 0082
C.    = 0001    C.AVAI= 0080      C.DP  = 0008      C.DSE = 0010
C.DTU = 0003    C.DVA = 0008      C.FAIL= 00FC      C.FMT = 007C
C.FNCT= 003E    C.GO  = 0001      C.INTC= 00FE      C.MAIN= 0020
C.NSA = 0080    C.RCT = 00FC      C.SER = 0080      C.SHR = 0040
C.SKPC= 000F    C.TAPE= 0040      C.WCS = 0002      D     =%0002
DATACT= 00D0    DATPE   4502      DATPEC  450D  G   DATPEE  4513
DATPO   44EC    DATPOC  44FA  G   DATPOE  4500      DBUS    4F28
DBUSCT= 00C0    DBUSST= 00C0      DCNTWD  45DF      DDRA  = 00D8
DDRAIN= 0010    DDRB  = 00D9      DDRBIN= 0002      DDRC  = 00DA
DDRCIN= 0001    DDRCO = 0088      DDRCTL= 00DB      DELOOP  456C
DELP1   4579    DIAFLG  4F22      DIAGPG= 4300      DIAGRM= 4F90
DIARD = 000B    DONE1 = 0045      DONINT= 0010      DPNUM   45E5
DSAVE   4F9E    DSE   = 0006      DTEND   46B8      D.ATH0= 0001
D.ATH1= 0002    D.EOTD= 0010      D.LAGC= 0020      D.NOTW= 0040
D.NTHR= 0004    D.TACH= 0008      D.WR4 = 0080      E     =%0003
EABCNT  45E3    ECCBAD= 0042      ECCCHK  4587      ECCCNT  45E9
ECCCOR= 0019    ECCOK = 0041      ECCON   459B  G   ECCON1  45A1
ECCSTA= 001A    ECCTST= 000E      EDATA = 0095      EOTCLR= 0003
ERFLG   4F93    ERLP  = 2809      ERLPA = 280F      ERLPB = 2812
ERLPE = 280C    ERNUM   4F90      ERRCNT= 00D6      ESAVE   4F9F
E.ACRC= 0010    E.AMT = 0020      E.CDP = 0080      E.CRC = 0080
E.PNTR= 0008    E.RPE = 0040      E.STEC= 0001      E.TTEC= 0002
E.UNC = 0004    FIFORD= 006A      FILTBL  4748      FILTBX  474C
FIMAG   4734    FMTTBL  46C5      FORMAT  4F25      FORMT   45E4
FWDTST= 0061    F10C    4787      F10CA   478C      F10D    4776
F10DA   477B    F10DE   47EF      F10HD   47F9      F10HDE  4801
F10HDF  481A    F10HDG  482B      F10HDH  483A      F10HDI  483F
F10MB   47A8    F10MC   47BA      F10MD   47D4      F11N    470C
F15N    4714    GCDAT   46D3      GCDATX  46EC      GCRID = 0089
GCRSET= 0002    GENDAT  45F8      GENOUT  4623      GOODTM= 0092
H     =%0004    HLSAVE  4FA0      IE    = 0008      IMBYTC  45E0
```

```
INTSTA= 00E0          ITERA   4F9A          I.PWR = 0020          I.RMPE= 0040
I5.5  = 0010          I6.5  = 0020          I7.5  = 0040          KCALL = 005F
KCLR  = 007B          KDEP  = 003F          KENAB = 0078          KEXAM = 003E
KEYBRD= 00C8          KEY1  = 0078          KEY10 = 006D          KEY11 = 006E
KEY12 = 006F          KEY13 = 005C          KEY14 = 005D          KEY15 = 005E
KEY16 = 005F          KEY17 = 003C          KEY18 = 003D          KEY19 = 003E
KEY2  = 0079          KEY20 = 003F          KEY3  = 007A          KEY4  = 007B
KEY5  = 0074          KEY6  = 0075          KEY7  = 0076          KEY8  = 0077
KEY9  = 006C          KINTA = 006F          KLDAD = 003D          KN0   = 003C
KN1   = 005C          KN2   = 005D          KN3   = 005E          KN4   = 006C
KN5   = 006D          KN6   = 006E          KN7   = 0074          KN8   = 0075
KN9   = 0076          KU2   = 0079          KU3   = 007A          KU8   = 0077
L     =%0005          LBLANK= 000F          LCE   = 000B          LCH   = 000C
LCL   = 000D          LCP   = 000E          LC0   = 0000          LC1   = 0001
LC2   = 0002          LC3   = 0003          LC4   = 0004          LC5   = 0005
LC6   = 0006          LC7   = 0007          LC8   = 0008          LC9   = 0009
LDBCNT  4763          LDBC1   4767          LDLEDA= 00CA          LDLEDB= 00CB
LDLEDC= 00CC          LDLEDD= 00CD          LDLEDE= 00CE          LDLEDF= 00CF
LFTCHK  45A4          LFTCMP  45C3          LFTFLG  45E7          LFTOFF  45CB
LFTOK1  45D7  G       LFTOK2  45BC  G       LFTON   45B0          LFTTST  45E6
LKDIAG= 2800          LKKBD = 004C          LKKEY = 0049          LKLWMG= 0058
LKLWMP= 0055          LKLWPG= 0052          LKLWPP= 004F          LKMOD7= 0046
LKOPR = 0046          LPFLG   4F94          LPNUM   4F92          M     =%0006
MBSEL = 00E0          MB.A  = 0008          MB.B  = 0004          MEMTOP= 4FFF
MINUS = 000A          MM    = 8000          MSE   = 0008          MSGN  = 000D
MTACLR= 0000          MT.ARA= 0020          MT.CPE= 0080          MT.DSE= 0001
MT.FWD= 0040          MT.INH= 0008          MT.LWR= 0004          MT.MOT= 0002
MT.NWT= 0080          MT.PEC= 0040          MT.PSB= 0004          MT.PS0= 0001
MT.PS1= 0002          MT.REV= 0020          MT.WRT= 0010          MT.Z  = 0008
M.ATA = 0080          M.CAPE= 0020          M.CONT= 0080          M.DEM = 0020
M.EBL = 0010          M.EXC = 0008          M.FAIL= 0008          M.ILR = 0010
M.INIT= 0010          M.OCC = 0020          M.ONLI= 0001          M.PE  = 0040
M.PORT= 0080          M.RDEN= 0002          M.RDPE= 0008          M.RUN = 0004
M.SCLK= 0001          M.TRA = 0040          M.UNIT= 0007          M.WCLK= 0040
M.WCLN= 0080          M.WREN= 0080          M5.5  = 0001          M6.5  = 0002
M7.5  = 0004          NOTCAP= 0088          OKAY  = 00FF          OPRRAM= 4300
OPSTRT= 0058          OPVER = 0040          PADCNT= 00D5          PADCRC= 0080
PDIAG = 0048          PEID  = 008A          PENAB = 004C          PESET = 0001
PL    = 00B1          PRDD  = 004C          PRENF = 009C          PS    = 00B2
PSTAT = 0048          PSW   =%0009          P.AMTP= 0001          P.BCTC= 0040
P.CMDP= 0020          P.INTE= 0080          P.LCS = 0040          P.LWR = 0020
P.RDP = 0002          P.RPEN= 0004          P.RPST= 0002          P.RPOE= 0020
P.RP1E= 0010          P.RP2E= 0020          P.RP3E= 0010          P.SING= 0080
P.STAT= 0002          P.STPE= 0080          P.TACH= 0008          P.TUPR= 0010
P.WCSP= 0004          P.WDS = 0040          P.WFLP= 0001          P.WPEN= 0010
P.WPOE= 0008          P.WP1E= 0004          P.WP2E= 0008          P.WP3E= 0004
P.5VOK= 0002          QUE   = 281B          QUEM  = 281E          RAMT  = 0010
RARA  = 0006          RARAI = 0004          RCHBD0= 0048          RCHBD1= 0047
RCHOK = 0046          RCHTST= 000C          RCLRT = 000D          RCMD  = 000B
RCMLP = 0003          RCONT = 0080          RDATA = 0017          RDCLK = 0010
RDON  = 0011          READG = 0007          REND  = 0014          REQST = 2806
RESCHR= 00D1          REVTST= 0064          REWIND= 0004          RFIFOL= 0008
RGCLK = 0002          RGCRI = 0003          RIBG  = 0001          RILL  = 0012
RINST = 000C          RMCTST= 0008          RMK2  = 0013          RNOP  = 0000
```

```
RPATH = 0001        RPBAD = 0044        RPCHI = 0001        RPCLK = 0003
RPCTL = 0009        RPEI  = 0002        RPFAIL= 0000        RPF1  = 009D
RPF2  = 009E        RPOK  = 0043        RPOSTN= 0016        RPSTA = 0015
RRCMT = 000A        RSTAT = 0002        RTIEB = 000A        RTIER = 0030
RTM   = 0005        RUNKI = 0009        RUPTST= 005E        RWDUNL= 0005
R.AMT = 0001        R.BOP = 0008        R.DATA= 0040        R.DON = 0002
R.DRDY= 0010        R.END = 0010        R.ILL = 0004        R.JVOK= 0004
R.MK2 = 0008        R.PLOD= 0008        R.PL00= 0010        R.PL01= 0020
R.POST= 0020        R.STNM= 0002        R.STOP= 0004        R.STPC= 0001
R00L  = 0080        R01H  = 0083        R01L  = 0082        R02H  = 0085
R02L  = 0084        R03H  = 0087        R03L  = 0086        R04H  = 0089
R04L  = 0088        R05H  = 008B        R05L  = 008A        R06H  = 008D
R06L  = 008C        R07H  = 008F        R07L  = 008E        R10H  = 0091
R10L  = 0090        R11H  = 0093        R11L  = 0092        R12H  = 0095
R12L  = 0094        R13H  = 0097        R13L  = 0096        R14H  = 0099
R14L  = 0098        R15H  = 009B        R15L  = 009A        R16H  = 009D
R16L  = 009C        R17H  = 009F        R17L  = 009E        R7.5  = 0010
SCCNTR  45EB        SCKB1   4402        SCKGUN  43EF        SCKWT   43C9
SCKWTC  43E6  G     SCKWTE  4419        SCKWT1  43D6        SCLKBI  43BE
SDDCOM  4678        SDDRA   45EC        SDDRAC  45F2        SDDRAI  45EF
SDDRB   45ED        SDDRBC  45F3        SDDRBT  45F0        SDDRC   45EE
SDDRCC  45F4        SDDRCT  45F1        SELCLR= 0000        SETATA= 00A1
SETDDR  4627        SHL2    4707        SHL4    4703        SHR2    46FE
SHR4    46FA        SID   = 0080        SINTST  45F7        SLFTDD  4666
SOD   = 0080        SOE   = 0040        SP    =%0008        SSCLK = 0040
SSTEP = 0005        STACK = 4FFF        STATRM= 4F20        STPCT   4F20
STRSP = 5000        SWMCST  45F5        SXFER   45F6        TADR00= 0080
TADR01= 0081        TADR02= 0082        TADR03= 0083        TADR04= 0084
TADR05= 0085        TADR06= 0086        TADR07= 0087        TADR10= 0088
TADR11= 0089        TADR12= 008A        TADR13= 008B        TAMT  = 0044
TASEL = 0080        TCMD  = 0040        TC.FWD= 0040        TC.INH= 0008
TC.LWR= 0004        TC.REV= 0020        TC.WRT= 0010        TEMP    4F99
TMF   = 0099        TMRDY = 0040        TRKENA= 00D2        TSET  = 2803
TSTEND= 2818        TSTS  = 0040        TST1    4300        TST1A   4371
TST1B   437C        TST1BC  43A3  G     TST1C   439A        TST1D   43AC
TST1DC  43B5  G     TST1L   430C        TST1ST  4305        TST1XC  44D4  G
TUSEL0= 00D1        TUSEL1= 00D2        TU78  = 0010        T.ATH0= 0001
T.ATH1= 0002        T.BOT = 0004        T.EOT = 0002        T.FPT = 0001
T.NTHR= 0004        T.ONL = 0020        T.PES = 0008        T.PSBJ= 0020
T.PSOJ= 0008        T.PS1J= 0010        T.RDY = 0080        T.RDYO= 0040
T.RWD = 0010        T.SCLK= 0002        T1RST   4364        UIBG  = 00A1
VALFC   4F98        VALTB   4F95        VELTST= 005B        WDR.P = 0010
WMCCTL= 00D3        WMCERR= 00DA        WMCSTA= 00D0        WRTCLK= 0000
WRTDAT= 00D3        W.ACRC= 0004        W.CRC = 0008        W.DIAG= 0002
W.DONN= 0040        W.ECC = 0010        W.ENAB= 0080        W.ERR = 0020
W.FMT = 0070        W.GCR = 0010        W.LEFT= 0004        W.ONES= 0020
W.RESI= 0002        W.REV = 0004        W.ROME= 0010        W.RST = 0001
W.SKIP= 000F        W.WRIT= 0008        W.XFER= 0020        X     =%000A
XFRBIT  442D        XFRB1   4435        XFRB2   4453        XFRB2C  4469  G
XFRB3   4472        XFRB3A  4479        XFRB3C  448F  G     XFRB4   4498
XFRB4P  44DA        XFRB5   4514        XFRB5C  4543  G     XFRGON  444B  G
X.DONN= 0080        X.ENAB= 0040        X.PEPE= 0002        X.ROME= 0001
X.WCLK= 0001        Y     =%000B        .     = 4852
```

ERRORS DETECTED: 0

*WMC3.A78/PTP,WMC3=NLIST,PARAM,MACRO,LIST,WMC3
RUN-TIME: 4 6 0 SECONDS
CORE USED: 10K

3 "MEMORY MAP" EXTENDED ADDRESS BIT
19 READ MICRO CONTROLLER CONTROL & STATUS WORDS
139 TRANSPORT/PORT COMMAND + STATUS WORDS
380 COMMON ADDRESS SPACE REGISTERS
444 MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485 MASSBUS CONTROL WORD DEFINITIONS
557 WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709 KEYBOARD/DISPLAY CONTROL + STATUS WORD
797 MISCELLANEOUS VARIABLE DEFINITIONS
801 RAM ADDRESS DEFINITIONS
819 LINKAGE TABLE ADDRESSES
842 BIT DEFINITIONS
856 RIM AND SIM INSTRUCTION BIT DEFINITIONS
877 INTERRUPTABLE INSTRUCTION MACROS
924 DIAGNOSTIC VARIABLES
945 COMMON VARIABLE STORAGE
959 MACRO CALL DEFINITIONS
1047 TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101 8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1333 TEST 1 - WRITE MICRO CHARACTER GENERATION TEST IN GCR MODE
2734 CLEAR ACRC SUBROUTINE
2747 CALCULATE ACRC CHARACTER
2816 FORMAT FINAL ACRC CHARACTER
2829 PARITY CALCULATION ROUTINE
3024 CLEAR ECC SUBROUTINE
3038 CALCUALTE ECC CHARACTER
3079 POLYNOMIAL BIT TRANSLATION

```
961                     ;*********************************************************************
962                     ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
963                     ;ERROR MACRO CALLS
964                     ;
965                     ;       1  -    REQUEST HOST CPU TO PRINT:
966                     ;               "BYTE/SCLK COUNT NUMBER = LLL"
967                     ;               WHERE:
968                     ;               LLL = THE VALUE STORED IN CAS REGISTER 5
969                     ;                     (THE BYTE COUNT REGISTER 16 BITS).
970                     ;
971                     ;       2  -    REQUEST HOST CPU TO PRINT:
972                     ;               DATA FORMAT = MM
973                     ;               SKIP COUNT  = NN
974                     ;               WHERE:
975                     ;               MM = DATA FORMAT FROM CAS REGISTER 2
976                     ;               NN = SKIP COUNT FROM CAS REGISTER 2
977                     ;
978                     ;       3  -    REQUEST HOST CPU TO PRINT:
979                     ;               BYTE-SCLK COUNT = LLL
980                     ;               DATA FORMAT = MM
981                     ;               SKIP COUNT  = NN
982                     ;               WHERE:
983                     ;               LLL = AS ABOVE
984                     ;               MM  = AS ABOVE
985                     ;               NN  = AS ABOVE
986                     ;
987                     ;       4  -    REQUEST HOST TO PRINT:
988                     ;               TRANSITION COUNT = LLL
989                     ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
990                     ;
991                     ;       5  -    REQUEST HOST CPU TO PRINT:
992                     ;               EXPECTED 16 BITS =      E EEEEEE
993                     ;               ACTUAL 18 BITS =        A AAAAAA
994                     ;
995                     ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
996                     ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
997                     ;               OF CAS REG 15 LOW BYTE.
998                     ;
999                     ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
1000                    ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
1001                    ;               17 LOW BYTE SIGN BIT.
1002                    ;
1003                    ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
1004                    ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
1005                    ;               AND/OR ACTUAL DATA.
1006                    ;
1007                    ;       7  -    REQUEST HOST CPU TO PRINT:
1008                    ;               "SUBGROUP NUMBER = LLL"
1009                    ;               WHERE:
1010                    ;               LLL = THE VALUE STORED IN CAS REGISTER 5
1011                    ;                     (THE BYTE COUNT REGISTER 16 BITS).
1012                    ;
1013                    ;       10 -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
1014                    ;*********************************************************************
```

```
1016                            ; ************************************************************************
1017                            ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                            ;TO CAUSE SOME HOST CPU ACTION.
1019                            ;
1020                            ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                            ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                            ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                            ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                            ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                            ;               - HOST RESPONSE CODE 31 OR 33
1026                            ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                            ;               - HOST RESPONSE CODE 31 OR 33
1028                            ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                            ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                            ;                       BIT0 = 1 TEST PORT 0
1031                            ;                       BIT1 = 1 TEST PORT 1
1032                            ;                       BIT2 = 1 TEST PORT 2
1033                            ;                       BIT3 = 1 TEST PORT 3
1034                            ; ************************************************************************
1035                            ; ************************************************************************
1036                            ;DATA PATTERN CODES
1037                            ;
1038                            ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                            ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                            ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                            ;               18 BITS OF ALL 1'S
1042                            ;               18 BITS OF ALL 0'S
1043                            ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                            ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                            ; ************************************************************************
1329    4300                            .-      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  WMC4 - WMC ECC, ACRC, CRC, AND RESID CHAR TESTS
                                .SBTTL  TEST 1 - WRITE MICRO CHARACTER GENERATION TEST IN GCR MODE
                                ;ID     WMC4-WRITE MICRO CONTROLLER PART #4
4300                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*GCR CHARACTER TESTS
4300                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST IS DESIGNED TO TEST THE WRITE MICROCONTROLLER CHARACTER GENERATION
                                ;*LOGIC.  A KNOWN DATA PATTERN IS FED INTO THE BYTE ASSEMBLY LOGIC FROM
                                ;*THE DIAGNOSTIC DATA REGISTERS - "DDR" - TO PROVIDE AN 8 BIT WIDE
                                ;*REPLACEMENT FOR MASSBUS DATA IN IMAGE MODE.
                                ;*
                                ;*THE BYTE ASSEMBLY LOGIC WILL "BREAK UP" THE INPUT DATA (8 BITS)
                                ;*INTO THE DESIRED BYTE FORMAT TO BE WRITTEN ON TAPE.
                                ;*
                                ;*THIS TEST WILL CHECK THE OUTPUT OF THE WRITE MICRO TO BE CORRECT
                                ;*IN REGARD TO THE ECC CHARACTERS GENERATED AND THE PROPER LOCATION
                                ;*OF THESE CHARACTERS IN THE DATA STREAM.
4300                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  INIT THE TEST
                                ;*  SET FORMAT INDEX NUMBER FOR IMAGE MODE
                                ;*  BGNDO
                                ;*  :  INIT DATA PATTERN NUMBER FOR PASS
                                ;*  :  RESTART WMC & SET CLOCK CONTROL TO SINGLE STEP
                                ;*  :  INIT WMC BYTE COUNTER + ERROR CODE COUNTER AND PAD COUNTER CONTROL
                                ;*  :  CLEAR ERROR CODE COUNTER
                                ;*  :  CLEAR PROGRAM FLAGS AND STORAGE AREAS
                                ;*  :  SET DDR CONTROL TO 'OUT' SO DATA INJECTION POSSIBLE
                                ;*  :  GENERATE DATA TO DDR REGISTERS AND CHECK STORAGE
                                ;*  :  INIT SIMULATED 'ECC' GENERATED CHARACTER
                                ;*  :  INIT SIMULATED 'CRC' GENERATED CHARACTER
                                ;*  .  INIT SIMULATED 'ACRC' GENERATED CHARACTER
                                ;*  :  CLOCK WMC THROUGH ITS RESTART AREA-WMC ROM PROGRAM
                                ;*  :  CALL SUBROUTINE LKMOD7-CALCULATE 'MOD7' + 'MOD32'
                                ;*  :  LOAD RESCHR WITH MOD7 REMAINDER + MOD32
                                ;*  :  CALCULATE PROEPR NUMBER OF PAD CHARACTERS AND LOAD PAD COUNTER
                                ;*  :  LOAD DATACTL WITH FORMAT 5 (IMAGE) AND SKIP COUNT 0
                                ;*  :  SET WMCCTL TO 'WRITE' & 'ENABLE'-START WMC ROM PGM
                                ;*  :  BGNDO
                                ;*  :  :  CLOCK SYSTEM
                                ;*  :  :  IF WMCSTA HAS NO 'XFER' BIT SET
                                ;*  :  :  :  THEN-UPDATE TIMEOUT COUNTERS
                                ;*  :  :  :  :  IF TIMEOUT DETECTED
                                ;*  :  :  :  :  :  THEN-REPORT LACK OF ACTIVITY IN SPECIFIED TIME
                                ;*  :  :  :  :  :  ELSE-CONTINUE
```

```
1377                                    ;*  :  :  :  :  ENDIF
1378                                    ;*  :  :  :  ELSE-GET WMCSTA WORD + SAVE FOR LATER CHECKING
1379                                    ;*  :  :  :  :  BGNDO
1380                                    ;*  :  :  :  :  :  CLOCK SYSTEM TO FINISH 1ST NIBBLE OF DATA
1381                                    ;*  :  :  :  :  :  IF > 20 CLOCKS
1382                                    ;*  :  :  :  :  :  :  THEN-REPORT FAILURE TO CLEAR 'XFER' BIT
1383                                    ;*  :  :  :  :  :  :  ELSE-CONTINUE
1384                                    ;*  :  :  :  :  :  ENDIF
1385                                    ;*  :  :  :  :  :  DO UNTIL 'XFER' CLEARS OR TIMEOUT
1386                                    ;*  :  :  :  :  ENDDO
1387                                    ;*  :  :  :  :  BGNDO
1388                                    ;*  :  :  :  :  :  CLOCK SYSTEM TO BEGIN 2ND NIBBLE OF DATA
1389                                    ;*  :  :  :  :  :  IF > 128 CLOCKS
1390                                    ;*  :  :  :  :  :  :  THEN-REPORT FAILURE TO SET 'XFER' FOR 2ND NIBBLE
1391                                    ;*  :  :  :  :  :  :  ELSE-CONTINUE
1392                                    ;*  :  :  :  :  :  ENDIF
1393                                    ;*  :  :  :  :  :  DO UNTIL 'XFER' SETS OR TIMEOUT
1394                                    ;*  :  :  :  :  ENDDO
1395                                    ;*  :  :  :  :  GET PARITY BIT FROM INTSTA WORD FOR ASSEMBLED BYTE
1396                                    ;*  :  :  :  :  BGNDO
1397                                    ;*  :  :  :  :  :  CLOCK SYSTEM TO FINISH 2ND NIBBLE
1398                                    ;*  :  :  :  :  :  IF > 20 CLOCKS
1399                                    ;*  :  :  :  :  :  :  THEN-REPORT FAILURE TO CLEAR 'XFER' ON 2ND NIBBLE
1400                                    ;*  :  :  :  :  :  :  ELSE-CONTINUE
1401                                    ;*  :  :  :  :  :  ENDIF
1402                                    ;*  :  :  :  :  :  DO UNTIL 'XFER' IS CLEAR OR TIMEOUT
1403                                    ;*  :  :  :  :  ENDDO
1404                                    ;*  :  :  :  :  CLOCK SYSTEM TWICE MORE TO 'SET' INTERNAL STATUS
1405                                    ;*  :  :  :  :  IF WRTDAT BYTE IS AN 'ECC' CHARACTER
1406                                    ;*  :  :  :  :  :  THEN-GET ECC CHARACTER FROM WRTDAT
1407                                    ;*  :  :  :  :  :  :  IF ECC BYTE=SIMULATED BYTE
1408                                    ;*  :  :  :  :  :  :  :  THEN-GET ECC BYTE PARITY BIT
1409                                    ;*  :  :  :  :  :  :  :  :  IF PARITY ON ECC IS OK
1410                                    ;*  :  :  :  :  :  :  :  :  :  THEN-EXIT DO LOOP
1411                                    ;*  :  :  :  :  :  :  :  :  :  ELSE-REPORT BAD PARITY ON ECC BYTE
1412                                    ;*  :  :  :  :  :  :  :  :  ENDIF
1413                                    ;*  :  :  :  :  :  :  :  :  ELSE-REPORT 'ECC' GENERATION ERROR
1414                                    ;*  :  :  :  :  :  :  :  ENDIF
1415                                    ;*  :  :  :  :  :  :  ELSE-IS BYTE IN WRTDAT AN 'ACRC' CHARACTER?
1416                                    ;*  :  :  :  :  :  :  :  IF IS AN 'ACRC' BYTE
1417                                    ;*  :  :  :  :  :  :  :  :  THEN-CHECK ACTUAL 'ACRC' WITH SIMULATED 'ACRC'
1418                                    ;*  :  :  :  :  :  :  :  :  :  IF OK
1419                                    ;*  :  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1420                                    ;*  :  :  :  :  :  :  :  :  :  :  ELSE-REPORT 'ACRC' GENERATION ERROR
1421                                    ;*  :  :  :  :  :  :  :  :  :  ENDIF
1422                                    ;*  :  :  :  :  :  :  :  :  :  IF PARITY BIT ON 'ACRC' IS OK
1423                                    ;*  :  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1424                                    ;*  :  :  :  :  :  :  :  :  :  :  ELSE-REPORT BAD PARITY ON 'ACRC' BYTE
1425                                    ;*  :  :  :  :  :  :  :  :  :  ENDIF
1426                                    ;*  :  :  :  :  :  :  :  :  ELSE-CONTINUE
1427                                    ;*  :  :  :  :  :  :  :  ENDIF
1428                                    ;*  :  :  :  :  :  :  :  IF BYTE ASSEMBLED IS A 'CRC' BYTE
1429                                    ;*  :  :  :  :  :  :  :  :  THEN-CHECK ACTUAL 'CRC' WITH SIMULATED 'CRC'
1430                                    ;*  :  :  :  :  :  :  :  :  :  IF SAME
```

```
1431                    ;*  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1432                    ;*  :  :  :  :  :  :  :  :  :  ELSE-REPORT 'CRC' GENERATION ERROR
1433                    ;*  :  :  :  :  :  :  :  :  ENDIF
1434                    ;*  :  :  :  :  :  :  :  :  IF PARITY BIT ON 'CRC' IS OK
1435                    ;*  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1436                    ;*  :  :  :  :  :  :  :  :  :  ELSE-REPORT BAD PARITY ON 'CRC' BYTE
1437                    ;*  :  :  :  :  :  :  :  :  ENDIF
1438                    ;*  :  :  :  :  :  :  :  :  IF RESIDUAL BYTE DETECTED BEFORE 'CRC'
1439                    ;*  :  :  :  :  :  :  :  :  :  THEN-REPORT ERROR
1440                    ;*  :  :  :  :  :  :  :  :  :  ELSE-CONTINUE
1441                    ;*  :  :  :  :  :  :  :  :  ENDIF
1442                    ;*  :  :  :  :  :  :  :  :  IF 'ACRC' NOT DETECTED YET
1443                    ;*  :  :  :  :  :  :  :  :  :  THEN-REPORT ERROR
1444                    ;*  :  :  :  :  :  :  :  :  :  ELSE-CONTINUE
1445                    ;*  :  :  :  :  :  :  :  :  ENDIF
1446                    ;*  :  :  :  :  :  :  :  ELSE-CONTINUE
1447                    ;*  :  :  :  :  :  :  :  ENDIF
1448                    ;*  :  :  :  :  :  :  :  IF BYTE ASSEMBLED IS A RESIDUAL CHARACTER
1449                    ;*  :  :  :  :  :  :  :  :  THEN-CHECK TO SEE IF 'ACRC' SENT
1450                    ;*  :  :  :  :  :  :  :  :  :  IF NO 'ACRC' BYTE
1451                    ;*  :  :  :  :  :  :  :  :  :  :  THEN-REPORT ERROR
1452                    ;*  :  :  :  :  :  :  :  :  :  :  ELSE-CONTINUE
1453                    ;*  :  :  :  :  :  :  :  :  :  ENDIF
1454                    ;*  :  :  :  :  :  :  :  :  :  IF ALL EXPECTED 'CRC' BYTES DETECTED
1455                    ;*  :  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1456                    ;*  :  :  :  :  :  :  :  :  :  :  ELSE-REPORT ERROR
1457                    ;*  :  :  :  :  :  :  :  :  :  ENDIF
1458                    ;*  :  :  :  :  :  :  :  :  :  IF RESIDUAL BYTE SAME AS EXPECTED
1459                    ;*  :  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1460                    ;*  :  :  :  :  :  :  :  :  :  :  ELSE-REPORT ERROR
1461                    ;*  :  :  :  :  :  :  :  :  :  ENDIF
1462                    ;*  :  :  :  :  :  :  :  :  ELSE-CONTINUE
1463                    ;*  :  :  :  :  :  :  :  ENDIF
1464                    ;*  :  :  :  :  :  :  :  BYTE ASSEMBLED MUST BE A DATA BYTE FROM WMC
1465                    ;*  :  :  :  :  :  :  :  UPDATE BYTE COUNTER
1466                    ;*  :  :  :  :  :  :  :  IF THIS BYTE EXPECTED TO BE A 'PAD'
1467                    ;*  :  :  :  :  :  :  :  :  THEN-UPDATE # RECEIVED 'PADS'
1468                    ;*  :  :  :  :  :  :  :  :  :  IF PAD BYTE=0
1469                    ;*  :  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1470                    ;*  :  :  :  :  :  :  :  :  :  :  ELSE-REPORT ERROR
1471                    ;*  :  :  :  :  :  :  :  :  :  ENDIF
1472                    ;*  :  :  :  :  :  :  :  :  :  IF PARITY ON 'PAD' IS OK
1473                    ;*  :  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1474                    ;*  :  :  :  :  :  :  :  :  :  :  ELSE-REPORT ERROR
1475                    ;*  :  :  :  :  :  :  :  :  :  ENDIF
1476                    ;*  :  :  :  :  :  :  :  :  :  IF ALL EXPECTED PADS CHECKED
1477                    ;*  :  :  :  :  :  :  :  :  :  :  THEN-CHECK PAD COUNTER
1478                    ;*  :  :  :  :  :  :  :  :  :  :  :  IF PADCNT=0
1479                    ;*  :  :  :  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
1480                    ;*  :  :  :  :  :  :  :  :  :  :  :  :  ELSE-REPORT ERROR
1481                    ;*  :  :  :  :  :  :  :  :  :  :  :  ENDIF
1482                    ;*  :  :  :  :  :  :  :  :  :  :  ELSE-CONTINUE
1483                    ;*  :  :  :  :  :  :  :  :  :  ENDIF
1484                    ;*  :  :  :  :  :  :  :  :  :  IF BYTE EXPECTED TO BE A DATA BYTE
```

```
                                ;*  :  :  :  :  :  :  :  :  :  THEN-CHECK ACTUAL WITH EXPECTED DATA
                                ;*  :  :  :  :  :  :  :  :  :  :  IF SAME
                                ;*  :  :  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
                                ;*  :  :  :  :  :  :  :  .  :  :  :  ELSE-REPORT WMC ASSEMBLY ERROR
                                ;*  :  :  :  :  :  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  :  :  :  :  :  :  IF PARITY BIT ON DATA ASSEMBLED IS OK
                                ;*  :  :  :  :  :  :  :  :  :  :  :  THEN-CONTINUE
                                ;*  :  :  :  :  :  :  :  :  :  :  :  ELSE-REPORT BAD PARITY DETECTED
                                ;*  :  :  :  :  :  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  :  :  :  :  :  ELSE-CONTINUE
                                ;*  :  :  :  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  :  :  :  REPORT 'XFER' SET WITH NO APPARANT REASON
                                ;*  :  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  ENDIF
                                ;*  :  :  :  ENDIF
                                ;*  :  :  ENDIF
                                ;*  :  :  DO UNTIL ALL BYTES ASSEMBLED AND CHECKED FOR CURRENT PACKING MODE
                                ;*  :  ENDDO
                                ;*  :  CHECK BYTE COUNTER TO SEE IF=0
                                ;*  :  IF BYTE COUNTER=0
                                ;*  :  :  THEN-UPDATE FORMAT NUMBER FOR NEXT PASS
                                ;*  :  :  ELSE-REPORT FAILURE OF BYTE COUNTER TO GO ZERO WHEN EXPECTED
                                ;*  :  ENDIF
                                ;*  :  DO UNTIL FORMAT NUMBER=7 (ILLEGAL #)
                                ;*  ENDDO
                                ;*ENDTST
1512  4300                      SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1513                            ;*WMC4 MICRO TEST 01
1514                            ;*WMC4 MICRO ERROR 01
1515                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1516                            ;*M8959
1517                            ;*WMC ROM PROGRAM FAILED TO REQUEST NEXT 'SCLK' CYCLE IN 128 CLOCK CYCLES
1518                            ;*TEST ABORTED!
1519                            ;*
1520                            ;*WMC4 MICRO TEST 01
1521                            ;*WMC4 MICRO ERROR 02
1522                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1523                            ;*M8959
1524                            ;*WMC ROM PGM FAILED TO SET 'XFER' BIT IN 128 CLOCK CYCLES AFTER XFR STARTED
1525                            ;*TEST ABORTED!
1526                            ;*
1527                            ;*WMC4 MICRO TEST 01
1528                            ;*WMC4 MICRO ERROR 03
1529                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1530                            ;*M8959
1531                            ;*WMC FAILED TO DROP 'SCLK' BIT AFTER SEVERAL SYSTEM CLOCKS
1532                            ;*TEST ABORTED!
1533                            ;*
1534                            ;*WMC4 MICRO TEST 01
1535                            ;*WMC4 MICRO ERROR 04
```

```
1536                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1537                            ;*M8959
1538                            ;*WMC FAILED TO CLEAR 'XFER' BIT ON 1ST NIBBLE OF THIS BYTE ASSY
1539                            ;*
1540                            ;*WMC4 MICRO TEST 01
1541                            ;*WMC4 MICRO ERROR 05
1542                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1543                            ;*M8959
1544                            ;*WMC FAILED TO SET 'XFER' BIT FOR 2ND NIBBLE ON THIS BYTE ASSY
1545                            ;*BYTE-SCLK COUNT = LLL
1546                            ;*DATA FORMAT = MM
1547                            ;*SKIP COUNT  = NN
1548                            ;*TEST ABORTED!
1549                            ;*
1550                            ;*WMC4 MICRO TEST 01
1551                            ;*WMC4 MICRO ERROR 06
1552                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1553                            ;*M8959
1554                            ;*WMC FAILED TO CLEAR 'XFER' BIT FOR 2ND NIBBLE OF THIS BYTE ASSY
1555                            ;*BYTE-SCLK COUNT = LLL
1556                            ;*DATA FORMAT = MM
1557                            ;*SKIP COUNT  = NN
1558                            ;*TEST ABORTED'
1559                            ;*
1560                            ;*WMC4 MICRO TEST 01
1561                            ;*WMC4 MICRO ERROR 07
1562                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1563                            ;*M8959
1564                            ;*PAD CHARACTER BYTE INCORRECT
1565                            ;*ACTUAL = NNNN
1566                            ;*EXPECTED = NNNN
1567                            ;*
1568                            ;*WMC4 MICRO TEST 01
1569                            ;*WMC4 MICRO ERROR 10
1570                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1571                            ;*M8959
1572                            ;*PARITY BIT ON PAD SHOULD =0 AND WAS =1
1573                            ;*ACTUAL = NNNN
1574                            ;*
1575                            ;*WMC4 MICRO TEST 01
1576                            ;*WMC4 MICRO ERROR 11
1577                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1578                            ;*M8959
1579                            ;*PARITY BIT ON PAD WAS =0 AND SHOULD BE =1
1580                            ;*ACTUAL = NNNN
1581                            ;*
1582                            ;*WMC4 MICRO TEST 01
1583                            ;*WMC4 MICRO ERROR 12
1584                            ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1585                            ;*M8959
1586                            ;*WMC PAD COUNTER NOT = -1 AFTER LAST PAD CHAR RECEIVED
1587                            ;*ACTUAL = NNNN
1588                            ;*BYTE/SCLK COUNT NUMBER = LLL
1589                            ;*
```

```
1590                                    ;*WMC4 MICRO TEST 01
1591                                    ;*WMC4 MICRO ERROR 13
1592                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1593                                    ;*M8959
1594                                    ;*WMC SET 'XFER' ONLY-AFTER ALL DATA AND PAD CHARACTERS OUTPUT
1595                                    ;*
1596                                    ;*WMC4 MICRO TEST 01
1597                                    ;*WMC4 MICRO ERROR 14
1598                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1599                                    ;*M8959
1600                                    ;*WMC BYTE ASSEMBLY FAULT-BYTE ASSEMBLED DOESN'T MATCH EXPECTED DATA
1601                                    ;*ACTUAL = NNNN
1602                                    ;*EXPECTED = NNNN
1603                                    ;*BYTE-SCLK COUNT = LLL
1604                                    ;*DATA FORMAT = MM
1605                                    ;*SKIP COUNT  = NN
1606                                    ;*
1607                                    ;*WMC4 MICRO TEST 01
1608                                    ;*WMC4 MICRO ERROR 15
1609                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1610                                    ;*M8959
1611                                    ;*PARITY BIT ON DATA SHOULD =0 AND WAS =1
1612                                    ;*ACTUAL = NNNN
1613                                    ;*
1614                                    ;*WMC4 MICRO TEST 01
1615                                    ;*WMC4 MICRO ERROR 16
1616                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1617                                    ;*M8959
1618                                    ;*PARITY BIT ON DATA WAS =0 AND SHOULD BE =1
1619                                    ;*ACTUAL = NNNN
1620                                    ;*
1621                                    ;*WMC4 MICRO TEST 01
1622                                    ;*WMC4 MICRO ERROR 17
1623                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1624                                    ;*M8959
1625                                    ;*WMC BYTE COUNTER EXPECTED TO BE = 0-ALL DATA BYTES RECEIVED FROM WMC
1626                                    ;*ACTUAL = NNNN
1627                                    ;*BYTE-SCLK COUNT = LLL
1628                                    ;*DATA FORMAT = MM
1629                                    ;*SKIP COUNT  = NN
1630                                    ;*
1631                                    ;*WMC4 MICRO TEST 01
1632                                    ;*WMC4 MICRO ERROR 20
1633                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1634                                    ;*M8959
1635                                    ;*SPECIAL TERMINATING 'CRC' CHARACTER INCORRECT
1636                                    ;*ACTUAL = NNNN
1637                                    ;*EXPECTED = NNNN
1638                                    ;*
1639                                    ;*WMC4 MICRO TEST 01
1640                                    ;*WMC4 MICRO ERROR 21
1641                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1642                                    ;*M8959
1643                                    ;*WMC BYTE COUNT AT END OF XFR WAS WRONG-SHOULD BE = 361 OCTAL
```

```
1644                                    ;*ACTUAL = NNNN
1645                                    ;*
1646                                    ;*WMC4 MICRO TEST 01
1647                                    ;*WMC4 MICRO ERROR 22
1648                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1649                                    ;*M8959
1650                                    ;*WMC DONE STATUS NOT INDICATED AFTER LAST WMC FUNCTION PERFORMED
1651                                    ;*ACTUAL = NNNN
1652                                    ;*
1653                                    ;*WMC4 MICRO TEST 01
1654                                    ;*WMC4 MICRO ERROR 23
1655                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1656                                    ;*M8959
1657                                    ;*WMC PRESENTED AN 'ECC' CHAR WHEN NOT EXPECTED-ACTUAL = 'XFER' COUNT
1658                                    ;*ACTUAL = NNNN
1659                                    ;*BYTE-SCLK COUNT = LLL
1660                                    ;*DATA FORMAT = MM
1661                                    ;*SKIP COUNT  = NN
1662                                    ;*
1663                                    ;*WMC4 MICRO TEST 01
1664                                    ;*WMC4 MICRO ERROR 24
1665                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1666                                    ;*M8959
1667                                    ;*WMC ECC WORD ON 'BAL' OUTPUT NOT AS EXPECTED
1668                                    ;*ACTUAL = NNNN
1669                                    ;*EXPECTED = NNNN
1670                                    ;*BYTE-SCLK COUNT = LLL
1671                                    ;*DATA FORMAT = MM
1672                                    ;*SKIP COUNT  = NN
1673                                    ;*
1674                                    ;*WMC4 MICRO TEST 01
1675                                    ;*WMC4 MICRO ERROR 25
1676                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1677                                    ;*M8959
1678                                    ;*PARITY BIT ON ECC SHOULD =0 AND WAS =1
1679                                    ;*ACTUAL = NNNN
1680                                    ;*
1681                                    ;*WMC4 MICRO TEST 01
1682                                    ;*WMC4 MICRO ERROR 26
1683                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1684                                    ;*M8959
1685                                    ;*PARITY BIT ON ECC WAS =0 AND SHOULD BE =1
1686                                    ;*ACTUAL = NNNN
1687                                    ;*
1688                                    ;*WMC4 MICRO TEST 01
1689                                    ;*WMC4 MICRO ERROR 27
1690                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1691                                    ;*M8959
1692                                    ;*WMC PRESENTED RESIDUAL CHARACTER BEFORE ACRC CHARACTER
1693                                    ;*
1694                                    ;*WMC4 MICRO TEST 01
1695                                    ;*WMC4 MICRO ERROR 30
1696                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1697                                    ;*M8959
```

```
1698                                    ;*WMC PRESENTED CRC CHARACTER BEFORE ACRC CHARACTER
1699                                    ;*
1700                                    ;*WMC4 MICRO TEST 01
1701                                    ;*WMC4 MICRO ERROR 31
1702                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1703                                    ;*M8959
1704                                    ;*WMC PRESENTED ACRC CHARACTER MORE THAN ONCE
1705                                    ;*
1706                                    ;*WMC4 MICRO TEST 01
1707                                    ;*WMC4 MICRO ERROR 32
1708                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1709                                    ;*M8959
1710                                    ;*WMC FAILED TO SENP PROPER # PAD CHAR. IN RESIDUAL GROUP
1711                                    ;*ACTUAL = NNNN
1712                                    ;*EXPECTED = NNNN
1713                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1714                                    ;*
1715                                    ;*WMC4 MICRO TEST 01
1716                                    ;*WMC4 MICRO ERROR 33
1717                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1718                                    ;*M8959
1719                                    ;*WMC FAILED TO SEND ALL DATA BYTES IN RESIDUAL GROUP
1720                                    ;*ACTUAL = NNNN
1721                                    ;*EXPECTED = NNNN
1722                                    ;*
1723                                    ;*WMC4 MICRO TEST 01
1724                                    ;*WMC4 MICRO ERROR 34
1725                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1726                                    ;*M8959
1727                                    ;*WMC 'ACRC' DATA BYTE NOT SAME AS CALCULATED 'ACRC' DATA BYTE
1728                                    ;*ACTUAL = NNNN
1729                                    ;*EXPECTED = NNNN
1730                                    ;*
1731                                    ;*WMC4 MICRO TEST 01
1732                                    ;*WMC4 MICRO ERROR 35
1733                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1734                                    ;*M8959
1735                                    ;*PARITY BIT ON ACRC SHOULD =0 AND WAS =1
1736                                    ;*ACTUAL = NNNN
1737                                    ;*
1738                                    ;*WMC4 MICRO TEST 01
1739                                    ;*WMC4 MICRO ERROR 36
1740                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1741                                    ;*M8959
1742                                    ;*PARITY BIT ON ACRC WAS =0 AND SHOULD BE =1
1743                                    ;*ACTUAL = NNNN
1744                                    ;*
1745                                    ;*WMC4 MICRO TEST 01
1746                                    ;*WMC4 MICRO ERROR 37
1747                                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1748                                    ;*M8959
1749                                    ;*WMC SET RESIDUAL CHARACTER BEFORE CRC CHARACTERS WERE ALL SENT
1750                                    ;*
1751                                    ;*WMC4 MICRO TEST 01
```

```
1752                    ;*WMC4 MICRO ERROR 40
1753                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1754                    ;*M8959
1755                    ;*WMC FAILED TO SEND ACRC CHARACTER BEFORE CRC CHARACTER
1756                    ;*
1757                    ;*WMC4 MICRO TEST 01
1758                    ;*WMC4 MICRO ERROR 41
1759                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1760                    ;*M8959
1761                    ;*WMC ACTUAL 'CRC' CHARACTER NOT SAME AS SIMULATED 'CRC' CHARACTER
1762                    ;*ACTUAL = NNNN
1763                    ;*EXPECTED = NNNN
1764                    ;*
1765                    ;*WMC4 MICRO TEST 01
1766                    ;*WMC4 MICRO ERROR 42
1767                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1768                    ;*M8959
1769                    ;*PARITY BIT ON CRC SHOULD =0 AND WAS =1
1770                    ;*ACTUAL = NNNN
1771                    ;*
1772                    ;*WMC4 MICRO TEST 01
1773                    ;*WMC4 MICRO ERROR 43
1774                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1775                    ;*M8959
1776                    ;*PARITY BIT ON CRC WAS =0 AND SHOULD BE =1
1777                    ;*ACTUAL = NNNN
1778                    ;*
1779                    ;*WMC4 MICRO TEST 01
1780                    ;*WMC4 MICRO ERROR 44
1781                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1782                    ;*M8959
1783                    ;*WMC FAILED TO SEND THE 'ACRC' CHARACTER BEFORE RESIDUAL CHARACTER WAS PRESENTED
1784                    ;*
1785                    ;*WMC4 MICRO TEST 01
1786                    ;*WMC4 MICRO ERROR 45
1787                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1788                    ;*M8959
1789                    ;*WMC FAILE  TO PRESENT CORRECT # CRC CHARACTERS IN CRC GROUP-ACTUAL - # RECEIVED
1790                    ;*ACTUAL = NNNN
1791                    ;*EXPECTED = NNNN
1792                    ;*
1793                    ;*WMC4 MICRO TEST 01
1794                    ;*WMC4 MICRO ERROR 46
1795                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1796                    ;*M8959
1797                    ;*WMC ACTUAL RESIDUAL DATA BYTE PRESENTED NOT SAME AS EXPECTED RESIDUAL CHAR
1798                    ;*ACTUAL = NNNN
1799                    ;*EXPECTED = NNNN
1800                    ;*
1801                    ;*WMC4 MICRO TEST 01
1802                    ;*WMC4 MICRO ERROR 47
1803                    ;*WMC4-WMC GCR CHARACTER GENERATION TEST
1804                    ;*M8959
1805                    ;*PARITY BIT ON RESIDUAL SHOULD =0 AND WAS  1
```

```
                                        ;*ACTUAL = NNNN
                                        ;*
                                        ;*WMC4 MICRO TEST 01
                                        ;*WMC4 MICRO ERROR 50
                                        ;*WMC4-WMC GCR CHARACTER GENERATION TEST
                                        ;*M8959
                                        ;*PARITY BIT ON RESIDUAL WAS =0 AND SHOULD BE =1
                                        ;*ACTUAL = NNNN
4300                                    $
 (1)                                    ; ***************************************************************************

4300                                    TST1:   TESTX   @1
 (1)  4300  3E  01            7.0                        MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                        CALL    TSET            ;SETUP THE TEST
                                        ;%WMC4-WMC GCR CHARACTER GENERATION TEST
                                        ;&M8959

                                        ;INIT FORMAT CODE COUNTER TO 5 - FORMAT CODE = IMAGE MODE

4305  3E  05                  7.0       TST1ST: MVI     A,@5            ;LOAD THE IMAGE MODE FORMAT CODE
4307  32  89  4A              13.0               STA     FORMT           ;SAVE FORMAT CODE
430A  AF                      4.0               XRA     A
430B  32  88  4A              13.0               STA     EABCNT          ;CLEAR THE LOOP CONTROL
430E  32  A7  4A              13.0               STA     TPBCNT          ;CLEAR THE TEST PASS BYTE COUNT

                                        ;HERE TO UPDATE THE NUMBER OF BYTES TO USE THIS TEST PASS

4311  3A  A7  4A              13.0      TST1NBC: LDA     TPBCNT          ;GET THE BYTE COUNT
4314  3C                      4.0               INR     A
4315  32  A7  4A              13.0               STA     TPBCNT          ;SAVE THE NEW BYTE COUNT TO USE
                                        ;INIT COUNTER CONTROL FOR BYTE COUNTER AND ECODE COUNTER

                                        ;RESTART WRITE MICROCONTROLLER ROM PROGRAM

4318  3E  FF                  7.0       TST1L:  MVI     A,@377          ;-1 TO OFFSET NUMBER
431A  32  8A  4A              13.0               STA     DPNUM           ;INIT DATA PATTERN OFFSET SO 1ST WILL BE "0"
431D  3E  01                  7.0               MVI     A,W.RST         ;GET 'RST' BIT
431F  D3  D3                 10.0               OUT     WMCCTL          ;RESTART WMC
4321  3E  05                  7.0               MVI     A,SSTEP         ;SET THE SYSTEM TO SINGLE STEP MODE
4323  32  84  4A              13.0               STA     CCTLWD          ;SAVE THE CONTROL WORD
4326  D3  F0                 10.0               OUT     CLKCTL          ;WRITE THE CLOCK CONTROL WORD
4328  CD  27  4B             18.0               CALL    CLKSYS
432B  CD  27  4B             18.0               CALL    CLKSYS
432E  AF                      4.0               XRA     A               ;FINISH THE WMC RESTART
432F  D3  D3                 10.0               OUT     WMCCTL
4331  3E  30                  7.0               MVI     A,@60
4333  D3  D7                 10.0               OUT     CNTCTL          ;SELECT BYTE COUNTER
4335  3E  70                  7.0               MVI     A,@160
4337  D3  D7                 10.0               OUT     CNTCTL          ;SELECT THE PAD COUNTER
4339  3E  B0                  7.0               MVI     A,@260
433B  D3  D7                 10.0               OUT     CNTCTL          ;SELECT ECODE COUNTER

                                        ;CLEAR ERROR CODE COUNTER
433D                                            CLRECT                  ;CLEAR ECODE COUNTER
```

```
 (1)  433D  AF              4.0                    XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  433E  D3  D6          10.0                   OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  4340  D3  D6          10.0                   OUT     ERRCNT          ;CLEAR BITS 15-8
1857
1858  4342  32  A3  4A      13.0           STA     RESCNT                  ;CLEAR 'RESIDUAL' CHARACTER COUNTER
1859  4345  32  9F  4A      13.0           STA     RPADCT                  ;CLEAR 'PAD' COUNTER
1860  4348  32  8C  4A      13.0           STA     ECCCNT                  ;CLEAR 'ECC' CHECK COUNTER
1861  434B  32  8D  4A      13.0           STA     CLKCNT                  ;CLEAR THE 'SCLK' COUNTER
1862  434E  32  86  4A      13.0           STA     ABCNT                   ;CLEAR THE ASSEM. BYTE COUNTER
1863  4351  32  87  4A      13.0           STA     ABCNTR
1864  4354  32  9D  4A      13.0           STA     HAVACRC                 ;CLEAR THE HAVE ACRC FLAG
1865  4357  32  9C  4A      13.0           STA     HAVCRC                  ;CLEAR THE HAVE CRC FLAG
1866  435A  32  9E  4A      13.0           STA     HAVRCH                  ;CLEAR THE HAVE RESIDUAL CHARACTER FLAG
1867  435D  32  88  4A      13.0           STA     EABCNT                  ;
1868  4360  32  A5  4A      13.0           STA     CLKCRC                  ;CLEAR THE EXTRA CLOCK-TO-CRC FLAG
1869
1870                                 ;SET DDR CONTROL TO "OUT" FOR DATA INJECTION INTO BAL LOGIC
1871
1872  4363  3E  88              7.0           MVI     A,@210                  ;SET "OUT" BITS
1873  4365  D3  DB              10.0          OUT     DDRCTL                  ;LOAD DDR CONTROL
1874
1875  4367  CD  AA  4A          18.0          CALL    GENDAT                  ;GENERATE THE 1ST SET OF DATA
1876  436A  11  EF  4A          10.0          LXI     D,CDATBL                ;LOAD THE ADDRESS OF "EXPECTED" DATA TABLE
1877  436D  CD  33  4B          18.0          CALL    CLECC                   ;INIT THE ECC WORD - "ECCCHR"
1878  4370  CD  02  4A          18.0          CALL    CLCRC                   ;INIT CRC WORD
1879  4373  CD  16  4A          18.0          CALL    CLACRC                  ;INIT ACRC WORD
1880  4376  06  B4              7.0           MVI     B,180                   ;CLOCK THE SYSTEM THRU THE RESTART AREA
1881  4378  CD  27  4B          18.0  T1RST:  CALL    CLKSYS                  ;CLOCK THE SYSTEM
1882  437B  05                  4.0           DCR     B
1883  437C  C2  78  43          10.0          JNZ     T1RST                   ;STAY HERE TILL DONE COUNTING
1884
1885  437F  D5                  12.0          PUSH    D                       ;SAVE D & E POINTER
1886  4380  3A  A7  4A          13.0          LDA     TPBCNT                  ;GET THE BYTE COUNT TO USE
1887  4383  32  23  4F          13.0          STA     BYTEL                   ;SAVE THE BYTE COUNT LOW VALUE
1888  4386  AF                  4.0           XRA     A
1889  4387  32  24  4F          13.0          STA     BYTEH                   ;CLEAR THE BYTE COUNT HIGH VALUE
1890  438A  CD  46  00          18.0          CALL    LKMOD7                  ;CALL MOD7 ROUTINE FROM OPR U-CODE
1891  438D  3A  A7  4A          13.0          LDA     TPBCNT                  ;GET THE BC AGAIN
1892  4390  3D                  4.0           DCR     A                       ;CALC MOD32
1893  4391  E6  1F              7.0           ANI     @37                     ;SAVE LOW 5 BITS
1894  4393  57                  4.0           MOV     D,A                     ;TEMP SAVE
1895  4394  78                  4.0           MOV     A,B                     ;GET MOD7 REMAINDER FROM LKMOD7
1896  4395  32  A4  4A          13.0          STA     IMRESCNT                ;SAVE THE # RESIDUAL DATA BYTES EXPECTED
1897  4398  0F                  4.0           RRC                             ;POSITION THE MOD7 NUMBER
1898  4399  0F                  4.0           RRC
1899  439A  0F                  4.0           RRC
1900  439B  B2                  4.0           ORA     D                       ;MAKE THE RESID CHAR WORD
1901  439C  32  9B  4A          13.0          STA     RCHRIM                  ;SAVE IT
1902  439F  D3  D1              10.0          OUT     RESCHR                  ;LOAD THE RESID CHAR WORD IN WMC
1903  43A1  3A  A7  4A          13.0          LDA     TPBCNT                  ;GET THE BC AGAIN
1904  43A4  90                  4.0           SUB     B                       ;CALC THE VALUE LOADED INTO THE BYTE COUNTER
1905  43A5  32  85  4A          13.0          STA     IMBYTC                  ;SAVE FOR LATER TESTING
1906
1907  43A8  3E  06              7.0   T1RSTA: MVI     A,6
```

```
1908  43AA  90              4.0          SUB     B               ;CALC THE PAD COUNTER VALUE
1909  43AB  32  A6  4A      13.0         STA     IMPADCT         ;SAVE THE PAD COUNT TO USE
1910  43AE  3C              4.0          INR     A               ;ADD 1 MORE FOR THE INTEL 8253 CHIP
1911  43AF  D3  D5          10.0         OUT     PADCNT          ;LOAD PAD COUNT 7-0
1912
1913  43B1  AF              4.0  T1RSTB: XRA     A
1914  43B2  D3  D5          10.0         OUT     PADCNT          ;LOAD PAD CCOUNTER 15-8
1915  43B4  3A  A6  4A      13.0         LDA     IMPADCT         ;GET PAD COUNT AGAIN
1916  43B7  21  A7  4A      10.0         LXI     H,TPBCNT        ;POINT TO BC TO USE
1917  43BA  86              7.0          ADD     M               ;CALC # TO SEE IF PADCRC BIT NEEDED
1918  43BB  3C              4.0          INR     A               ;IN THE DATACTL WORD
1919  43BC  0F              4.0          RRC                     ;LOAD THE PADCRC BIT - IF =1 THEN PAD
1920  43BD  E6  80          7.0          ANI     BIT7            ;SAVE ONLY THIS BIT
1921  43BF  32  A8  4A      13.0         STA     PDCRC           ;SAVE AS A FLAG
1922
1923  43C2  3A  A6  4A      13.0         LDA     IMPADCT         ;GET THE PAD COUNT EXPECTED
1924  43C5  3C              4.0          INR     A               ;CORRECT FOR THE EXTRA PAD BYTE
1925  43C6  32  A6  4A      13.0         STA     IMPADCT         ;SAVE THE NEW COUNT
1926
1927                                 ;LOAD DATA CONTROL WITH SELECTED FORMAT CODE
1928
1929  43C9  3A  89  4A      13.0  T1RSTC: LDA     FORMT           ;GET THE FORMAT CODE
1930  43CC  CD  1E  4B      18.0         CALL    SHL4            ;POSITION IT TO BITS 6-4
1931  43CF                               ROUT    RO2H            ;ALSO SAVE IN CAS REG 02
 (1)  43CF  D3  85          10.0                 OUT     RO2H            ;WRITE AC INTO RO2H
 (1)  43D1  7F              4.0                  MOV     A,A             ;RETRY LINK
1932  43D2  47              4.0          MOV     B,A             ;SAVE THE FMT DATA
1933  43D3  3A  A8  4A      13.0         LDA     PDCRC           ;GET THE PAD CRC BIT
1934  43D6  B0              4.0          ORA     B               ;ADD IN THE FORMAT NUMBER
1935  43D7  D3  D0          10.0         OUT     DATACTL         ;LOAD DATA CONTROL WORD
1936  43D9  3E  04          7.0          MVI     A,4             ;SET RECORD COUNT TO 1
1937  43DB                               ROUT    RO2L
 (1)  43DB  D3  84          10.0                 OUT     RO2L            ;WRITE AC INTO RO2L
 (1)  43DD  7F              4.0                  MOV     A,A             ;RETRY LINK
1938  43DE  D1              10.0         POP     D               ;RESET THE POINTER IN D & E
1939
1940                                 ;ENABLE WRITE MICROCONTROLLER - START OF DATA XFR - WRITE
1941
1942  43DF  3E  98          7.0  TST1A:  MVI     A,W.ENAB.W.WRITE!W.GCR  ;SELECT 'WMC EN' + 'WRITE'
1943  43E1  D3  D3          10.0         OUT     WMCCTL          ;LOAD WRITE MICROCONTROLLER CONTROL
1944  43E3  AF              4.0          XRA     A
1945  43E4  32  8B  4A      13.0         STA     CLKON           ;CLEAR 'SCLK' ON FLAG
1946
1947  43E7  06  FF          7.0          MVI     B,@377          ;SET SINGLE STEP CLOCK 'TIMEOUT' LIMIT TO MAX POS. COUNT
1948  43E9  48              4.0          MOV     C,B             ;COPY FOR THE 'XFER' TIMEOUT COUNTER
1949
1950  43EA  CD  73  44      18.0  TST1B:  CALL    CKSCLK          ;ISSUE SINGLE STEP CLOCK
1951  43ED  DA  18  43      10.0         JC      TST1L           ;JUMP BACK IF ERROR DETECTED
1952  43F0  DB  D0          10.0         IN      WMCSTA          ;GET WMC STATUS WORD
1953  43F2  E6  20          7.0          ANI     W.XFER          ;SAVE ONLY 'XFER' BIT
1954  43F4  C2  89  44      10.0         JNZ     XFRBIT          ;JUMP IF 'XFER' IS SET - DATA READY IN WRITE DATA WORD
1955  43F7  3A  95  4A      13.0         LDA     SXFER           ;GET THE SOFTWARE XFER FLAG
1956  43FA  A7              4.0          ANA     A
1957  43FB  CA  07  44      10.0         JZ      TST1C           ;JUMP IF XFER NOT SET
```

```
1958  43FE  AF                    4.0          XRA     A               ;CLEAR SOFTWARE FLAG
1959  43FF  32  95  4A            13.0         STA     SXFER           ;FOR NEXT PASS
1960  4402  06  7F                7.0          MVI     B,@177          ;SETUP A WATCHDOG COUNT
1961  4404  C3  AE  44            10.0         JMP     XFRB2           ;PROCESS THE 'XFER' BIT DETECTED
1962
1963                                        ;HERE TO SEE IF 'SCLK' TIMEOUT HAPPENED
1964
1965  4407  05                    4.0    TST1C:  DCR     B               ;COUNT CLOCK CYCLE
1966  4408  C2  19  44            10.0         JNZ     TST1D           ;TIMEOUT? - JUMP IF OK
1967
1968  440B                                      ERR     TST1L,TST1BC
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  440B  CD  09  28            18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0001                                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  440E  01                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  440F  00                                              .BYTE
 (1)  4410  CD  15  28            18.0                TST1BC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4413  DA  18  43            10.0                       JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1969                                        ;>WMC ROM PROGRAM FAILED TO REQUEST NEXT 'SCLK' CYCLE IN 128 CLOCK CYCLES
1970                                        ;<TEST ABORTED!
1971  4416  C3  18  28            10.0         JMP     TSTEND          ;TERMINATE TESTING NOW. - FATAL ERROR
1972
1973                                        ;HERE TO SEE IF 'XFER' BIT TIMEOUT HAPPENED
1974
1975  4419  0D                    4.0    TST1D:  DCR     C               ;DECREMENT 'XFER' WATCHDOG COUNTER
1976  441A  C2  EA  43            10.0         JNZ     TST1B           ;BACK TO LOOP IF OK
1977
1978  441D                                      ERR     TST1L,TST1DC
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  441D  CD  09  28            18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4420  02                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4421  00                                              .BYTE
 (1)  4422  CD  15  28            18.0                TST1DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4425  DA  18  43            10.0                       JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1979                                        ;>WMC ROM PGM FAILED TO SET 'XFER' BIT IN 128 CLOCK CYCLES AFTER XFR STARTED
1980                                        ;<TEST ABORTED!
1981  4428  C3  18  28            10.0         JMP     TSTEND
1982  442B                                S
 (1)                                        ; ***************************************************************************
1983                                        ;SCLKBIT -- HERE WHEN A CLOCK CYCLE RESULTS IN 'SCLK' SETTING IN DBUS
1984                                        ;       CONTROL WORD.  LOAD NEXT DATA SET (18 BITS) INTO DDR REGS.
1985  442B                                S
 (1)                                        ; ***************************************************************************
1986
1987  442B  3A  8B  4A            13.0   SCLKBIT: LDA     CLKON           ;SEE IF 'SCLK' WAS ON
1988  442E  A7                     4.0          ANA     A               ;SET CONDITION BITS
1989  442F  C0                    12.0          RNZ                     ;EXIT IF CLOCK WAS SET-HAVE DATA ALREADY
1990  4430  CD  CA  4A            18.0          CALL    SETDDR          ;THEN LOAD THE DATA INTO THE DDR REGS
1991  4433  C5                    12.0          PUSH    B               ;SAVE REG B & C
1992  4434  06  14                7.0          MVI     B,20            ;GET A WATCHDOG COUNT
1993  4436  CD  27  4B            18.0   SCKWT:  CALL    CLKSYS          ;CLOCK THE SYSTEM - WAIT FOR 'SCLK' TO GO AWAY
1994  4439  DB  D0                10.0          IN      WMCSTA          ;GET THE WMC STATUS WORD
1995  443B  E6  20                 7.0          ANI     W.XFER          ;DID 'XFER' BIT SET?
```

```
 1996  443D  CA  43  44   10.0          JZ      SCKWT1          ;NO - JUMP OVER
 1997  4440  32  95  4A   13.0          STA     SXFER           ;YUP - SET THE FLAG
 1998  4443  DB  C0       10.0  SCKWT1: IN      DBUSSTA         ;GET THE DATA BUS STATUS INFO
 1999  4445  E6  02        7.0          ANI     T.SCLK          ;SEE IF SCLK IS GONE YET
 2000  4447  CA  5E  44   10.0          JZ      SCKGON          ;YUP - PROCEED
 2001  444A  05         4.0          DCR     B               ;NOPE! - DOWNCOUNT WATCHDOG
 2002  444B  C2  36  44   10.0          JNZ     SCKWT           ;STAY IN LOOP TILL NO 'SCLK' OR DIE
 2003  444E                             ERR     SCKWTE,SCKWTC
  (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  444E  CD  09  28   18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
  (1)       0003                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4451  03                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4452  00                                 .BYTE
  (1)  4453  CD  15  28   18.0          SCKWTC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
  (1)  4456  DA  5C  44   10.0                  JC      SCKWTE          ;LOOP ADDRESS IF LOOP SPECIFIED
 2004                          ;>WMC FAILED TO DROP 'SCLK' BIT AFTER SEVERAL SYSTEM CLOCKS
 2005                          ;<TEST ABORTED!
 2006  4459  C3  18  28   10.0          JMP     TSTEND          ;EXIT IF NOT LOOPING ON ERROR
 2007
 2008  445C  C1         10.0  SCKWTE: POP     B
 2009  445D  C9         10.0          RET
 2010  445E  C1         10.0  SCKGON: POP     B               ;ALL OK - NOW RESET REG B & C
 2011  445F  C2  65  44   10.0          JNZ     SCKB1           ;JMP IF NOT TIME TO RELOAD DDR
 2012
 2013  4462  CD  AA  4A   18.0          CALL    GENDAT          ;GENERATE NEW DDR DATA FOR INJECTION
 2014
 2015  4465  06  7F        7.0  SCKB1:  MVI     B,@177          ;GET NEW TIMEOUT COUNT
 2016  4467  78         4.0          MOV     A,B             ;COPY TO REG A
 2017  4468  32  8B  4A   13.0          STA     CLKON           ;SET SCLK "ON" FLAG
 2018  446B  3A  8D  4A   13.0          LDA     CLKCNT          ;GET THE CURRENT SCLK COUNT
 2019  446E  3C         4.0          INR     A               ;UPDATE IT
 2020  446F  32  8D  4A   13.0          STA     CLKCNT          ;SAVE THE NEW COUNT
 2021  4472  C9         10.0          RET
```

```
2023  4473                              S
 (1)                                    ; ***************************************************************************
2024                                    ;CKSCLK -- ROUTINE TO CLOCK THE SYSTEM ONCE AND THEN CHECK FOR PRESENCE
2025                                    ;       OF 'SCLK' FROM THE WMC. IF 'SCLK' IS ON, THEN LOAD NEW DATA INTO
2026                                    ;       THE DDR.
2027  4473                              S
 (1)                                    ; ***************************************************************************
2028
2029  4473  CD  27  4B      18.0  CKSCLK: CALL    CLKSYS          ;CLOCK SYSTEM ONCE
2030  4476  DB  C0          10.0        IN      DBUSSTA         ;GET STATUS OF DATA BUS
2031  4478  E6  02           7.0        ANI     T.SCLK          ;SAVE ONLY THE 'SCLK' BIT
2032  447A  C2  82  44      10.0        JNZ     CKSC1           ;JUMP IF SCLK IS ON
2033  447D  AF               4.0        XRA     A               ;NO SCLK - CLEAR CLOCK ON FLAG
2034  447E  32  8B  4A      13.0        STA     CLKON
2035  4481  C9              10.0        RET                     ;EXIT
2036
2037  4482  CD  2B  44      18.0  CKSC1:  CALL    SCLKBIT         ;LOAD NEW DATA INTO DDR REGS
2038  4485  DA  18  43      10.0        JC      TST1L
2039  4488  C9              10.0        RET                     ;EXIT
2040
2041  4489                              S
 (1)                                    ; ***************************************************************************
2042                                    ;XFRBIT -- HERE WHEN WMC STATUS INDICATES 'XFER' BIT IS SET. ISSUE 20
2043                                    ;       MORE CLOCKS THEN CHECK WRITE DATA WORD.
2044  4489                              S
 (1)                                    ; ***************************************************************************
2045
2046  4489  DB  D0          10.0  XFRBIT: IN      WMCSTA          ;GET THE CURRENT WMC STATUS
2047  448B  32  94  4A      13.0        STA     SWMCSTA         ;SAVE IT FOR THE ECC CHECK (LATER)
2048  448E  06  14           7.0        MVI     B,20            ;ALLOW UP TO 20. WMC CLOCKS TO DROP XFER
2049  4490  CD  73  44      18.0  XFRB1:  CALL    CKSCLK          ;NEED TWO MORE CLOCK CYCLES BEFORE
2050  4493  DA  18  43      10.0        JC      TST1L           ;JUMP IF ERROR DETECTED
2051  4496  DB  D0          10.0        IN      WMCSTA          ;GET WMC STATUS
2052  4498  E6  20           7.0        ANI     W.XFER          ;SAVE THE XFER BIT
2053  449A  CA  A6  44      10.0        JZ      XFRGON          ;OK TO CONTINUE IF XFER IS GONE
2054  449D  05               4.0        DCR     B               ;DECREMENT WATCHDOG COUNTER
2055  449E  C2  90  44      10.0        JNZ     XFRB1           ;LOOP TIL =0 OR XFER IS GONE
2056  44A1                              ERR     TST1L,XFRGON
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44A1  CD  09  28      18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0004                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44A4  04                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44A5  00                                      .BYTE
 (1)  44A6  CD  15  28      18.0              XFRGON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44A9  DA  18  43      10.0                      JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2057                                    ;>WMC FAILED TO CLEAR 'XFER' BIT ON 1ST NIBBLE OF THIS BYTE ASSY
2058                                    ;HERE TO LOOK FOR 'XFER' TO SET FOR THE 2ND NIBBLE OF THIS BYTE
2059
2060  44AC  06  7F           7.0        MVI     B,@177
2061  44AE  CD  73  44      18.0  XFRB2:  CALL    CKSCLK          ;SEE IF SCLK IS SET (ALSO CLOCK)
2062  44B1  DA  18  43      10.0        JC      TST1L           ;JUMP IF ERROR DETECTED
2063  44B4  DB  D0          10.0        IN      WMCSTA          ;GET WMC STATUS
2064  44B6  E6  20           7.0        ANI     W.XFER          ;SAVE ONLY XFER BIT
2065  44B8  C2  CD  44      10.0        JNZ     XFRB3           ;JUMP IF XFER IS SET - OK
```

```
2066  44BB  05                    4.0          DCR     B               ;LOOK FOR TIMEOUT
2067  44BC  C2  AE  44           10.0          JNZ     XFRB2           ;GO BACK TO LOOP IF OK
2068  44BF                                    ERR     TST1L,XFRB2C,3
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44BF  CD  09  28           18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0005                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44C2  05                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44C3  03                                 .BYTE   3
 (1)  44C4  CD  15  28           18.0          XFRB2C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44C7  DA  18  43           10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2069                                    ;>WMC FAILED TO SET 'XFER' BIT FOR 2ND NIBBLE ON THIS BYTE ASSY
2070                                    ;<TEST ABORTED!
2071  44CA  C3  18  28           10.0          JMP     TSTEND
2072
2073                                    ;HERE TO LOOK FOR 'XFER' BIT TO GO AWAY TO INDICATE FULL BYTE IS AVAIL.
2074                                    ;FROM THE 'WRTDAT' WORD
2075
2076  44CD  06  14                 7.0  XFRB3:  MVI     B,20
2077  44CF  DB  E0                10.0          IN      INTSTA          ;GET THE DATA PARITY BIT
2078  44D1  32  96  4A            13.0          STA     SINTSTA         ;SAVE FOR PARITY CHECK
2079  44D4  CD  73  44            18.0  XFRB3A: CALL    CKSCLK          ;CLOCK THE SYSTEM TILL XFER BIT GOES AWAY
2080  44D7  DA  18  43            10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
2081  44DA  DB  D0                10.0          IN      WMCSTA          ;GET THE WMC STATUS
2082  44DC  E6  20                 7.0          ANI     W.XFER          ;SAVE ONLY THE XFER BIT
2083  44DE  CA  F3  44            10.0          JZ      XFRB4           ;JUMP IF GONE - OK
2084  44E1  05                     4.0          DCR     B               ;DOWNCOUNT WATCHDOG
2085  44E2  C2  D4  44            10.0          JNZ     XFRB3A          ;BACK TO LOOP IF OK
2086  44E5                                    ERR     TST1L,XFRB3C,3
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44E5  CD  09  28           18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0006                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44E8  06                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44E9  03                                 .BYTE   3
 (1)  44EA  CD  15  28           18.0          XFRB3C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44ED  DA  18  43           10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2087                                    ;>WMC FAILED TO CLEAR 'XFER' BIT FOR 2ND NIBBLE OF THIS BYTE ASSY
2088                                    ;<TEST ABORTED!
2089  44F0  C3  18  28           10.0          JMP     TSTEND
2090                                    ;HERE TO TEST THE ASSEMBLED BYTE DATA AGAINST THE EXPECTED BYTE DATA
2091
2092  44F3  CD  27  4B            18.0  XFRB4:  CALL    CLKSYS          ;ALLOW THE BYTE COUNTER TO TICK
2093  44F6  CD  27  4B            18.0          CALL    CLKSYS          ;TO LATCH IT.
```

```
2095  44F9                              $
 (1)                                    ; ***************************************************************************
2096                                    ;XFRRDY -- ROUTINE TO DETERMINE WHAT CHARACTER IS ON WMC OUTPUT WHEN
2097                                    ;       'XFER' IS SET.  ALSO, CHECKS FOR ERRORS BASED ON BYTE COUNTER
2098                                    ;       AND PAD COUNTER.
2099  44F9                              $
 (1)                                    ; ***************************************************************************
2100
2101  44F9  3A  94  4A      13.0  XFRRDY: LDA     SWMCSTA         ;GET WMC STATUS WORD
2102  44FC  E6  1E           7.0          ANI     @36             ;SAVE JUST THE SPECIAL CHAR INFO.
2103  44FE  CA  28  45      10.0          JZ      XFRB4D          ;NOTHING SPECIAL - CHECK DATA BYTE
2104
2105  4501  3A  94  4A      13.0          LDA     SWMCSTA         ;GET THE STATUS AGAIN
2106  4504  E6  10           7.0          ANI     W.ECC           ;SEE IF ITS 'ECC' CHARACTER
2107  4506  C2  32  47      10.0          JNZ     CKECC           ;JUMP IF ITS 'ECC'
2108
2109                                    ;TEST FOR 'ACRC', 'CRC', OR 'RESID' CHARACTER
2110
2111  4509  3A  8C  4A      13.0          LDA     ECCCNT          ;GET 'XFER' COUNT
2112  450C  3C               4.0          INR     A               ;ADD 1
2113  450D  32  8C  4A      13.0          STA     ECCCNT          ;SAVE THE UPDATED  OUNT
2114  4510  3A  94  4A      13.0          LDA     SWMCSTA         ;GET WMC STATUS WORD AGAIN
2115  4513  E6  04           7.0          ANI     W.ACRC          ;SEE IF 'ACRC' CHARACTER READY
2116  4515  C2  B1  47      10.0          JNZ     CKACRC          ;GO TEST 'ACRC' CHARACTER
2117
2118  4518  3A  94  4A      13.0          LDA     SWMCSTA         ;GET STATUS AGAIN
2119  451B  E6  08           7.0          ANI     W.CRC           ;SEE IF ITS 'CRC' READY BIT
2120  451D  C2  8F  48      10.0          JNZ     CKCRC           ;GO TEST 'CRC' DATA BYTE
2121
2122  4520  3A  94  4A      13.0          LDA     SWMCSTA         ;GET STATUS AGAIN
2123  4523  E6  02           7.0          ANI     W.RESID         ;SEE IF 'RESID' READY BIT (IF NOT ITS FATAL!)
2124  4525  C2  3C  49      10.0          JNZ     CKRESID         ;GO TEST 'RESID' DATA BYTE
2125
2126                                    ;HERE TO CHECK THE DATA BYTE JUST ASSEMBLED
2127
2128  4528  3A  8C  4A      13.0  XFRB4D: LDA     ECCCNT          ;GET 'XFER' COUNT
2129  452B  3C               4.0          INR     A               ;ADD 1
2130  452C  32  8C  4A      13.0          STA     ECCCNT          ;SAVE THE UPDATED COUNT
2131  452F  E5              12.0          PUSH    H               ;SAVE H&L
2132  4530  21  85  4A      10.0          LXI     H,IMBYTC        ;POINT TO THE AMT OF BYTES EXPECTED
2133  4533  3A  86  4A      13.0          LDA     ABCNT           ;THEN GET # BYTES ASSEMBLED BY THE WMC
2134  4536  BE               7.0          CMP     M               ;SEE IF DONE
2135  4537  E1              10.0          POP     H               ;RESET REG H & L
2136  4538  DA  09  46      10.0          JC      XFRB4C          ;CONTINUE TO COMPARE DATA BYTES
2137
2138
2139  453B  E5              12.0  XFRB4R: PUSH    H               ;SAVE H&L
2140  453C  21  A4  4A      10.0          LXI     H,IMRESCNT      ;GET THE RESIDUAL DATA BYTE COUNT
2141  453F  3A  A3  4A      13.0          LDA     RESCNT          ;GET THE NUMBER OF RESIDUAL CHARACTERS TO DATE
2142  4542  BE               7.0          CMP     M               ;SEE IF MORE RESIDUAL DATA BYTES EXPECTED
2143  4543  E1              10.0          POP     H               ;RESTORE H&L
2144  4544  CA  51  45      10.0           7      CKPAD           ;JUMP IF NO MORE DATA BYTES EXPECTED
2145  4547  3A  A3  4A      13.0          .DA     RESCNT          ;GET THE RESIDUAL CHARACTERS COUNT
2146  454A  3C               4.0          INR     A               ;UPDATE FOR THIS CHARACTER
```

```
2147  454B  32  A3  4A      13.0          STA     RESCNT          ;SAVE 'T
2148  454E  C3  09  46      10.0          JMP     XFRB4C          ;GO CHECK THE RESIDUAL DATA BYTE
2149
2150                                ;HERE TO CHECK THE PAD CHARACTERS EXPECTED
2151
2152  4551  3A  A6  4A      13.0  CKPAD:  LDA     IMPADCT         ;GET THE PAD COUNT EXPECTED
2153  4554  47                     4.0          MOV     B,A             ;TEMP STORE
2154  4555  3A  A8  4A      13.0          LDA     PDCRC           ;GET THE PAD CRC FLAG
2155  4558  A7                     4.0          ANA     A
2156  4559  C2  5D  45      10.0          JNZ     CKPADA          ;JUMP OVER IF EXTRA PAD EXPECTED
2157  455C  05                     4.0          DCR     B               ;CORRECT THE PAD COUNT FOR PHANTOM XFR
2158
2159  455D  3A  9F  4A      13.0  CKPADA: LDA     RPADCT          ;GET THE RECEIVED PAD COUNT #
2160  4560  B8                     4.0          CMP     B               ;SAME AS EXPECTED (TIME FOR PHANTOM XFR)
2161  4561  CA  CC  45      10.0          JZ      CKHACK          ;GO CHECK FOR THE PHANTOM 'XFR'
2162  4564  3A  9F  4A      13.0          LDA     RPADCT          ;GET THE PAD CHARACTERS COUNT
2163  4567  3C                     4.0          INR     A               ;UPDATE FOR THIS CHARACTER
2164  4568  32  9F  4A      13.0          STA     RPADCT          ;SAVE IT
2165  456B  AF                     4.0          XRA     A               ;CLEAR THE ACCUMULATOR
2166  456C                               ROUT    EDATA           ;STORE THE EXPECTED PAD CHARACTER
 (1)  456C  D3  95          10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  456E  7F                     4.0          MOV     A,A             ;RETRY LINK
2167  456F  DB  D3          10.0          IN      WRTDAT          ;GET THE ACTUAL PAD CHARACTER
2168  4571                               ROUT    ADATA           ;STORE THE ACTUAL PAD CHARACTER
 (1)  4571  D3  94          10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4573  7F                     4.0          MOV     A,A             ;RETRY LINK
2169  4574  A7                     4.0          ANA     A               ;SET THE CONDITION BITS
2170  4575  CA  7D  45      10.0          JZ      PADCNO          ;CONTINUE IF ZERO
2171  4578                               ERRB    TST1L,PADCNO
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4578  CD  12  28      18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0007                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  457B  07                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  457C  00                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  457D  CD  15  28      18.0          PADCNO::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4580  DA  18  43      10.0                 JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2172                          ;>PAD CHARACTER BYTE INCORRECT
2173                          ;HERE TO CHECK THE PARITY BIT ON THE PAD ASSEMBLED
2174
2175  4583  DB  D3          10.0          IN      WRTDAT          ;GET THE DATA BYTE AGAIN
2176  4585                               ROUT    ADATA           ;SAVE AS ACTUAL DATA
 (1)  4585  D3  94          10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4587  7F                     4.0          MOV     A,A             ;RETRY LINK
2177  4588  A7                     4.0          ANA     A               ;SET THE CONDITION BITS
2178  4589  E4  9B  45      18.0          CPO     PADPO           ;CHECK FOR ODD PARITY
2179  458C  DA  18  43      10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
2180  458F  DB  D3          10.0          IN      WRTDAT          ;GET THE DATA BYTE AGAIN
2181  4591  A7                     4.0          ANA     A               ;SET THE CONDITION BITS
2182  4592  EC  AF  45      18.0          CPE     PADPE           ;CHECK FOR PARITY EVEN - BIT -1
2183  4595  DA  18  43      10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
2184  4598  C3  C1  45      10.0          JMP     CKPAD1          ;ON TO THE NEXT SECTION....
2185
2186                          ;HERE TO CHECK FOR PARITY ODD - BIT -0
2187
```

```
2188  459B  3A 96 4A      13.0  PADPO:  LDA     SINTSTA         ;GET THE PARITY BIT (IN BIT 0)
2189  459E  E6 10            7.0          ANI     WDR.P           ;SAVE THE PARITY BIT
2190  45A0  CA AE 45        10.0          JZ      PADPOE          ;EXIT IF OK
2191  45A3                               ERRA    PADPOE,PADPOC
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  45A3  CD 0F 28        18.0                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0008                           MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45A6  08                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45A7  00                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45A8  CD 15 28        18.0          PADPOC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45AB  DA AE 45        10.0                  JC      PADPOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2192                                ;>PARITY BIT ON PAD SHOULD =0 AND WAS =1
2193  45AE  C9                10.0  PADPOE: RET
2194
2195                                ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
2196
2197  45AF  3A 96 4A      13.0  PADPE:  LDA     SINTSTA         ;GET THE BIT
2198  45B2  E6 10            7.0          ANI     WDR.P           ;SAVE THE PARITY BIT
2199  45B4  C0               12.0          RNZ
2200  45B5                               ERRA    PADPEE,PADPEC
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  45B5  CD 0F 28        18.0                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0009                           MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45B8  09                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45B9  00                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45BA  CD 15 28        18.0          PADPEC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45BD  DA C0 45        10.0                  JC      PADPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
2201                                ;>PARITY BIT ON PAD WAS =0 AND SHOULD BE =1
2202  45C0  C9                10.0  PADPEE: RET
2203  45C1  DB D3            10.0  CKPAD1: IN      WRTDAT          ;GET THE PAD CHARACTER AGAIN
2204  45C3  CD 3D 4B         18.0          CALL    GENECC          ;GO INCLUDE THE PAD CHARACTER IN THE ECC
2205  45C6  CD D8 49         18.0          CALL    GENCRC          ;GO INCLUDE THE PAD CHARACTER IN THE CRC
2206  45C9  C3 89 48         10.0          JMP     SCHREX          ;CONTINUE WITH THE TEST
2207
2208                                ;HERE IF A 'XFER' DATA BYTE WAS DETECTED AND ALL DATA BYTES WERE
2209                                ;RECEIVED AND ALL PAD BYTES REVEIVED...I.E. WE ARE ABOUT TO GIVE
2210                                ;AN EXTRA CLOCK TO THE CRC GENERATOR (CRC GROUP - IS 9 CHAR).
2211
2212  45CC  3A A5 4A      13.0  CKHACK: LDA     CLKCRC          ;GET THE EXTRA CLOCK-TO-CRC FLAG
2213  45CF  A7                4.0          ANA     A               ;SEE IF HERE BEFORE
2214  45D0  C2 FB 45         10.0          JNZ     XFRERR          ;YES - THEN NO MORE PHANTOM 'XFR'S' ALLOWED
2215  45D3  3E FF             7.0          MVI     A,@377          ;NO - THEN ALLOW ONLY THE ONE
2216  45D5  32 A5 4A      13.0          STA     CLKCRC          ;UPDATE THE RECEIVED FLAG
2217  45D8  3A 8C 4A      13.0          LDA     ECCCNT          ;GET THE ECC CHAR. COUNTER
2218  45DB  3D                4.0          DCR     A               ;ADJUST FOR ADDITION OF THE PHANTOM 'XFR'
2219  45DC  32 8C 4A      13.0          STA     ECCCNT          ;RESTORE ADJUSTED COUNT
2220
2221                                ;CHECK TO SEE IF PAD COUNTER = -1 AFTER LAST PAD RECEIVED
2222
2223  45DF  DB D5            10.0          IN      PADCNT          ;GET LOW BYTE OF COUNTER
2224  45E1  47                4.0          MOV     B,A             ;TEMP STORAGE
2225  45E2  DB D5            10.0          IN      PADCNT          ;GET HIGH BYTE (THROW AWAY)
2226  45E4  78                4.0          MOV     A,B             ;RESET 'A'
2227  45E5  FE FF             7.0          CPI     @377            ;SEE IF PHANTOM CAUSED COUNT OF -1
```

```
2228  45E7  CA  F2  45   10.0          JZ      XFRRC1              ;JUMP IF COUNTER = -1 (EXPECTED)
2229  45EA                             ROUT    ADATA               ;ERROR - SAVE PAD COUNTER VALUE
 (1)  45EA  D3  94        10.0                 OUT     ADATA               ;WRITE AC INTO ADATA
 (1)  45EC  7F             4.0                 MOV     A,A                 ;RETRY LINK
2230  45ED                             ERRA    TST1L,XFRRC1,1
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  45ED  CD  0F  28    18.0                 CALL    ERLPA               ;PROCESS ERROR - DO 2.3
 (1)      000A                             MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45F0  0A                             .BYTE   MSGN                ;MESSAGE NUMBER ID
 (1)  45F1  01                             .BYTE   1                   ;PRINT ROUTINE NUMBER
 (1)  45F2  CD  15  28    18.0          XFRRC1::        CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45F5  DA  18  43    10.0                 JC      TST1L               ;LOOP ADDRESS IF LOOP SPECIFIED
2231                             ;>WMC PAD COUNTER NOT - -1 AFTER LAST PAD CHAR RECEIVED
2232  45F8  C3  89  48    10.0          JMP     SCHREX              ;CONTINUE TO CLOCK THE WMC
2233  45FB                       XFRERR: ERR     TST1L,XFB4C
 (1)                                  ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45FB  CD  09  28    18.0                 CALL    ERLP                ;PROCESS ERROR - DO 2.3
 (1)      000B                             MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45FE  0B                             .BYTE   MSGN                ;MESSAGE NUMBER ID
 (1)  45FF  00                             .BYTE
 (1)  4600  CD  15  28    18.0          XFB4C:: CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4603  DA  18  43    10.0                 JC      TST1L               ;LOOP ADDRESS IF LOOP SPECIFIED
2234                             ;>WMC SET 'XFER' ONLY - AFTER ALL DATA AND PAD CHARACTERS OUTPUT
2235
2236  4606  C3  89  48    10.0          JMP     SCHREX              ;CONTINUE WITH THE TEST
2237
2238                             ;GET ASSEMBLED BAL DATA BYTE FROM WRITE DATA WORDS
2239
2240  4607  3A  86  4A    13.0    XFRB4C: LDA     ABCNT               ;GET THE BYTE # UNDER TEST
2241  460C  3C             4.0            INR     A                   ;ADD 1
2242  460D  32  86  4A    13.0            STA     ABCNT               ;SAVE THE BYTE NUMBER - UPDATED
2243  4610                             ROUT    R05L
 (1)  4610  D3  8A        10.0                 OUT     R05L                ;WRITE AC INTO R05L
 (1)  4612  7F             4.0                 MOV     A,A                 ;RETRY LINK
2244  4613  AF             4.0            XRA     A
2245  4614                             ROUT    R05H
 (1)  4614  D3  8B        10.0                 OUT     R05H                ;WRITE AC INTO R05H
 (1)  4616  7F             4.0                 MOV     A,A                 ;RETRY LINK
2246  4617  EB             4.0            XCHG                        ;PUT 'EXPECTED' POINTER IN H & L
2247  4618  7E             7.0            MOV     A,M                 ;GET CORRECT BAL BYTE
2248  4619                             ROUT    EDATA               ;SAVE 'EXPECTED' DATA
 (1)  4619  D3  95        10.0                 OUT     EDATA               ;WRITE AC INTO EDATA
 (1)  461B  7F             4.0                 MOV     A,A                 ;RETRY LINK
2249  461C  DB  D3        10.0            IN      WRTDAT              ;GET ASSEMBLED DATA BYTE
2250  461E                             ROUT    ADATA               ;SAVE 'ACTUAL' DATA
 (1)  461E  D3  94        10.0                 OUT     ADATA               ;WRITE AC INTO ADATA
 (1)  4620  7F             4.0                 MOV     A,A                 ;RETRY LINK
2251  4621  BE             7.0            CMP     M                   ;EXPECTED = ACTUAL?
2252  4622  EB             4.0            XCHG                        ;RESET H & L REGS
2253  4623  13             6.0            INX     D                   ;POINT TO NEXT 'EXPECTED' ENTRY
2254  4624  CA  32  46    10.0            JZ      TST1XC              ;JUMP OVER ERROR IF DATA BYTE OK
2255  4627  3A  86  4A    13.0            LDA     ABCNT               ;GET THE COUNT OF ASSEM. BYTES AT FAILURE
2256  462A  32  88  4A    13.0            STA     EABCNT              ;SAVE FOR THE LOOP CONTROL
```

```
2258  462D                                        ERRB    DELOOP,TST1XC,3
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  462D  CD  12  28        18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000C                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4630  0C                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4631  03                                    .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  4632  CD  15  28        18.0        TST1XC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4635  DA  17  47        10.0                JC      DELOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2259                                    ;>WMC BYTE ASSEMBLY FAULT - BYTE ASSEMBLED DOESN'T MATCH EXPECTED DATA
2260
2261                                    ;HERE TO CHECK THE PARITY BIT ON THE DATA ASSEMBLED
2262
2263  4638  DB  D3            10.0  XFRB4P: IN      WRTDAT          ;GET THE DATA BYTE AGAIN
2264  463A                                  ROUT    ADATA           ;SAVE AS ACTUAL DATA
 (1)  463A  D3  94            10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  463C  7F                 4.0                MOV     A,A             ;RETRY LINK
2265  463D  A7                 4.0          ANA     A               ;SET THE CONDITION BITS
2266  463E  E4  50  46         18.0          CPO     DATPO           ;CHECK FOR ODD PARITY
2267  4641  DA  18  43         10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
2268  4644  DB  D3            10.0          IN      WRTDAT          ;GET THE DATA BYTE AGAIN
2269  4646  A7                 4.0          ANA     A               ;SET THE CONDITION BITS
2270  4647  EC  64  46         18.0          CPE     DATPE           ;CHECK FOR PARITY EVEN - BIT =1
2271  464A  DA  18  43         10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
2272  464D  C3  76  46         10.0          JMP     XFRB5           ;ON TO THE NEXT SECTION....
2273
2274                                    ;HERE TO CHECK FOR PARITY ODD - BIT =0
2275
2276  4650  3A  96  4A         13.0  DATPO:  LDA     SINTSTA         ;GET THE PARITY BIT (IN BIT 0)
2277  4653  E6  10               7.0          ANI     WDR.P           ;SAVE THE PARITY BIT
2278  4655  CA  63  46         10.0          JZ      DATPOE          ;EXIT IF OK
2279  4658                                    ERRA    DATPOE,DATPOC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4658  CD  0F  28        18.0                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       000D                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  465B  0D                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  465C  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  465D  CD  15  28        18.0        DATPOC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4660  DA  63  46        10.0                JC      DATPOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2280                                    ;>PARITY BIT ON DATA SHOULD =0 AND WAS =1
2281  4663  C9                 10.0  DATPOE: RET
2282
2283                                    ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
2284
2285  4664  3A  96  4A         13.0  DATPE:  LDA     SINTSTA         ;GET THE BIT
2286  4667  E6  10               7.0          ANI     WDR.P           ;SAVE THE PARITY BIT
2287  4669  C0                 12.0          RNZ
```

```
2289  466A                                    ERRA    DATPEE,DATPEC
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  466A  CD  0F  28        18.0                    CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000E                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  466D  0E                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  466E  00                                        .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  466F  CD  15  28        18.0            DATPEC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4672  DA  75  46        10.0                    JC      DATPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
2290                                    ;>PARITY BIT ON DATA WAS =0 AND SHOULD BE =1
2291  4675  C9                10.0    DATPEE: RET
2292  4676  DB  D3            10.0    XFRB5:  IN      WRTDAT          ;GET THE DATA AGAIN
2293  4678  CD  3D  4B        18.0            CALL    GENECC          ;CALCULATE THE ECC
2294  467B  CD  D8  49        18.0            CALL    GENCRC          ;ADD DATA TO CRC CHAR
2295  467E  CD  1E  4A        18.0            CALL    GENACRC         ;AND TO THE ACRC CHAR
2296
2297  4681  E5                12.0            PUSH    H
2298  4682  21  85  4A        10.0            LXI     H,IMBYTC        ;POINT TO THE AMT OF BYTES EXPECTED
2299  4685  3A  86  4A        13.0            LDA     ABCNT           ;THEN GET # BYTES ASSEMBLED BY THE WMC
2300  4688  BE                 7.0            CMP     M               ;SEE IF DONE
2301  4689  E1                10.0            POP     H               ;RESET REG H & L
2302  468A  C2  89  48        10.0            JNZ     SCHREX          ;BACK TO LOOP IF MORE TO DO
2303
2304  468D  DB  D4            10.0            IN      BYTCNT          ;GET BYTE COUNTER 7-0
2305  468F  47                 4.0            MOV     B,A             ;TEMP STORE
2306  4690  DB  D4            10.0            IN      BYTCNT          ;GET BYTE CNT 15-8
2307  4692  78                 4.0            MOV     A,B
2308  4693  A7                 4.0            ANA     A
2309  4694  CA  9F  46        10.0            JZ      XFRB5C          ;JUMP IF BYTE COUNT OK
2310  4697                                    ROUT    ADATA           ;STORE THE ACTUAL BYTE COUNTER 7-0
 (1)  4697  D3  94            10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4699  7F                 4.0                    MOV     A,A             ;RETRY LINK
2311  469A                                    ERRA    TST1L,XFRB5C,3
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  469A  CD  0F  28        18.0                    CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        000F                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  469D  0F                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  469E  03                                        .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  469F  CD  15  28        18.0            XFRB5C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46A2  DA  18  43        10.0                    JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2312                                    ;>WMC BYTE COUNTER EXPECTED TO BE = 0 - ALL DATA BYTES RECEIVED FROM WMC
2313
2314  46A5  C3  89  48        10.0            JMP     SCHREX          ;BACK TO CLOCK LOOP FOR OTHER CHARACTERS
```

```
2316  46A8                                  S
 (1)                                        ; ***************************************************************************
2317                                        ;HERE IF ALL DATA, ECC, CRC, ACRC, AND RESID CHAR TESTED
2318  46A8                                  S
 (1)                                        ; ***************************************************************************
2319
2320  46A8  DB  D3            10.0  QUIT:   IN      WRTDAT          ;GET ACTUAL WMC 'CRC' CHARACTER
2321  46AA                                  ROUT    ADATA
 (1)  46AA  D3  94            10.0          OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46AC  7F                 4.0          MOV     A,A             ;RETRY LINK
2322  46AD  21  98  4A         10.0          LXI     H,CRCCHR        ;POINT TO EXPECTED CRC CHAR (SIMULATED)
2323  46B0  BE                  7.0          CMP     M               ;ACTUAL=SIMULATED?
2324  46B1  CA  BD  46         10.0          JZ      QUIT1           ;JUMP IF OK
2325  46B4  7E                  7.0          MOV     A,M             ;GET SIMULATED CRC CHAR.
2326  46B5                                  ROUT    EDATA
 (1)  46B5  D3  95            10.0          OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  46B7  7F                 4.0          MOV     A,A             ;RETRY LINK
2327  46B8                                  ERRB    TST1L,QUIT1
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46B8  CD  12  28        18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0010                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46BB  10                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46BC  00                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  46BD  CD  15  28  \     18.0  QUIT1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46C0  DA  18  43        10.0          JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2328                                        ;>SPECIAL TERMINATING 'CRC' CHARACTER INCORRECT
2329  46C3  3A  85  4A         13.0          LDA     IMBYTC          ;GET THE IMAGE BYTE COUNTER
2330  46C6  A7                  4.0          ANA     A               ;SET THE CONDITION BITS
2331  46C7  CA  E3  46         10.0          JZ      QUIT4           ;SKIP THE BYTE COUNTER CHECK IF THE BYTECOUNTER
2332                                                                ;WAS LOADED WITH ZEROS
2333  46CA  DB  D4             10.0          IN      BYTCNT          ;GET BYTE COUNT BITS 7-0
2334  46CC  47                  4.0          MOV     B,A             ;TEMP SAVE
2335  46CD  DB  D4             10.0          IN      BYTCNT          ;GET BITS 15-8
2336  46CF  78                  4.0          MOV     A,B             ;USE ONLY 7-0 FOR THE CHECK
2337  46D0                                  ROUT    ADATA           ;SAVE THE ACTUAL BYTE COUNT
 (1)  46D0  D3  94            10.0          OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46D2  7F                 4.0          MOV     A,A             ;RETRY LINK
2338  46D3  FE  F1             7.0          CPI     @361            ;SHOULD TERMINATE WITH -17 OCTAL (361)
2339  46D5  CA  DD  46         10.0          JZ      QUIT2           ;JUMP IF BYTE COUNTER OK
2340  46D8                                  ERRA    TST1L,QUIT2
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  46D8  CD  0F  28        18.0                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)      0011                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46DB  11                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46DC  00                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  46DD  CD  15  28        18.0  QUIT2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46E0  DA  18  43        10.0          JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2341                                        ;>WMC BYTE COUNT AT END OF XFR WAS WRONG - SHOULD BE = 361 OCTAL
2342
2343  46E3  DB  D0             10.0  QUIT4:  IN      WMCSTA          ;GET THE WMC STATUS
2344  46E5                                  ROUT    ADATA           ;SAVE THE ACTUAL STATUS FOR PRINTOUT
 (1)  46E5  D3  94            10.0          OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46E7  7F                 4.0          MOV     A,A             ;RETRY LINK
2345  46E8  E6  40             7.0          ANI     W.DONN          ;DONE?
```

```
2346  46EA  CA  F2  46      10.0          JZ      QUIT3             ;YES - EXIT THE TEST
2347  46ED                                ERRA    TST1L,QUIT3
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  46ED  CD  0F  28      18.0                  CALL    ERLPA             ;PROCESS ERROR - DO 2.3
 (1)       0012                             MSGN    =       MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46F0  12                              .BYTE   MSGN              ;MESSAGE NUMBER ID
 (1)  46F1  00                              .BYTE                     ;PRINT ROUTINE NUMBER
 (1)  46F2  CD  15  28      18.0          QUIT3:: CALL    CKLOP             ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46F5  DA  18  43      10.0                  JC      TST1L             ;LOOP ADDRESS IF LOOP SPECIFIED
2348                              ;>WMC DONE STATUS NOT INDICATED AFTER LAST WMC FUNCTION PERFORMED
2349
2350  46F8  3A  A7  4A      13.0          LDA     TPBCNT            ;GET THE CURRENT BYTE COUNT USED
2351  46FB  FE  26           7.0          CPI     DTEND-CDATBL      ;DONE WITH ALL COMBINATIONS?
2352  46FD  C2  11  43      10.0          JNZ     TST1NBC           ;JUMP IF NOT AT END OF ROPE...
2353
2354  4700                                ENDTST  TST1ST
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4700                            REQ     7                 ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4700  CD  06  28      18.0                  CALL    REQST
 (2)  4703  00                              .BYTE                     ;DATA PATTERN NUMBER
 (2)  4704  00  00                          .WORD                     ;SYSTEM '"' COUNT
 (2)  4706  00  00                          .WORD                     ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4708  00                              .BYTE                     ;DATA COMPARE FLAG IF =1
 (2)  4709  07                              .BYTE   7                 ;REQUEST CODE
 (1)  470A  3A  9A  4F      13.0          LDA     ITERA             ;GET ITERATION COUNT
 (1)  470D  3D               4.0          DCR     A                 ;DOWNCOUNT
 (1)  470E  32  9A  4F      13.0          STA     ITERA             ;SAVE COUNT
 (1)  4711  F2  05  43      10.0          JP      TST1ST            ;DO TEST UNTIL TILL = 0
2355  4714  C3  18  28      10.0          JMP     TSTEND            ;ALL DONE
2356
2357                              ;HERE IS THE DATA COMPARE ERROR LOOP CONTROL ROUTINE
2358
2359                              ;WILL ENABLE THE TEST TO LOOP AT THE SPECIFIED BYTE IN ERROR
2360
2361  4717  3A  94  4F      13.0  DELOOP: LDA     LPFLG             ;SEE IF LOOP FLAG STILL SET
2362  471A  A7                    4.0          ANA     A
2363  471B  C2  24  47      10.0          JNZ     DELP1             ;JUMP IF LOOPING ON ERROR
2364  471E  32  88  4A      13.0          STA     EABCNT            ;NOT LOOPING - CLEAR THE FLAG
2365  4721  C3  76  46      10.0          JMP     XFRB5             ;AND EXIT
2366
2367  4724  3A  86  4A      13.0  DELP1:  LDA     ABCNT             ;GET THE CURRENT COUNT OF ASSEMBLED BYTES
2368  4727  47                    4.0          MOV     B,A               ;TEMP STORE
2369  4728  3A  88  4A      13.0          LDA     EABCNT            ;GET THE BYTE COUNT AT THE DETECTED ERROR
2370  472B  B8                    4.0          CMP     B                 ;SEE IF AT THE SAME BYTE COUNT
2371  472C  CA  18  43      10.0          JZ      TST1L             ;THEN JUMP TO RESTART TEST IF SAME
2372  472F  C3  76  46      10.0          JMP     XFRB5             ;ELSE, CONTINUE WITH THE TEST
```

```
                                        S
                                        ; ***************************************************************************
                                        ;CKECC -- ROUTINE TO CHECK THE 'ECC' CHARACTER PRESENTED BY THE WMC
                                        ;       LOGIC AGAINST A CACULATED ECC BYTE. WILL REPORT AN ERROR IF
                                        ;       THE 'ECC' FLAG (IN WMCSTA WORD) COMES UP BEFORE OR AFTER THE
                                        ;       8TH ASSEMBLED BYTE FROM THE WMC OR IF AN ACTUAL ECC BYTE
                                        ;       ASSEMBLY ERROR WAS DETECTED.
                                        S
                                        ; ***************************************************************************

4732  3A  8C  4A      13.0  CKECC:  LDA     ECCCNT          ;GET 'XFER' CYCLE COUNT
4735                          ROUT    ADATA
4735  D3  94            10.0          OUT     ADATA           ;WRITE AC INTO ADATA
4737  7F                 4.0          MOV     A,A             ;RETRY LINK
4738  FE  07             7.0          CPI     7               ;READY FOR ECC CHAR?
473A  CA  48  47         10.0          JZ      ECCCK           ;JUMP IF EXPECTED NOW
473D                          ERRA    TST1L,CKECCC,3
                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
473D  CD  0F  28         18.0                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
      0013                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
4740  13                      .BYTE   MSGN            ;MESSAGE NUMBER ID
4741  03                      .BYTE   3               ;PRINT ROUTINE NUMBER
4742  CD  15  28         18.0  CKECCC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
4745  DA  18  43         10.0          JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                              ;>WMC PRESENTED AN 'ECC' CHAR WHEN NOT EXPECTED - ACTUAL = 'XFER' COUNT

                              ;HERE TO CHECK THE ACTUAL ECC BYTE FROM THE WMC LOGIC

4748  21  97  4A       10.0  ECCCK:  LXI     H,ECCCHR        ;POINT TO 'IMAGE' OF HARDWARE ECC WORD
474B  7E                 7.0          MOV     A,M
474C                          ROUT    EDATA
474C  D3  95            10.0          OUT     EDATA           ;WRITE AC INTO EDATA
474E  7F                 4.0          MOV     A,A             ;RETRY LINK
474F  DB  D3            10.0          IN      WRTDAT          ;GET ACTUAL ECC WORD FROM WMC
4751                          ROUT    ADATA
4751  D3  94            10.0          OUT     ADATA           ;WRITE AC INTO ADATA
4753  7F                 4.0          MOV     A,A             ;RETRY LINK
4754  BE                 7.0          CMP     M               ;IS IT OK?
4755  CA  5D  47         10.0          JZ      ECCCK1          ;JUMP IF SAME-AS EXPECTED
4758                          ERRB    TST1L,ECCCK1,3
                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
4758  CD  12  28         18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
      0014                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
475B  14                      .BYTE   MSGN            ;MESSAGE NUMBER ID
475C  03                      .BYTE   3               ;PRINT ROUTINE NUMBER
475D  CD  15  28         18.0  ECCCK1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
4760  DA  18  43         10.0          JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                              ;>WMC ECC WORD ON 'BAL' OUTPUT NOT AS EXPECTED

4763  3A  9E  4A       13.0          LDA     HAVRCH          ;GET THE "HAVE RESID" FLAG
4766  A7                 4.0          ANA     A
4767  CA  6D  47         10.0          JZ      ECCCK2          ;JUMP IF DON'T HAVE RESID YET
476A  32  A9  4A         13.0          STA     LASTCRC         ;THIS MUST BE LAST ECC...SET EXPECT LAST CRC FLAG
```

```
2406                                    ;HERE TO CHECK THE PARITY BIT ON THE ECC ASSEMBLED
2407
2408  476D  DB  D3          10.0  ECCCK2: IN      WRTDAT          ;GET THE ECC BYTE AGAIN
2409  476F                        ROUT    ADATA           ;SAVE AS ACTUAL DATA
 (1)  476F  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4771  7F               4.0                MOV     A,A             ;RETRY LINK
2410  4772  A7               4.0          ANA     A               ;SET THE CONDITION BITS
2411  4773  E4  85  47       18.0          CPO     ECCPO           ;CHECK FOR ODD PARITY
2412  4776  DA  18  43       10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
2413  4779  DB  D3           10.0          IN      WRTDAT          ;GET THE ECC BYTE AGAIN
2414  477B  A7               4.0          ANA     A               ;SET THE CONDITION BITS
2415  477C  EC  99  47       18.0          CPE     ECCPE           ;CHECK FOR PARITY EVEN - BIT =1
2416  477F  DA  18  43       10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
2417  4782  C3  AB  47       10.0          JMP     CKECC1          ;ON TO THE NEXT SECTION....
2418
2419                                    ;HERE TO CHECK FOR PARITY ODD - BIT =0
2420
2421  4785  3A  96  4A       13.0  ECCPO:  LDA     SINTSTA         ;GET THE PARITY BIT (IN BIT 0)
2422  4788  E6  10            7.0          ANI     WDR.P           ;SAVE THE PARITY BIT
2423  478A  CA  98  47       10.0          JZ      ECCPOE          ;EXIT IF OK
2424  478D                        ERRA    ECCPOE,ECCPOC
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  478D  CD  0F  28       18.0                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0015                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4790  15                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4791  00                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4792  CD  15  28       18.0                ECCPOC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4795  DA  98  47       10.0                JC      ECCPOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2425                                    ;>PARITY BIT ON ECC SHOULD =0 AND WAS =1
2426  4798  C9               10.0  ECCPOE: RET
2427
2428                                    ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
2429
2430  4799  3A  96  4A       13.0  ECCPE:  LDA     SINTSTA         ;GET THE BIT
2431  479C  E6  10            7.0          ANI     WDR.P           ;SAVE THE PARITY BIT
2432  479E  C0               12.0          RNZ
2433  479F                        ERRA    ECCPEE,ECCPEC
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  479F  CD  0F  28       18.0                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0016                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47A2  16                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47A3  00                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  47A4  CD  15  28       18.0                ECCPEC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47A7  DA  AA  47       10.0                JC      ECCPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
2434                                    ;>PARITY BIT ON ECC WAS =0 AND SHOULD BE =1
2435  47AA  C9               10.0  ECCPEE: RET
2436  47AB  CD  33  4B       18.0  CKECC1: CALL    CLECC
2437  47AE  C3  89  48       10.0          JMP     SCHREX          ;GO BACK TO CLOCK LOOP
```

```
2439  47B1                                  S
 (1)                                        ; ************************************************************************
2440                                        ;CKACRC -- ROUTINE TO CHECK 'ACRC' CHARACTER ON WMC OUTOUT.  ALSO CHECKS
2441                                        ;       THAT THE RESIDUAL CHARACTER HAS NOT BEEN PRESENTED BEFORE 'ACRC',
2442                                        ;       THAT THE WMC HAS NOT PREVIOUSLY SET 'CRC', AND THAT A PREVIOUS
2443                                        ;       'ACRC' CHARACTER HAS NOT BEEN PRESENTED.
2444                                        ;       'HAVRCH' IS THE RESIDUAL CHARACTER FLAG.
2445                                        ;       'HAVACRC' IS THE AUX-CRC CHARACTER FLAG.
2446                                        ;       'HAVCRC' IS THE CRC CHARACTER FLAG.
2447  47B1                                  S
 (1)                                        ; ************************************************************************
2448
2449  47B1   3A   9E   4A      13.0  CKACRC: LDA     HAVRCH          ;GET RESID FLAG
2450  47B4   A7                4.0         ANA     A
2451  47B5   CA   BD   47      10.0         JZ      CKA1            ;JUMP IF RESIDUAL CHAR NOT RECEIVED YET
2452  47B8                               ERR     TST1L,CKA1
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  47B8   CD   09   28      18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0017                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47BB   17                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47BC   00                                    .BYTE
 (1)  47BD   CD   15   28      18.0           CKA1:: CALL    CK_OP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  47C0   DA   18   43      10.0                 JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2453                                        ;>WMC PRESENTED RESIDUAL CHARACTER BEFORE ACRC CHARACTER
2454  47C3   3A   9C   4A      13.0         LDA     HAVCRC          ;GET CRC RECEIVED FLAG
2455  47C6   A7                4.0         ANA     A
2456  47C7   CA   CF   47      10.0         JZ      CKA2
2457  47CA                               ERR     TST1L,CKA2
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  47CA   CD   09   28      18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0018                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47CD   18                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47CE   00                                    .BYTE
 (1)  47CF   CD   15   28      18.0           CKA2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  47D2   DA   18   43      10.0                 JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2458                                        ;>WMC PRESENTED CRC CHARACTER BEFORE ACRC CHARACTER
2459  47D5   3A   9D   4A      13.0         LDA     HAVACRC         ;GET ACRC CHAR RECEIVED FLAG
2460  47D8   A7                4.0    '    ANA     A
2461  47D9   CA   E1   47      10.0         JZ      CKA3
2462  47DC                               ERR     TST1L,CKA3
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  47DC   CD   09   28      18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0019                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47DF   19                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47E0   00                                    .BYTE
 (1)  47E1   CD   15   28      18.0           CKA3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  47E4   DA   18   43      10.0                 JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2463                                        ;>WMC PRESENTED ACRC CHARACTER MORE THAN ONCE
2464  47E7   3A   9F   4A      13.0  CKA3A: LDA     RPADCT          ;GET # RECEIVED PAD CHARACTERS
2465  47EA                               ROUT    ADATA           ;SAVE ACTUAL
 (1)  47EA   D3   94            10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  47EC   7F                 4.0                  MOV     A,A             ;RETRY LINK
2466  47ED   47                 4.0         MOV     B,A
2467  47EE   3A   A6   4A      13.0         LDA     IMPADCT         ;GET # DATA BYTES IN RESID. GROUP
```

```
2468  47F1  3D                    4.0          DCR     A
2469  47F2                               CKA3B:  ROUT    EDATA
 (1)  47F2  D3  95                10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  47F4  7F                     4.0                  MOV     A,A             ;RETRY LINK
2470  47F5  B8                    4.0          CMP     B                       ;EXPECTED=ACTUAL
2471  47F6  CA  FE  47             10.0         JZ      CKA3C                   ;JUMP IF OK
2472  47F9                               ERRB    TST1L,CKA3C,1
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  47F9  CD  12  28            18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       001A                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47FC  1A                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47FD  01                                       .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  47FE  CD  15  28            18.0          CKA3C:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4801  DA  18  43            10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2473                          ;>WMC FAILED TO SEND PROPER # PAD CHAR. IN RESIDUAL GROUP
2474
2475  4804  3A  A3  4A            13.0  CKAC4:  LDA     RESCNT          ;GET # RECEIVED RESIDUAL DATA BYTES
2476  4807  47                    4.0          MOV     B,A
2477  4808                               ROUT    ADATA           ;SAVE ACTUAL
 (1)  4808  D3  94                10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  480A  7F                     4.0                  MOV     A,A             ;RETRY LINK
2478  480B  3A  A4  4A            13.0         LDA     IMRESCNT        ;GET EXPECTED #
2479  480E                               ROUT    EDATA
 (1)  480E  D3  95                10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4810  7F                     4.0                  MOV     A,A             ;RETRY LINK
2480  4811  B8                    4.0          CMP     B                       ;EXPECTED=ACTUAL
2481  4812  CA  1A  48            10.0         JZ      CKAC4C                  ;JUMP IF OK
2482  4815                               ERRB    TST1L,CKAC4C
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4815  CD  12  28            18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       001B                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4818  1B                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4819  00                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  481A  CD  15  28            18.0          CKAC4C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  481D  DA  18  43            10.0                  JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2483                          ;>WMC FAILED TO SEND ALL DATA BYTES IN RESIDUAL GROUP
2484  4820  CD  6D  4A            18.0  CKA5:   CALL  . ACRCFI          ;FINISH THE 'ACRC' SIMULATION CHARACTER
2485  4823  3E  FE                  7.0          MVI     A,@376
2486  4825  32  9D  4A            13.0         STA     HAVACRC         ;SET THE 'ACRC' RECEIVED FLAG
2487  4828  DB  D3                  10.0         IN      WRTDAT          ;GET ACTUAL 'ACRC' CHARACTER
2488  482A  21  A0  4A            10.0         LXI     H,ACRCHR        ;POINT TO 'EXPECTED' CHAR
2489  482D  BE                    7.0          CMP     M               ;ACTUAL 'ACRC' RECEIVED OK?
2490  482E  CA  3D  48            10.0         JZ      CKA4            ;JUMP IF OK
2491  4831                               ROUT    ADATA           ;SAVE ACTUAL 'ACRC'
 (1)  4831  D3  94                10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4833  7F                     4.0                  MOV     A,A             ;RETRY LINK
2492  4834  7E                    7.0          MOV     A,M             ;GET EXPECTED
2493  4835                               ROUT    EDATA           ;SAVE IT
 (1)  4835  D3  95                10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4837  7F                     4.0                  MOV     A,A             ;RETRY LINK
```

```
2495  4838                                        ERRB    TST1L,CKA4
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4838  CD  12  28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       001C                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  483B  1C                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  483C  00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  483D  CD  15  28        18.0              CKA4::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4840  DA  18  43        10.0                      JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2496                                    ;>WMC 'ACRC' DATA BYTE NOT SAME AS CALCULATED 'ACRC' DATA BYTE
2497                                    ;HERE TO CHECK THE PARITY BIT ON THE ACRC ASSEMBLED
2498
2499  4843  DB  D3        10.0          IN      WRTDAT          ;GET THE ACRC BYTE AGAIN
2500  4845                              ROUT    ADATA           ;SAVE AS ACTUAL DATA
 (1)  4845  D3  94        10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4847  7F             4.0                  MOV     A,A             ;RETRY LINK
2501  4848  A7             4.0          ANA     A               ;SET THE CONDITION BITS
2502  4849  E4  5B  48     18.0          CPO     ACRCPO          ;CHECK FOR ODD PARITY
2503  484C  DA  18  43     10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
2504  484F  DB  D3        10.0          IN      WRTDAT          ;GET THE ACRC BYTE AGAIN
2505  4851  A7             4.0          ANA     A               ;SET THE CONDITION BITS
2506  4852  EC  6F  48     18.0          CPE     ACRCPE          ;CHECK FOR PARITY EVEN - BIT  1
2507  4855  DA  18  43     10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
2508  4858  C3  81  48     10.0          JMP     CACRC1          ;ON TO THE NEXT SECTION....
2509
2510                                    ;HERE TO CHECK FOR PARITY ODD - BIT =0
2511  485B  3A  96  4A     13.0  ACRCPO: LDA     SINTSTA         ;GET THE PARITY BIT (IN BIT 0)
2512  485E  E6  10          7.0          ANI     WDR.P           ;SAVE THE PARITY BIT
2513  4860  CA  6E  48     10.0          JZ      ACRCOE          ;EXIT IF OK
2514  4863                              ERRA    ACRCOE,ACRCOC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4863  CD  0F  28        18.0                        CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       001D                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4866  1D                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4867  00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4868  CD  15  28        18.0              ACRCOC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  486B  DA  6E  48        10.0                      JC      ACRCOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2515                                    ;>PARITY BIT ON ACRC SHOULD =0 AND WAS =1
2516  486E  C9             10.0  ACRCOE: RET
2517
2518                                    ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
2519  486F  3A  96  4A     13.0  ACRCPE: LDA     SINTSTA         ;GET THE BIT
2520  4872  E6  10          7.0          ANI     WDR.P           ;SAVE THE PARITY BIT
2521  4874  C0             12.0          RNZ
```

```
2523  4875                                        ERRA    ACRCEE,ACRCEC
 (1)                                              ;FLAG ERROR - WITH ALTUAL DATA ''ADATA'' VALID
 (1)  4875  CD  0F  28      18.0                          CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       001E                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4878  1E                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4879  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  487A  CD  15  28      18.0                ACRCEC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  487D  DA  80  48      10.0                            JC      ACRCEE          ;LOOP ADDRESS IF LOOP SPECIFIED
2524                                  ;>PARITY BIT ON ACRC WAS =0 AND SHOULD BE =1
2525  4880  C9                 10.0  ACRCEE: RET
2526  4881  DB  D3            10.0  CACRC1: IN      WRTDAT          ;GET THE 'ACRC' CHARACTER AGAIN
2527  4883  CD  3D  4B        18.0          CALL    GENECC          ;GO INCLUDE THE 'ACRC' CHARACTER IN THE ECC
2528  4886  CD  D8  49        18.0          CALL    GENCRC          ;ADD IT TO 'CRC' SIMULATION CHARACTER
2529  4889  06  7F             7.0  SCHREX: MVI     B,@177          ;RESET THE WATCHDOG COUNTERS
2530  488B  48                  4.0          MOV     C,B
2531  488C  C3  07  44        10.0          JMP     TST1C           ;GO BACK TO CLOCK LOOP
```

```
                                        $
 (1)                                    ; *****************************************************************************
                                        ;CKCRC -- ROUTINE TO CHECK THE WMC 'CRC' DATA BYTE.  CHECKS THAT 'ACRC'
                                        ;       CHARACTER WAS ALREADY RECEIVED, THAT THIS CRC CHARACTER IS NOT
                                        ;       THE 7TH CRC CHARACTER, AND THAT THE ACTUAL WMC 'CRC' CHARACTER
                                        ;       IS THE SAME AS THE CALCULATED 'CRC' CHARACTER.
                                        $
 (1)                                    ; *****************************************************************************

488F 3A 9E 4A      13.0  CKCRC:  LDA     HAVRC           ;GET THE HAVE RESIDUAL CHARACTER FLAG
4892 A7             4.0          ANA     A               ;SET THE CONDITION BITS
4893 CA 9D 48      10.0          JZ      CKCRC1          ;CONTINUE - NO RESIDUAL CHAR. YET
4896 3A A9 4A      13.0          LDA     LASTCRC         ;GET THE "LAST CRC CHARACTER" FLAG
4899 A7             4.0          ANA     A               ;SET THE CONDITION BITS
489A C2 A6 46      10.0          JNZ     QUIT            ;ALL CHARACTERS RECEIVED - WMC SHOULD BE DONE

489D 3A 9E 4A      13.0  CKCRC1: LDA     HAVRCH          ;GET RESIDUAL CHAR. RECEIVED FLAG
48A0 A7             4.0          ANA     A
48A1 CA A9 48      10.0          JZ      CKC1            ;JUMP IF OK-NO RESIDUAL CHAR. YET
48A4                             ERR     TST1L,CKC1
 (1)                             ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1) 48A4 CD 09 28  18.0                         CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)      001F                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 48A7 1F                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 48A8 00                                     .BYTE
 (1) 48A9 CD 15 28  18.0                 CKC1::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1) 48AC DA 18 43  10.0                         JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                 ;>WMC SET RESIDUAL CHARACTER BEFORE CRC CHARACTERS WERE ALL SENT

48AF 3A 9D 4A      13.0          LDA     HAVACRC         ;GET ACRC CHAR. RECEIVED FLAG
48B2 A7             4.0          ANA     A
48B3 C2 BB 48      10.0          JNZ     CKC2            ;JUMP IF RECEIVED ACRC
48B6                             ERR     TST1L,CKC2
 (1)                             ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1) 48B6 CD 09 28  18.0                         CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)      0020                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 48B9 20                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 48BA 00                                     .BYTE
 (1) 48BB CD 15 28  18.0                 CKC2::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1) 48BE DA 18 43  10.0                         JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                 ;>WMC FAILED TO SEND ACRC CHARACTER BEFORE CRC CHARACTER
48C1 06 06          7.0          MVI     B,6             ;SET THE CRC LIMIT TO 6
48C3 3A A8 4A      13.0          LDA     PDCRC           ;SEE IF ANOTHER EXPECTED
48C6 A7             4.0          ANA     A
48C7 C2 CC 48      10.0          JNZ     CRC2A           ;JUMP IF 1 MORE PAD CHAR EXPECTED
48CA 3E 07          7.0          MVI     A,7             ;SET LIMIT TO 7
48CC 3A 9C 4A      13.0  CRC2A:  LDA     HAVCRC          ;GET # CRC RECEIVED
48CF 3C             4.0          INR     A               ;UPDATE THE COUNT
48D0 32 9C 4A      13.0          STA     HAVCRC          ;SAVE IT
48D3 FE 01          7.0          CPI     1               ;1ST CRC CHARACTER?
48D5 C2 DB 48      10.0          JNZ     CRC4
48D8 CD 0D 4A      18.0          CALL    CRCFIN          ;FINISH CRC CHAR SIMULATION-ONLY ON 1ST CHAR.
48DB DB D3         10.0  CRC4:   IN      WRTDAT          ;GET ACTUAL WMC 'CRC' CHARACTER
48DD                             ROUT    ADATA
```

```
  (1)  48DD  D3  94          10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  48DF  7F               4.0                 MOV     A,A             ;RETRY LINK
2571   48E0  21  98  4A       10.0         LXI     H,CRCCHR        ;POINT TO EXPECTED CRC CHAR (SIMULATED)
2572   48E3  BE               7.0         CMP     M               ;ACTUAL=SIMULATED?
2573   48E4  CA  F0  48       10.0         JZ      CRC5            ;JUMP IF OK
2574   48E7  7E               7.0         MOV     A,M             ;GET SIMULATED CRC CHAR.
2575   48E8                               ROUT    EDATA
  (1)  48E8  D3  95          10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  48EA  7F               4.0                 MOV     A,A             ;RETRY LINK
2576   48EB                               ERRB    TST1L,CRC5
  (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  48EB  CD  12  28      18.0                         CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0021                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  48EE  21                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  48EF  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  48F0  CD  15  28      18.0                 CRC5::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  48F3  DA  18  43      10.0                         JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2577                                   ;>WMC ACTUAL 'CRC' CHARACTER NOT SAME AS SIMULATED 'CRC' CHARACTER
2578
2579                                   ;HERE TO CHECK THE PARITY BIT ON THE CRC ASSEMBLED
2580
2581   48F6  DB  D3               10.0         IN      WRTDAT          ;GET THE CRC BYTE AGAIN
2582   48F8                               ROUT    ADATA           ;SAVE AS ACTUAL DATA
  (1)  48F8  D3  94          10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  48FA  7F               4.0                 MOV     A,A             ;RETRY LINK
2583   48FB  A7               4.0         ANA     A               ;SET THE CONDITION BITS
2584   48FC  E4  0E  49       18.0         CPO     CRCPO           ;CHECK FOR ODD PARITY
2585   48FF  DA  18  43       10.0         JC      TST1L           ;JUMP IF ERROR DETECTED
2586   4902  DB  D3           10.0         IN      WRTDAT          ;GET THE CRC BYTE AGAIN
2587   4904  A7               4.0         ANA     A               ;SET THE CONDITION BITS
2588   4905  EC  22  49       18.0         CPE     CRCPE           ;CHECK FOR PARITY EVEN - BIT =1
2589   4908  DA  18  43       10.0         JC      TST1L           ;JUMP IF ERROR DETECTED
2590   490B  C3  34  49       10.0         JMP     CKCRC2          ;ON TO THE NEXT SECTION....
2591
2592                                   ;HERE TO CHECK FOR PARITY ODD - BIT =0
2593
2594   490E  3A  96  4A       13.0  CRCPO:  LDA     SINTSTA         ;GET THE PARITY BIT (IN BIT 0)
2595   4911  E6  10               7.0         ANI     WDR.P           ;SAVE THE PARITY BIT
2596   4913  CA  21  49       10.0         JZ      CRCPOE          ;EXIT IF OK
2597   4916                               ERRA    CRCPOE,CRCPOC
  (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
  (1)  4916  CD  0F  28      18.0                         CALL    ERLPA           ;PROCESS ERROR - DO 2.3
  (1)       0022                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4919  22                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  491A  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  491B  CD  15  28      18.0                 CRCPOC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  491E  DA  21  49      10.0                         JC      CRCPOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2598                                   ;>PARITY BIT ON CRC SHOULD =0 AND WAS =1
2599   4921  C9               10.0  CRCPOE: RET
2600                                   ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
2601   4922  3A  96  4A       13.0  CRCPE:  LDA     SINTSTA         ;GET THE BIT
2602   4925  E6  10               7.0         ANI     WDR.P           ;SAVE THE PARITY BIT
2603   4927  C0               12.0         RNZ
```

```
2605  4928                                        ERRA    CRCPEE,CRCPEC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4928  CD  0F  28      18.0                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0023                                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  492B  23                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  492C  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  492D  CD  15  28      18.0          CRCPEC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4930  DA  33  49      10.0                  JC      CRCPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
2606                                  ;>PARITY BIT ON CRC WAS =0 AND SHOULD BE =1
2607  4933  C9                 10.0  CPCPEE: RET
2608  4934  DB  D3            10.0  CKCRC2: IN      WRTDAT          ;GET THE CRC CHARACTER AGAIN
2609  4936  CD  3D  4B        18.0          CALL    GENECC          ;GO INCLUDE THE PAD CHARACTER IN THE ECC
2610  4939  C3  89  48        10.0          JMP     SCHREX          ;BACK TO CLOCK LOOP
```

```
                                        S
                                        ; ***************************************************************************
                                        ;CKRESID -- ROUTINE TO CHECK THE WMC RESIDUAL CHARACTER AND RESIDUAL
                                        ;       DATA GROUP BYTE RECEIVED COUNT, THAT 'ACRC' WAS RECEIVED, THAT
                                        ;       6 'CRC' CHARACTERS WERE RECEIVED, AND THAT THE PROPER AMOUNT
                                        ;       OF PAD CHARACTERS WERE RECEIVED.
                                        S
                                        ; ***************************************************************************

493C 3A 9D 4A      13.0  CKRESID: LDA    HAVACRC          ;GET ACRC CHAR RECEIVED FLAG
493F A7             4.0          ANA    A
4940 C2 48 49      10.0          JNZ    CKR1             ;JUMP IF ACRC RECEIVED-OK
4943                             ERR    TST1L,CKR1
                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
4943 CD 09 28      18.0                 CALL   ERLP             ;PROCESS ERROR - DO 2.3
     0024                             MSGN   =      MSGN+1      ,UPDATE MESSAGE NUMBER FOR THIS
4946 24                               .BYTE  MSGN             ;MESSAGE NUMBER ID
4947 00                               .BYTE
4948 CD 15 28      18.0          CKR1:: CALL   CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
494B DA 18 43      10.0                 JC     TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                 ;>WMC FAILED TO SEND THE 'ACRC' CHARACTER BEFORE RESIDUAL CHARACTER WAS PRESENTED

494E 06 06          7.0          MVI    B,6              ;GET # EXPECTED CRC CHAR
4950 3A A8 4A      13.0          LDA    PDCRC            ;SEE IF 1 LESS
4953 A7             4.0          ANA    A                ;IF PADCRC IS SET, MAKE COUNT 5
4954 CA 59 49      10.0          JZ     CKR1A
4957 06 05          7.0          MVI    B,5
4959 3A 9C 4A      13.0  CKR1A:  LDA    HAVCRC           ;GET CRC CHAR RECEIVED COUNT
495C B8             4.0          CMP    B                ;HAVE PROPER AMT OF CRC BYTES?
495D CA 6C 49      10.0          JZ     CKR2             ;JUMP IF OK
4960                             ROUT   ADATA            ;SAVE ACTUAL # OF RECEIVED CRC CHAR.
4960 D3 94         10.0                 OUT    ADATA           ;WRITE AC INTO ADATA
4962 7F             4.0                 MOV    A,A             ;RETRY LINK
4963 78             4.0          MOV    A,B              ;GET THE EXPECTED COUNT
4964                             ROUT   EDATA            ;SAVE FOR ERROR PRINTOUT
4964 D3 95         10.0                 OUT    EDATA           ;WRITE AC INTO EDATA
4966 7F             4.0                 MOV    A,A             ;RETRY LINK
4967                             ERRB   TST1L,CKR2
                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
4967 CD 12 28      18.0                 CALL   ERLPB            ;PROCESS ERROR - DO 2.3
     0025                             MSGN   =      MSGN+1      ;UPDATE MESSAGE NUMBER FOR THIS
496A 25                               .BYTE  MSGN             ;MESSAGE NUMBER ID
496B 00                               .BYTE                   ;PRINT ROUTINE NUMBER
496C CD 15 28      18.0          CKR2:: CALL   CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
496F DA 18 43      10.0                 JC     TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                 ;>WMC FAILED TO PRESENT CORRECT # CRC CHARACTERS IN CRC GROUP - ACTUAL = # RECEIVED
4972 3A 9B 4A      13.0          LDA    RCHRIM           ;GET EXPECTED RESIDUAL CHAR
4975                             ROUT   EDATA            ;SAVE EXPECTED
4975 D3 95         10.0                 OUT    EDATA           ;WRITE AC INTO EDATA
4977 7F             4.0                 MOV    A,A             ;RETRY LINK
4978 47             4.0          MOV    B,A
4979 DB D3         10.0          IN     WRTDAT           ;GET ACTUAL 'RESID' CHAR.
497B B8             4.0          CMP    B                ;ACTUAL=EXPECTED
497C CA 87 49      10.0          JZ     CKR5             ;JUMP IF OK
```

```
2644  497F                                    ROUT    ADATA
 (1)  497F  D3  94          10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4981  7F               4.0                     MOV     A,A             ;RETRY LINK
2645  4982                                    ERRB    TST1L,CKR5
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4982  CD  12  28      18.0                         CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0026                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4985  26                                         .BYTE   MSGN;           ;MESSAGE NUMBER ID
 (1)  4986  00                                         .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4987  CD  15  28      18.0                 CKR5::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  498A  DA  18  43      10.0                         JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2646                                    ;>WMC ACTUAL RESIDUAL DATA BYTE PRESENTED NOT SAME AS EXPECTED RESIDUAL CHAR
2647
2648                                    ;HERE TO CHECK THE PARITY BIT ON THE RESIDUAL ASSEMBLED
2649
2650  498D  DB  D3               10.0         IN      WRTDAT          ;GET THE RESIDUAL BYTE AGAIN
2651  498F                                    ROUT    ADATA           ;SAVE AS ACTUAL DATA
 (1)  498F  D3  94          10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4991  7F               4.0                     MOV     A,A             ;RETRY LINK
2652  4992  A7               4.0         ANA     A               ;SET THE CONDITION BITS
2653  4993  E4  A5  49       18.0         CPO     RESPO           ;CHECK FOR ODD PARITY
2654  4996  DA  18  43       10.0         JC      TST1L           ;JUMP IF ERROR DETECTED
2655  4999  DB  D3           10.0         IN      WRTDAT          ;GET THE RESIDUAL BYTE AGAIN
2656  499B  A7               4.0         ANA     A               ;SET THE CONDITION BITS
2657  499C  EC  B9  49       18.0         CPE     RESPE           ;CHECK FOR PARITY EVEN - BIT  1
2658  499F  DA  18  43       10.0         JC      TST1L           ;JUMP IF ERROR DETECTED
2659  49A2  C3  CB  49       10.0         JMP     CKRES1          ;ON TO THE NEXT SECTION....
2660
2661                                    ;HERE TO CHECK FOR PARITY ODD - BIT =0
2662
2663  49A5  3A  96  4A        13.0 RESPO:  LDA     SINTSTA         ;GET THE PARITY BIT (IN BIT 0)
2664  49A8  E6  10             7.0         ANI     WDR.P           ;SAVE THE PARITY BIT
2665  49AA  CA  B8  49        10.0         JZ      RESPOE          ;EXIT IF OK
2666  49AD                                 ERRA    RESPOE,RESPOC
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  49AD  CD  0F  28        18.0                     CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)      0027                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  49B0  27                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  49B1  00                                         .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  49B2  CD  15  28        18.0                 RESPOC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  49B5  DA  B8  49        10.0                         JC      RESPOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2667                                    ;>PARITY BIT ON RESIDUAL SHOULD =0 AND WAS =1
2668  49B8  C9                 10.0 RESPOE: RET
2669
2670                                    ;HERE TO CHECK FOR PARITY BIT -1 (EVEN)
2671
2672  49B9  3A  96  4A        13.0 RESPE:  LDA     SINTSTA         ;GET THE BIT
2673  49BC  E6  10             7.0         ANI     WDR.P           ;SAVE THE PARITY BIT
2674  49BE  C0                12.0         RNZ
```

```
2676  49BF                                        ERRA    RESPEE,RESPEC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  49BF   CD   0F   28        18.0                        CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0028                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  49C2   28                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  49C3   00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  49C4   CD   15   28        18.0                RESPEC::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  49C7   DA   CA   49        10.0                        JC      RESPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
2677                                ;>PARITY BIT ON RESIDUAL WAS =0 AND SHOULD BE =1
2678  49CA   C9                    10.0  RESPEE: RET
2679  49CB   DB   D3               10.0  CKRES1: IN      WRTDAT          ;GET THE RESIDUAL CHARACTER AGAIN
2680  49CD   CD   3D   4B         18.0          CALL    GENECC          ;GO INCLUDE THE RESIDUAL CHARACTER IN THE ECC
2681  49D0   3E   FF              7.0          MVI     A,@377
2682  49D2   32   9E   4A        13.0          STA     HAVRCH          ;SET HAVE RESID CHAR FLAG
2683  49D5   C3   89   48        10.0          JMP     SCHREX          ;BACK TO THE CLOCK LOOP
```

```
2685  49D8                              S
 (1)                                    ;*****************************************************************************
2686                                    ;CRC -- ROUTINE TO SIMULATE THE CRC GENERATOR IN THE TM78.  CALLED
2687                                    ;       WITH ACCUMULATOR CONTAINING DATA BYTE TO BE ADDED TO CRCWORD
2688  49D8                              ;
 (1)                                    ;*****************************************************************************
2689
2690  49D8  CD  76  4A      18.0  GENERC: CALL    PARITY          ;CALC. PARITY ON DATA IN ACCUMULATOR
2691  49DB  F5                12.0        PUSH    PSW
2692  49DC  C5                12.0        PUSH    B
2693  49DD  E5                12.0        PUSH    H
2694  49DE  21  98  4A        10.0        LXI     H,CRCCHR        ;POINT TO CRC CHAR STORAGE
2695  49E1  AE                 7.0        XRA     M               ;XOR DATA IN ACCUMULATOR WITH PREVIOUS CRC 7 TO 0
2696  49E2  23                 6.0        INX     H               ;POINT TO CRC PARITY BIT STORAGE
2697  49E3  47                 4.0        MOV     B,A             ;SAVE RESULT OF XOR
2698  49E4  3A  A2  4A        13.0        LDA     PBIT            ;GET PARITY OF DATA BYTE ON CALL
2699  49E7  AE                 7.0        XRA     M               ;XOR WITH CRC PARITY BIT
2700  49E8  1F                 4.0        RAR                     ;PUT PARITY BIT IN 'C' BIT POSITION
2701  49E9  78                 4.0        MOV     A,B             ;GET XOR OF 7 TO 0
2702  49EA  1F                 4.0        RAR                     ;SHIFT RIGHT
2703  49EB  D2  F9  49        10.0        JNC     CRC2            ;COMPLIMENT BITS 5,4,3,+2 IF 'C' IS SET
2704
2705  49EE  EE  3C             7.0  CRC1:   XRI     @74             ;COMPLIMENT BITS
2706  49F0  32  98  4A        13.0        STA     CRCCHR          ;SAVE THE BYTE
2707  49F3  3E  01             7.0        MVI     A,1             ;AND SET THE 'P' BIT TO A 1 FOR NEXT TIME
2708  49F5  77                 7.0        MOV     M,A             ;SAVE THE 1
2709  49F6  C3  FE  49        10.0        JMP     CRC3            ;EXIT
2710
2711  49F9  32  98  4A        13.0  CRC2:   STA     CRCCHR          ;SAVE CRC CHARACTER
2712  49FC  AF                 4.0        XRA     A               ;CLEAR THE 'P' BIT
2713  49FD  77                 7.0        MOV     M,A             ;FOR THE NEXT TIME
2714  49FE  E1                10.0  CRC3:   POP     H
2715  49FF  C1                10.0        POP     B
2716  4A00  F1                10.0        POP     PSW
2717  4A01  C9                10.0        RET
2718
2719                                    ;INITIALIZE CRC CHARACTER STORAGE
2720
2721  4A02  AF                 4.0  CLCRC:  XRA     A
2722  4A03  32  98  4A        13.0        STA     CRCCHR
2723  4A06  32  9A  4A        13.0        STA     CRCCNT
2724  4A09  32  99  4A        13.0        STA     CRCCHR+1
2725  4A0C  C9                10.0        RET
2726
2727                                    ;TRANSFER COMPLETE... FINISH THE CRC GENERATION
2728
2729  4A0D  3A  98  4A        13.0  CRCFIN: LDA     CRCCHR          ;GET CRC WORD
2730  4A10  EE  D7             7.0        XRI     @327            ;COMPLIMENT BITS 7,6,4,2,1,+0
2731  4A12  32  98  4A        13.0        STA     CRCCHR
2732  4A15  C9                10.0        RET
```

```
                                        .SBTTL  CLEAR ACRC SUBROUTINE

                                        S
                                        ; ************************************************************************
                                        ;THIS SUBROUTINE CLEARS THE ACRC CHARACTER AND ACRC CHARACTER PARITY -
                                        ;MEMORY LOCATIONS "ACRCHR" AND "ACRCP" - USED BY THE ACRC SUBROUTINE.
                                        ;IT SHOULD BE CALLED PRIOR TO STARTING A DATA TRANSFER.
                                        S
                                        ; ************************************************************************

                                4.0     CLACRC: XRA     A               ;CLEAR THE ACCUMULATOR
                                13.0            STA     ACRCHR          ;CLEAR ACRC CHARACTER
                                13.0            STA     ACRCP           ;CLEAR ACRC PARITY
                                10.0            RET

                                        .SBTTL  CALCULATE ACRC CHARACTER

                                        S
                                        ; ************************************************************************
                                        ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND THE
                                        ;CONTENTS OF "ACRCHR" AND "ACRCP" TO UPDATE THE CONTENTS OF "ACRCHR"
                                        ;AND "ACRCP" ACCORDING TO THE ANSI STANDARD POLYNOMIAL.
                                        S
                                        ; ************************************************************************

                                12.0    GENACRC: PUSH   PSW             ;SAVE THE DATA BYTE IN REG A
                                12.0            PUSH    B               ;SAVE REGISTERS B&C
                                12.0            PUSH    D               ;SAVE REGISTERS D&E
                                12.0            PUSH    H               ;SAVE REGISTERS H&L
                                4.0             MOV     E,A             ;SAVE THE CHARACTER
                                18.0            CALL    PARITY          ;GENERATE ODD PARITY FOR CHARACTER
                                13.0            LDA     ACRCHR          ;GET THE PREVIOUS ACRC CHARACTER
                                4.0             XRA     E               ;XOR DATA AND PREVIOUS ACRC CHARACTER
                                4.0             MOV     E,A             ;SAVE XOR RESULT IN REG E

                                13.0            LDA     PBIT            ;GET THE CHARACTER PARITY BIT
                                4.0             MOV     D,A             ;SAVE IN REG D
                                13.0            LDA     ACRCP           ;GET THE ACRC PARITY BIT
                                4.0             XRA     D               ;XOR THE DATA PARITY BIT AND PREVIOUS
                                                                        ;ACRC PARITY BIT
                                4.0             MOV     D,A             ;SAVE XOR RESULT IN REG D
                                                                        ;
                                4.0             MOV     A,E             ;GET THE BYTE XOR FROM REG E
                                7.0             ANI     $08             ;BIT 3 SET?
                                4.0             MOV     A,E             ;GET THE BYTE XOR FROM REG E
                                10.0            JZ      ACRC1           ;NO-LEAVE RESULT ALONE
                                7.0             XRI     @140            ;ELSE-XOR RESULT WITH 140 - BITS 5 & 6

                                4.0     ACRC1:  MOV     E,A             ;STORE RESULT BACK IN REG. E
                                4.0             XRA     A               ;CLEAR A
                                4.0             MOV     C,A             ;CLEAR THE TRANSLATE RESULT
                                                                        ;
                                10.0            LXI     H,ACRCTB        ;POINT TO ACRC TRANSLATION TBL
                                18.0            CALL    TRANS           ;TRANSLATE THE BITS
```

```
2784                                                                    ;
2785  4A46  7B                  4.0          MOV     A,E                ;GET THE BYTE BEING TRANSLATED
2786  4A47  E6  02               7.0          ANI     $02                ;BIT 1 SET
2787  4A49  3E  00               7.0          MVI     A,$00              ;CLEAR A
2788  4A4B  CA  50  4A          10.0          JZ      ACRC2              ;GO CLEAR ACRC P BIT IF ZERO
2789  4A4E  3E  01               7.0          MVI     A,$01              ;ELSE-SET ACRC P BIT
2790  4A50  32  A1  4A          13.0  ACRC2:  STA     ACRCP              ;MODIFY ACRC P BIT
2791                                                                    ;
2792  4A53  7A                   4.0          MOV     A,D                ;GET THE P BIT XOR RESULT
2793  4A54  A7                   4.0          ANA     A                  ;SET THE CONDITION CODE
2794  4A55  CA  5C  4A          10.0          JZ      ACRC3              ;CONTINUE IF ZERO
2795  4A58  79                   4.0          MOV     A,C                ;ELSE-GET TRANSLATION RESULTS
2796
2797  4A59  F6  01               7.0          ORI     $01                ;OR IN TRANSLATION BIT
2798  4A5B  4F                   4.0          MOV     C,A                ;SAVE RESULTS IN C
2799  4A5C  79                   4.0  ACRC3:  MOV     A,C                ;GET THE TRANSLATED BYTE
2800  4A5D  32  A0  4A          13.0          STA     ACRCHR             ;STORE IN ACRC CHARACTER
2801  4A60  E1                  10.0          POP     H                  ;RESTORE REGISTERS H&L
2802  4A61  D1                  10.0          POP     D                  ;RESTORE REGISTERS D&E
2803  4A62  C1                  10.0          POP     B                  ;RESTORE REGISTERS B&C
2804  4A63  F1                  10.0          POP     PSW                ;RESET THE ORRIG. DATA BYTE IN REG A
2805  4A64  C9                  10.0          RET                        ;RETURN
2806
2807  4A65  40                         ACRCTB: $40                       ;BIT 0 = POSITION 6
2808  4A66  00                                 $00                       ;BIT 1 = NO REPLACEMENT
2809  4A67  20                                 $20                       ;BIT 2 = POSITION 5
2810  4A68  04                                 $04                       ;BIT 3 = POSITION 2
2811  4A69  80                                 $80                       ;BIT 4 = POSITION 7
2812  4A6A  02                                 $02                       ;BIT 5 = POSITION 1
2813  4A6B  10                                 $10                       ;BIT 6 = POSITION 4
2814  4A6C  08                                 $08                       ;BIT 7 = POSITION 3
```

```
                                        .SBTTL  FORMAT FINAL ACRC CHARACTER

4A6D                                    S
 (1)                                    ; ************************************************************************
                                        ;THIS SUBROUTINE TAKES THE CONTENTS OF "ACRCHR" AND "ACRCP" AND
                                        ;TRANSFORMS THEM INTO A 8 BIT ACRC CHARACTER.  THE FINAL ACRC CHARACTER
                                        ;IS AVAILABLE IN "ACRCHR".
4A6D                                    S
 (1)                                    ; ************************************************************************

4A6D  3A  A0  4A        13.0  ACRCFI: LDA     ACRCHR          ;GET THE ACRC CHAR
4A70  EE  BC             7.0          XRI     @274            ;INVERT THE PROPER BITS
4A72  32  A0  4A        13.0          STA     ACRCHR          ;STORE THE FINAL CHARACTER
4A75  C9                10.0          RET                     ;RETURN

                                        .SBTTL  PARITY CALCULATION POUTINE

                                        ;THIS SUBROUTINE CALCULATES ODD PARITY ON THE CONTENTS OF THE
                                        ;ACCUMULATOR AND SETS OR CLEARS THE MEMORY LOCATION "PBIT AS REQUIRED.

4A76  F5                12.0  PARITY: PUSH    PSW             ;SAVE ACCUMULATOR
4A77  A7                 4.0          ANA     A               ;SET CONDITION BITS
4A78  3E  00             7.0          MVI     A,$00           ;CLEAR ACCUMULATOR
4A7A  E2  7F  4A        10.0          JPO     PAR1            ;SKIP IF PARITY ODD
4A7D  3E  01             7.0          MVI     A,$01           ;ELSE-LOAD ACC. WITH 1
4A7F  32  A2  4A        13.0  PAR1:   STA     PBIT            ;SET UP THE P BIT
4A82  F1                10.0          POP     PSW             ;RESTORE ACCUMULATOR
4A83  C9                10.0          RET                     ;RETURN

                                        ;HERE IS THE NECESSARY STORAGE FOR TEST PARAMETERS AND DDR DATA STORAGE

4A84  00                                CCTLWD: .BYTE   0       ;IMAGE OF CLOCK CONTROL WORD
4A85  00                                IMBYTC: .BYTE   0       ;IMAGE OF BYTE COUNTER 7-0 BITS

4A86  00                                ABCNT:  .BYTE   0       ;ASSEMBLED BYTE COUNTER
4A87  00                                ABCNTR: .BYTE   0
4A88  00                                EABCNT: .BYTE   0       ;BYTE COUNT AT A DETECTED ERROR

4A89  00                                FORMT:  .BYTE   0       ;FORMAT CODE SELECTED FOR TEST
4A8A  00                                DPNUM:  .BYTE   0       ;DATA PATTERN OFFSET NUMBER (0 TO 17)
4A8B  00                                CLKON:  .BYTE   0       ;'SCLK' ON FLAG-USED TO TELL WHEN TO FILL DDR REGS
4A8C  00                                ECCCNT: .BYTE   0       ;COUNT BY 7 FOR ECC CHARACTER CHECK
4A8D  00                                CLKCNT: .BYTE   0       ;COUNT OF SCLKS GENERATED

4A8E  00                                SDDRA:  .BYTE   0       ;DDRA DATA ACTUALLY LOADED
4A8F  00                                SDDRB:  .BYTE   0       ;DDRB DATA ACTUALLY LOADED
4A90  00                                SDDRC:  .BYTE   0       ;DDRC DATA ACTUALLY LOADED

4A91  00                                SDDRAT: .BYTE   0       ;DDRA DATA FROM PGM TABLE
4A92  00                                SDDRBT: .BYTE   0       ;DDRB DATA FROM PGM TABLE
4A93  00                                SDDRCT: .BYTE   0       ;DDRC DATA FROM PGM TABLE

4A94  00                                SWMCSTA: .BYTE  0       ;COPY OF THE WMC STATUS AT 'XFER' BIT TIME
4A95  00                                SXFER:  .BYTE   0       ;SOFTWARE 'XFER' FLAG
```

```
2868  4A96  00                SINTSTA: .BYTE  0          ;SOFTWARE COPY OF INTSTA WORD
2869
2870  4A97  00                ECCCHR: .BYTE   0          ;ECC CHARACTER CALCULATED BY THE SOFTWARE
2871  4A98  00    00          CRCCHR: .WORD   0          ;IMAGE OF THE CRC CHARACTER
2872  4A9A  00                CRCCNT: .BYTE   0          ;# CRC CHAR RECEIVED
2873  4A9B  00                RCHRIM: .BYTE   0          ;IMAGE OF 'RESIDUAL' CHAR. TO BE SEEN ON WMC WHEN 'RESID
2874  4A9C  00                HAVCRC: .BYTE   0          ;RECEIVED 'CRC' CHAR
2875  4A9D  00                HAVACRC: .BYTE  0          ;RECEIVED 'ACRC' CHAR
2876  4A9E  00                HAVRCH: .BYTE   0          ;FLAG TO INDICATE 'RESID CHAR' RECEIVED
2877  4A9F  00                RPADCT: .BYTE   0          ;# RECEIVED PAD CHAR
2878  4AA0  00                ACRCHR: .BYTE   0          ;ACRC CHARACTER
2879  4AA1  00                ACRCP:  .BYTE   0          ;ACRC CHARACTER PARITY
2880  4AA2  00                PBIT:   .BYTE   0
2881  4AA3  00                RESCNT: .BYTE   0          ;RESIDUAL CHARACTERS COUNT
2882  4AA4  00                IMRESCNT: .BYTE 0          ;SOFTWARE COPY OF RESIDUAL COUNT GIVEN TO THE WMC
2883  4AA5  00                CLKCRC: .BYTE   0          ;EXTRA CLOCK-TO-CRC FLAG
2884  4AA6  00                IMPADCT: .BYTE  0          ;SOFTWARE COPY OF THE PAD COUNT GIVEN TO THE WMC
2885  4AA7  00                TPBCNT: .BYTE   0          ;TEST PASS BYTE COUNTER
2886  4AA8  00                PDCRC:  .BYTE   0          ;PAD/CRC FLAG IF SET, WMC WILL ISSUE A PAD
2887  4AA9  00                LASTCRC: .BYTE  0          ;LAST CRC BYTE EXPECTED FLAG
```

```
2889  4AAA                              S
 (1)                                    ; ***************************************************************************
2890                                    ;GENDAT -- ROUTINE TO GENERATE DDR REG DATA FOR NEXT "SCLK" CYCLE.
2891                                    ;       DATA IS LOADED IN DDR REGS AND A COPY IS STORED IN 'SDDRAC', +
2892                                    ;       'SDDRCC' FOR REFERENCE.
2893  4AAA                              S
 (1)                                    ; ***************************************************************************
2894
2895  4AAA  F5              12.0  GENDAT: PUSH    PSW             ;SAVE PSW + A
2896  4AAB  C5              12.0          PUSH    B               ;SAVE B + C
2897  4AAC  E5              12.0          PUSH    H               ;SAVE H + L
2898  4AAD  3A  8A  4A      13.0          LDA     DPNUM           ;GET THE LAST PATTERN # USED
2899  4AB0  3C               4.0          INR     A               ;POINT TO NEXT
2900  4AB1  32  8A  4A      13.0          STA     DPNUM           ;SAVE NUMBER
2901  4AB4  4F               4.0          MOV     C,A
2902  4AB5  06  00           7.0          MVI     B,0             ;B + C HAVE OFFSET
2903  4AB7  21  EF  4A      10.0          LXI     H,CDATBL        ;POINT TO DATA TABLE
2904  4ABA  09              10.0          DAD     B               ;ADD IN OFFSET TO H + L
2905
2906  4ABB  7E               7.0          MOV     A,M             ;GET DATA BYTE FOR DDR 'A'
2907  4ABC  32  91  4A      13.0          STA     SDDRAT          ;SAVE DDR 'A' IMAGE
2908  4ABF  AF               4.0          XRA     A               ;
2909  4AC0  32  92  4A      13.0          STA     SDDRBT          ;SAVE DDR 'B' IMAGE
2910  4AC3  32  93  4A      13.0          STA     SDDRCT          ;SAVE DDR 'C' IMAGE
2911
2912  4AC6  E1              10.0  GENOUT: POP     H               ;RESTORE REGS
2913  4AC7  C1              10.0          POP     B
2914  4AC8  F1              10.0          POP     PSW
2915  4AC9  C9              10.0          RET
```

```
2917  4ACA                                  S
 (1)                                        ; ************************************************************************
2918                                        ;SETDDR -- ROUTINE TO TRANSLATE THE DESIRED DATA FROM A 1-FOR-1 BIT
2919                                        ;       POSITION TO THE DDR TO BE USED AS DATA TO EITHER THE TM78 OR
2920                                        ;       THE MASSBUS. WMC "LEFT" FLAG SET INDICATES THAT THIS ROUTINE IS NOT
2921                                        ;       NECESSARY, THE DATA IS IN THE CORRECT FORMAT ALREADY. IF WMC
2922                                        ;       "LEFT" FLAG IS NOT SET, THE DATA MUST BE TRANSLATED TO A COMPATABLE
2923                                        ;       DDR FORMAT.
2924  4ACA                                  S
 (1)                                        ; ************************************************************************
2925
2926                                        ;HERE TO TRANSLATE THE DATA INTO AN ACCEPTABLE FORMAT FOR DDR INJECTION
2927
2928  4ACA  3A  91  4A        13.0  SETDDR: LDA     SDDRAT          ;GET DDRA TEMP
2929  4ACD  E6  3F              7.0         ANI     @77             ;STRIP 2 BITS OFF
2930  4ACF  07                  4.0         RLC
2931  4AD0  07                  4.0         RLC                     ;LEFT JUSTIFY
2932  4AD1  47                  4.0         MOV     B,A             ;TEMP SAVE
2933  4AD2  3A  93  4A        13.0          LDA     SDDRCT          ;GET DDRC TEMP
2934  4AD5  E6  03              7.0         ANI     @3              ;SAVE 2 BITS
2935  4AD7  B0                  4.0         ORA     B
2936  4AD8  32  8E  4A        13.0          STA     SDDRA           ;SAVE DDRA DESIRED
2937  4ADB  D3  D8             10.0         OUT     DDRA            ;LOAD THE REAL DDRA
2938
2939  4ADD  3A  91  4A        13.0          LDA     SDDRAT          ;GET DDRA AGAIN
2940  4AE0  E6  C0              7.0         ANI     @300            ;SAVE 2 BITS
2941  4AE2  07                  4.0         RLC
2942  4AE3  07                  4.0         RLC
2943  4AE4  32  8F  4A        13.0          STA     SDDRB           ;SAVE DDRB DATA
2944  4AE7  D3  D9             10.0         OUT     DDRB            ;LOAD THE REAL DDRB
2945
2946  4AE9  3A  90  4A        13.0          LDA     SDDRC
2947  4AEC  D3  DA             10.0         OUT     DDRC
2948  4AEE  C9                 10.0         RET                     ;EXIT
```

```
2950  4AEF  F7                    CDATBL: .BYTE   @367            ;1ST SET OF DATA HAS KNOWN RESULTS
2951  4AF0  FB                            .BYTE   @373
2952  4AF1  00                            .BYTE   @0
2953  4AF2  23                            .BYTE   @43
2954  4AF3  88                            .BYTE   @210
2955  4AF4  C0                            .BYTE   @300
2956  4AF5  63                            .BYTE   @143            ;EXPECTED 'ECC' IS @311
2957                                                              ;EXPECTED INTERMEDIATE 'CRC' IS @142
2958                                                              ;EXPECTED INTERMEDIATE 'ACRC' IS @144
2959
2960  4AF6  12                            .BYTE   $12             ;HEX 12
2961  4AF7  23                            .BYTE   $23             ;HEX 23
2962  4AF8  34                            .BYTE   $34             ;HEX 34
2963  4AF9  45                            .BYTE   $45             ;HEX 45
2964  4AFA  56                            .BYTE   $56             ;HEX 56
2965  4AFB  67                            .BYTE   $67             ;HEX 67
2966  4AFC  78                            .BYTE   $78             ;HEX 78
2967  4AFD  89                            .BYTE   $89             ;HEX 89
2968  4AFE  9A                            .BYTE   $9A             ;HEX 9A
2969  4AFF  AB                            .BYTE   $AB             ;HEX AB
2970  4B00  BC                            .BYTE   $BC             ;HEX BC
2971  4B01  CD                            .BYTE   $CD             ;HEX CD
2972  4B02  DE                            .BYTE   $DE             ;HEX DE
2973  4B03  EF                            .BYTE   $EF             ;HEX EF
2974  4B04  F0                            .BYTE   $F0             ;HEX F0
2975  4B05  01                            .BYTE   $01             ;HEX 01
2976  4B06  02                            .BYTE   $02             ;FLOATING 1'S
2977  4B07  04                            .BYTE   $04
2978  4B08  08                            .BYTE   $08
2979  4B09  10                            .BYTE   $10
2980  4B0A  20                            .BYTE   $20
2981  4B0B  40                            .BYTE   $40
2982  4B0C  80                            .BYTE   $80
2983  4B0D  FE                            .BYTE   $FE             ;FLOATING 0'S
2984  4B0E  FD                            .BYTE   $FD
2985  4B0F  FB                            .BYTE   $FB
2986  4B10  F7                            .BYTE   $F7
2987  4B11  EF                            .BYTE   $EF
2988  4B12  DF                            .BYTE   $DF
2989  4B13  BF                            .BYTE   $BF
2990  4B14  7F                            .BYTE   $7F
2991  4B15                        DTEND:                          ;END OF THE DATA TABLE
2992
2993                                      ;ROUTINE TO SHIFT CONTENTS OF ACCUMULATOR RIGHT OR LEFT EITHER 2 OR 4 TIMES
2994
2995  4B15  E6  FC          7.0   SHR4:   ANI     @374            ;STRIP OFF 2 BITS
2996  4B17  0F              4.0           RRC                     ;SHIFT RIGHT 2 TIMES
2997  4B18  0F              4.0           RRC
2998  4B19  E6  FC          7.0   SHR2:   ANI     @374            ;STRIP OFF 2 BITS
2999  4B1B  0F              4.0           RRC                     ;SHIFT RIGHT 2 TIMES
3000  4B1C  0F              4.0           RRC
3001  4B1D  C9             10.0           RET
3002
3003  4B1E  E6  3F          7.0   SHL4:   ANI     @77             ;SHIFT LEFT TWICE
```

```
3004  4B20  07                  4.0          RLC
3005  4B21  07                  4.0          RLC
3006  4B22  E6  3F              7.0  SHL2:   ANI     @77
3007  4B24  07                  4.0          RLC                     ;SHIFT LEFT TWICE
3008  4B25  07                  4.0          RLC
3009  4B26  C9                 10.0          RET
3010
3011
3012  4B27                                   S
 (1)                                         ; *****************************************************************************
3013                                         ;CLKSYS -- ROUTINE TO CLOCK THE SYSTEM 1 TIME BY WRITING CLOCK CONTROL
3014                                         ;       'CLK' BIT TO A '1' THEN A '0'.
3015  4B27                                   S
 (1)                                         ; *****************************************************************************
3016
3017  4B27  3A  84  4A          13.0  CLKSYS: LDA     CCTLWD            ;GET SOFTWARE CLOCK CONTROL IMAGE
3018  4B2A  F6  40               7.0          ORI     SSCLK             ;ADD IN 'CLK' BIT
3019  4B2C  D3  F0              10.0          OUT     CLKCTL            ;LOAD CLOCK CONTROL
3020  4B2E  E6  3F               7.0          ANI     @77               ;STRIP OFF CLOCK BIT
3021  4B30  D3  F0              10.0          OUT     CLKCTL            ;LOAD CLOCK CONTROL WORD
3022  4B32  C9                  10.0          RET                       ;EXIT - CLOCK CYCLE COMPLETE
```

```
3024                                    .SBTTL  CLEAR ECC SUBROUTINE
3025  4B33                              S
 (1)                                    ; ************************************************************************
3026                                    ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION "ECCCHR" -
3027                                    ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
3028                                    ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
3029  4B33                              S
 (1)                                    ; ************************************************************************
3030
3031  4B33  F5              12.0  CLECC:  PUSH    PSW             ;SAVE THE ACCUMULATOR
3032  4B34  AF               4.0          XRA     A               ;CLEAR THE ACCUMULATOR
3033  4B35  32  97  4A      13.0          STA     ECCCHR          ;CLEAR THE ECC CHARACTER
3034  4B38  32  8C  4A      13.0          STA     ECCCNT          ;CLEAR THE COUNT OF ASSEMBLED BYTES
3035  4B3B  F1              10.0          POP     PSW             ;RESTORE THE ACCUMULATOR
3036  4B3C  C9              10.0          RET                     ;RETURN TO USER
3037
3038                                          .SBTTL  CALCUALTE ECC CHARACTER
3039  4B3D                              S
 (1)                                    ; ************************************************************************
3040                                    ;GENECC --THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
3041                                    ;THE CONTENTS OF "ECCCHR" TO UPDATE THE CONTENTS OF "ECCCHR" ACCORDING
3042                                    ;TO THE ANSI STANDARD ECC POLYNOMIAL.
3043  4B3D                              S
 (1)                                    ; ************************************************************************
3044
3045  4B3D  F5              12.0  GENECC: PUSH    PSW             ;SAVE THE ACCUMULATOR
3046  4B3E  C5              12.0          PUSH    B               ;SAVE B&C
3047  4B3F  D5              12.0          PUSH    D               ;SAVE D&E
3048  4B40  E5              12.0          PUSH    H               ;SAVE H&L
3049  4B41  21  97  4A      10.0          LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
3050  4B44  AE               7.0          XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
3051  4B45  5F               4.0          MOV     E,A             ;SAVE XOR RESULT IN E
3052  4B46  E6  10           7.0          ANI     $10             ;IS BIT #4 OF RESULT SET
3053  4B48  7B               4.0          MOV     A,E             ;RESTORE XOR RESULT FROM B
3054  4B49  CA  4E  4B      10.0          JZ      ECC1            ;CONTINUE IF BIT #4 RESET
3055  4B4C  EE  23           7.0          XRI     $23             ;ELSE-XOR WITH 23
3056  4B4E  5F               4.0  ECC1:   MOV     E,A             ;STORE THE ECC RESULT IN E
3057  4B4F  AF               4.0          XRA     A               ;CLEAR A
3058  4B50  4F               4.0          MOV     C,A             ;CLEAR THE TRANSLATE RESULT
3059  4B51  21  60  4B      10.0          LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
3060  4B54  CD  68  4B      18.0          CALL    TRANS           ;TRANSLATE THE BITS
3061  4B57  79               4.0          MOV     A,C             ;GET THE TRANSLATED RESULT
3062  4B58  32  97  4A      13.0          STA     ECCCHR          ;STORE RESULT
3063  4B5B  E1              10.0          POP     H               ;RESTORE H&L
3064  4B5C  D1              10.0          POP     D               ;RESTORE D&E
3065  4B5D  C1              10.0          POP     B               ;RESTORE B&C
3066  4B5E  F1              10.0          POP     PSW             ;RESTORE ACCUM.
3067  4B5F  C9              10.0          RET
```

```
3069  4B60  08                        ECCTBL: $08                     ;BIT 0 = POSITION 3
3070  4B61  20                                $20                     ;BIT 1 = POSITION 5
3071  4B62  02                                $02                     ;BIT 2 = POSITION 1
3072  4B63  40                                $40                     ;BIT 3 = POSITION 6
3073  4B64  80                                $80                     ;BIT 4 = POSITION 7
3074  4B65  01                                $01                     ;BIT 5 = POSITION 0
3075  4B66  10                                $10                     ;BIT 6 = POSITION 4
3076  4B67  04                                $04                     ;BIT 7 = POSITION 2
3077
3078
3079                                          .SBTTL  POLYNOMIAL BIT TRANSLATION
3080
3081                                    ;
3082                                    ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
3083                                    ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
3084                                    ;
3085
3086  4B68  06  01            7.0   TRANS:  MVI     B,1                 ;INIT 'B' TO BIT POSITION 0
3087  4B6A  7B                4.0   TRANS1: MOV     A,E                 ;GET CHAR TO BE TRANSLATED
3088  4B6B  A0                4.0           ANA     B                   ;SEE IF BIT POSITION IN 'B' IS SET
3089  4B6C  CA  72  4B       10.0           JZ      TRANS2              ;DO NEXT BIT POSITION IF NOT SET
3090  4B6F  79                4.0           MOV     A,C                 ;GET PREVIOUS RESULT OF 'OR'
3091  4B70  B6                7.0           ORA     M                   ;'OR' IN NEW POSITION
3092  4B71  4F                4.0           MOV     C,A                 ;SAVE RESULT
3093
3094  4B72  78                4.0   TRANS2: MOV     A,B
3095  4B73  07                4.0           RLC                         ;POSITION MASK TO NEXT BIT
3096  4B74  47                4.0           MOV     B,A
3097  4B75  D8               12.0           RC                          ;EXIT WHEN ALL POSITIONS DONE
3098  4B76  23                6.0           INX     H                   ;POINT TO NEXT TABLE ENTRY
3099  4B77  C3  6A  4B       10.0           JMP     TRANS1              ;PROCESS NEXT BIT
3100        0000                     .END
```

```
A      =%0007          ABCNT   4A86          ABCNTR  4A87          ACRCEC  487A  G
ACRCEE  4880           ACRCFI  4A6D          ACRCHR  4AA0          ACRCOC  4868  G
ACRCOE  486E           ACRCP   4AA1          ACRCPE  486F          ACRCPO  485B
ACRCTB  4A65           ACRC1   4A3D          ACRC2   4A50          ACRC3   4A5C
ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002          ARAIDF= 0098
ASAVE   4F9B           ATTCD   4F97          AXNUM   4F91          B      =%0000
BADST  = 0090          BIT0   = 0001          BIT1   = 0002          BIT15  = 8000
BIT2   = 0004          BIT3   = 0008          BIT4   = 0010          BIT5   = 0020
BIT6   = 0040          BIT7   = 0080          BIT8   = 0100          BIT9   = 0200
BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60          BRKXCT= 4F00
BSAVE   4F9C           BYTCNT= 00D4          BYTEH   4F24          BYTEL   4F23
C      =%0001          CACRC1  4881          CASCT   4F21          CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1
CBYTH  = 008B          CBYTL  = 008A          CCTLWD  4A84          CDATBL  4AEF
CDG1H  = 0087          CDG1L  = 0086          CDG2H  = 0093          CDG2L  = 0092
CDG3H  = 0095          CDG3L  = 0094          CDVTH  = 008D          CDVTL  = 008C
CHPTIE= 0028           CHOTIE= 0020          CH1TIE= 0021          CH2TIE= 0022
CH3TIE= 0023           CH4TIE= 0024          CH5TIE= 0025          CH6TIE= 0026
CH7TIE= 0027           CKACRC  47B1          CKAC4   4804          CKAC4C  481A  G
CKA1    47BD  G        CKA2    47CF  G       CKA3    47E1  G       CKA3A   47E7
CKA3B   47F2           CKA3C   47FE  G       CKA4    483D  G       CKA5    4820
CKCRC   488F           CKCRC1  489D          CKCRC2  4934          CKC1    48A9  G
CKC2    48BB  G        CKECC   4732          CKECCC  4742  G       CKECC1  47AB
CKHACK  45CC           CKLOP  = 2815          CKPAD   4551          CKPADA  455D
CKPAD1  45C1           CKRESI  493C          CKRES1  49CB          CKR1    4948  G
CKR1A   4959           CKR2    496C  G       CKR5    4987  G       CKSCLK  4473
CKSC1   4482           CLACRC  4A16          CLCRC   4A02          CLECC   4B33
CLKCNT  4A8D           CLKCRC  4AA5          CLKCTL= 00F0          CLKON   4A8B
CLKSYS  4B27           CLOCK   4F26          CMCOH  = 0099          CMCOL  = 0098
CMC1H  = 009B          CMC1L  = 009A          CMC2H  = 009D          CMC2L  = 009C
CMC3H  = 009F          CMC3L  = 009E          CMINH  = 0097          CMINL  = 0096
CNTCTL= 00D7           CRCCHR  4A98          CRCCNT  4A9A          CRCFIN  4A0D
CRCPE   4922           CRCPEC  492D  G       CRCPEE  4933          CRCPO   490E
CRCPOC  491B  G        CRCPOE  4921          CRCWRD= 0018          CRC1    49EE
CRC2    49F9           CRC2A   48CC          CRC3    49FE          CRC4    48DB
CRC5    48F0  G        CSAVE   4F9D          CSRLH  = 0091          CSRLL  = 0C90
CTCH   = 0085          CTCL   = 0084          CTSTH  = 008F          CTSTL  = 008E
CXCTH  = 0081          CXCTL  = 0080          CXINH  = 0083          CXINL  = 0082
C.     = 0001          C.AVAI= 0080          C.DP   = 0008          C.DSE  = 0010
C.DTU  = 0003          C.DVA  = 0008          C.FAIL= 00FC          C.FMT  = 0070
C.FNCT= 003E           C.GO   = 0001          C.INTC= 00FE          C.MAIN= 0020
C.NSA  = 0080          C.RCT  = 00FC          C.SER  = 0080          C.SHR  = 0040
C.SKPC= 000F           C.TAPE= 0040          C.WCS  = 0002          D      =%0002
DATACT= 00D0           DATPE   4664          DATPEC  466F  G       DATPEE  4675
DATPO   4650           DATPOC  465D  G       DATPOE  4663          DBUS    4F28
DBUSCT= 00C0           DBUSST= 00C0          DDRA   = 00D8          DDRAIN= 0010
DDRB   = 00D9          DDRBIN= 0002          DDRC   = 00DA          DDRCIN= 0001
DDRCO  = 0088          DDRCTL= 00DB          DE_OOP  4717          DELP1   4724
DIAFLG  4F22           DIAGPG= 4300          DIAGRM= 4F90          DIARD  = 000B
DONE1  = 0045          DONINT= 0010          DPNUM   4A8A          DSAVE   4F9E
DSE    = 0006          DTEND   4B15          D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010           D.LAGC= 0020          D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008           D.WR4  = 0080          E      =%0003          EABCNT  4A88
ECCBAD= 0042           ECCCHR  4A97          ECCCK   4748          ECCCK1  475D  G
```

```
ECCCK2  476D          ECCCNT  4A8C          ECCCOR= 0019          ECCOK = 0041
ECCPE   4799          ECCPEC  47A4  G       ECCPEE  47AA          ECCPO   4785
ECCPOC  4792  G       ECCPOE  4798          ECCSTA= 001A          ECCTBL  4B60
ECCTST= 000E          ECC1    4B4E          EDATA = 0095          EOTCLR= 0003
ERFLG   4F93          ERLP  = 2809          ERLPA = 280F          ERLPB = 2812
ERLPE = 280C          ERNUM   4F90          ERRCNT= 00D6          ESAVE   4F9F
E.ACRC= 0010          E.AMT = 0020          E.CDP = 0080          E.CRC = 0080
E.PNTR= 0008          E.RPE = 0040          E.STEC= 0001          E.TTEC= 0002
E.UNC = 0004          FIFORD= 006A          FORMAT  4F25          FORMT   4A89
FWDTST= 0061          GCRID = 0089          GCRSET= 0002          GENACR  4A1E
GENCRC  49D8          GENDAT  4AAA          GENECC  4B3D          GENOUT  4AC6
GOODTM= 0092          H     =%0004          HAVACR  4A9D          HAVCRC  4A9C
HAVRCH  4A9E          HLSAVE  4FA0          IE    = 0008          IMBYTC  4A85
IMPADC  4AA6          IMRESC  4AA4          INTSTA= 00E0          ITERA   4F9A
I.PWR = 0020          I.RMPE= 0040          I5.5  = 0010          I6.5  = 0020
I7.5  = 0040          KCALL = 005F          KCLR  = 007B          KDEP  = 003F
KENAB = 0078          KEXAM = 003E          KEYBRD= 00C8          KEY1  = 0078
KEY10 = 006D          KEY11 = 006E          KEY12 = 006F          KEY13 = 005C
KEY14 = 005D          KEY15 = 005E          KEY16 = 005F          KEY17 = 003C
KEY18 = 003D          KEY19 = 003E          KEY2  = 0079          KEY20 = 003F
KEY3  = 007A          KEY4  = 007B          KEY5  = 0074          KEY6  = 0075
KEY7  = 0076          KEY8  = 0077          KEY9  = 006C          KINTA = 006F
KLDAD = 003D          KN0   = 003C          KN1   = 005C          KN2   = 005D
KN3   = 005E          KN4   = 006C          KN5   = 006D          KN6   = 006E
KN7   = 0074          KN8   = 0075          KN9   = 0076          KU2   = 0079
KU3   = 007A          KU8   = 0077          L     =%0005          LASTCR  4AA9
LBLANK= 000F          LCE   = 000B          LCH   = 000C          LCL   = 000D
LCP   = 000E          LC0   = 0000          LC1   = 0001          LC2   = 0002
LC3   = 0003          LC4   = 0004          LC5   = 0005          LC6   = 0006
LC7   = 0007          LC8   = 0008          LC9   = 0009          LDLEDA= 00CA
LDLEDB= 00CB          LDLEDC= 00CC          LDLEDD= 00CD          LDLEDE= 00CE
LDLEDF= 00CF          LKDIAG= 2800          LKKBD = 004C          LKKEY = 0049
LKLWMG= 0058          LKLWMP= 0055          LKLWPG= 0052          LKLWPP= 004F
LKMOD7= 0046          LKOPR = 0046          LPFLG   4F94          LPNUM   4F92
M     =%0006          MBSEL = 00E0          MB.A  = 0008          MB.B  = 0004
MEMTOP= 4FFF          MINUS = 000A          MM    = 8000          MSE   = 0008
MSGN  = 0028          MTACLR= 0000          MT.ARA= 0020          MT.CPE= 0080
MT.DSE= 0001          MT.FWD= 0040          MT.INH= 0008          MT.LWR= 0004
MT.MOT= 0002          MT.NWT= 0080          MT.PEC= 0040          MT.PSB= 0004
MT.PS0= 0001          MT.PS1= 0002          MT.REV= 0020          MT.WRT= 0010
MT.Z  = 0008          M.ATA = 0080          M.CAPE= 0020          M.CONT= 0080
M.DEM = 0020          M.EBL = 0010          M.EXC = 0008          M.FAIL= 0008
M.ILR = 0010          M.INIT= 0010          M.OCC = 0020          M.ONLI= 0001
M.PE  = 0040          M.PORT= 0080          M.RDEN= 0002          M.RDPE= 0008
M.RUN = 0004          M.SCLK= 0001          M.TRA = 0040          M.UNIT= 0007
M.WCLK= 0040          M.WCLN= 0080          M.WREN= 0080          M5.5  = 0001
M6.5  = 0002          M7.5  = 0004          NOTCAP= 0088          OKAY  = 00FF
OPRRAM= 4300          OPSTRT= 0058          OPVER = 0040          PADCNT= 00D5
PADCNC  457D  G       PADCRC= 0080          PADPE   45AF          PADPEC  45BA  G
PADPEE  45C0          PADPO   459B          PADPOC  45A8  G       PADPOE  45AE
PARITY  4A76          PAR1    4A7F          PBIT    4AA2          PDCRC   4AA8
PDIAG = 0048          PEID  = 008A          PENAB = 004C          PESET = 0001
PL    = 00B1          PRDD  = 004C          PRENF = 009C          PS    = 00B2
PSTAT = 0048          PSW   =%0009          P.AMTP= 0001          P.BCTC= 0040
```

```
P.CMDP= 0020          P.INTE= 0080          P.LCS = 0040          P.LWR = 0020
P.RDP = 0002          P.RPEN= 0004          P.RPST= 0002          P.RPOE= 0020
P.RP1E= 0010          P.RP2E= 0020          P.RP3E= 0010          P.SING= 0080
P.STAT= 0002          P.STPE= 0080          P.TACH= 0008          P.TUPR= 0010
P.WCSP= 0004          P.WDS = 0040          P.WFLP= 0001          P.WPEN= 0010
P.WPOE= 0008          P.WP1E= 0004          P.WP2E= 0008          P.WP3E= 0004
P.5VOK= 0002          QUE   = 281B          QUEM  = 281E          QUIT    46A8
QUIT1   46BD  G       QUIT2   46DD  G       QUIT3   46F2  G       QUIT4   46E3
RAMT  = 0010          RARA  = 0006          RARAI = 0004          RCHBD0= 0048
RCHBD1= 0047          RCHOK = 0046          RCHRIM  4A9B          RCHTST= 000C
RCLRT = 000D          RCMD  = 000B          RCMLP = 0003          RCONT = 0080
RDATA = 0017          RDCLK = 0010          RDON  = 0011          READG = 0007
REND  = 0014          REQST = 2806          RESCHR= 00D1          RESCNT  4AA3
RESPE   49B9          RESPEC  49C4  G       RESPEE  49CA          RESPO   49A5
RESPOC  49B2  G       RESPOE  49B8          REVTST= 0064          REWIND= 0004
RFIFOL= 0008          RGCLK = 0002          RGCRI = 0003          RIBG  = 0001
RILL  = 0012          RINST = 000C          RMCTST= 0008          RMK2  = 0013
RNOP  = 0000          RPADCT  4A9F          RPATH = 0001          RPBAD = 0044
RPCHI = 0001          RPCLK = 0003          RPCTL = 0009          RPEI  = 0002
RPFAIL= 0000          RPF1  = 009D          RPF2  = 009E          RPOK  = 0043
RPOSTN= 0016          RPSTA = 0015          RRCMT = 000A          RSTAT = 0002
RTIEB = 000A          RTIER = 0030          RTM   = 0005          RUNKI = 0009
RUPTST= C05E          RWDUNL= 0005          R.AMT = 0001          R.BOP = 0008
R.DATA= 0040          R.DON = 0002          R.DRDY= 0010          R.END = 0010
R.ILL = 0004          R.JVOK= 0004          R.MK2 = 0008          R.PLOD= 0008
R.PLOO= 0010          R.PLO1= 0020          R.POST= 0020          R.STNM= 0002
R.STOP= 0004          R.STPC= 0001          R.TBJN= 0080          R.TSTD= 0040
R.VOK = 0080          R00H  = 0081          R00L  = 0080          R01H  = 0083
R01L  = 0082          R02H  = 0085          R02L  = 0084          R03H  = 0087
R03L  = 0086          R04H  = 0089          R04L  = 0088          R05H  = 008B
R05L  = 008A          R06H  = 008D          R06L  = 008C          R07H  = 008F
R07L  = 008E          R10H  = 0091          R10L  = 0090          R11H  = 0093
R11L  = 0092          R12H  = 0095          R12L  = 0094          R13H  = 0097
R13L  = 0096          R14H  = 0099          R14L  = 0098          R15H  = 009B
R15L  = 009A          R16H  = 009D          R16L  = 009C          R17H  = 009F
R17L  = 009E          R7.5  = 0010          SCHREX  4889          SCKB1   4465
SCKGON  445E          SCKWT   4436          SCKWTC  4453  G       SCKWTE  445C
SCKWT1  4443          SCLKBI  442B          SDDRA   4A8E          SDDRAT  4A91
SDDRB   4A8F          SDDRBT  4A92          SDDRC   4A90          SDDRCT  4A93
SELCLR= 0000          SETATA= 00A1          SETDDR  4ACA          SHL2    4B22
SHL4    4B1E          SHR2    4B19          SHR4    4B15          SID   = 0080
SINTST  4A96          SOD   = 0080          SOE   = 0040          SP    =%0008
SSCLK = 0040          SSTEP = 0005          STACK = 4FFF          STATRM= 4F20
STPCT   4F20          STRSP = 5000          SWMCST  4A94          SXFER   4A95
TADR00= 0080          TADR01= 0081          TADR02= 0082          TADR03= 0083
TADR04= 0084          TADR05= 0085          TADR06= 0086          TADR07= 0087
TADR10= 0088          TADR11= 0089          TADR12= 008A          TADR13= 008B
TAMT  = 0044          TASEL = 0080          TCMD  = 0040          TC.FWD= 0040
TC.INH= 0008          TC.LWR= 0004          TC.REV= 0020          TC.WRT= 0010
TEMP    4F99          TMF   = 0099          TMRDY = 0040          TPBCNT  4AA7
TRANS   4B68          TRANS1  4B6A          TRANS2  4B72          TRKENA= 00D2
TSET  = 2803          TSTEND= 2818          TSTS  = 0040          TST1    4300
TST1A   43DF          TST1B   43EA          TST1BC  4410  G       TST1C   4407
TST1D   4419          TST1DC  4422  G       TST1L   4318          TST1NB  4311
```

```
TST1ST  4305            TST1XC  4632  G         TUSEL0= 00D1            TUSEL1= 00D2
TU78  = 0010            T.ATH0= 0001            T.ATH1= 0002            T.BOT = 0004
T.EOT = 0002            T.FPT = 0001            T.NTHR= 0004            T.ONL = 0020
T.PES = 0008            T.PSBJ= 0020            T.PSOJ= 0008            T.PS1J= 0010
T.RDY = 0080            T.RDYO= 0040            T.RWD = 0010            T.SCLK= 0002
T1RST   4378            T1RSTA  43A8            T1RSTB  43B1            T1RSTC  43C9
UIBG  = 00A1            VALFC   4F98            VALTB   4F95            VELTST= 005B
WDR.P = 0010            WMCCTL= 00D3            WMCERR= 00DA            WMCSTA= 00D0
WRTCLK= 0000            WRTDAT= 00D3            W.ACRC= 0004            W.CRC = 0008
W.DIAG= 0002            W.DONN= 0040            W.ECC = 0010            W.ENAB= 0080
W.ERR = 0020            W.FMT = 0070            W.GCR = 0010            W.LEFT= 0004
W.ONES= 0020            W.RESI= 0002            W.REV = 0004            W.ROME= 0010
W.RST = 0001            W.SKIP= 000F            W.WRIT= 0008            W.XFER= 0020
X       =%000A          XFB4C   4600  G         XFRBIT  4489            XFRB1   4490
XFRB2   44AE            XFRB2C  44C4  G         XFRB3   44CD            XFRB3A  44D4
XFRB3C  44EA  G         XFRB4   44F3            XFRB4C  4609            XFRB4D  4528
XFRB4P  4638            XFRB4R  453B            XFRB5   4676            XFRB5C  469F  G
XFRERR  45FB            XFRGON  44A6  G         XFRRC1  45F2  G         XFRRDY  44F9
X.DONN= 0080            X.ENAB= 0040            X.PEPE= 0002            X.ROME= 0001
X.WCLK= 0001            Y       =%000B          .       = 4B7A


ERRORS DETECTED:  0

*WMC4.A78/PTP,WMC4=NLIST,PARAM,MACRO,LIST,WMC4
RUN-TIME: 5 9 0 SECONDS
CORE USED:  10K
```

     3   "MEMORY MAP" EXTENDED ADDRESS BIT
    19   READ MICRO CONTROLLER CONTROL & STATUS WORDS
   139   TRANSPORT/PORT COMMAND + STATUS WORDS
   380   COMMON ADDRESS SPACE REGISTERS
   444   MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
   485   MASSBUS CONTROL WORD DEFINITIONS
   557   WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
   709   KEYBOARD/DISPLAY CONTROL + STATUS WORD
   797   MISCELLANEOUS VARIABLE DEFINITIONS
   801   RAM ADDRESS DEFINITIONS
   819   LINKAGE TABLE ADDRESSES
   842   BIT DEFINITIONS
   856   RIM AND SIM INSTRUCTION BIT DEFINITIONS
   877   INTERRUPTABLE INSTRUCTION MACROS
   924   DIAGNOSTIC VARIABLES
   945   COMMON VARIABLE STORAGE
   959   MACRO CALL DEFINITIONS
  1047   TM78 MICRO-DIAGNOSTIC MACRO CALLS
  1101   8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
  1333   TEST 1 - WRITE MICRO ECC, ACRC, CRC TESTS
  2265   SUBROUTINE CKECC
  2329   SUBROUTINE CKCRC
  2404   PARITY CALCULATION ROUTINE
  2458   SUBROUTINE GENDAT
  2487   SUBROUTINE SETDDR
  2592   CLEAR ECC SUBROUTINE
  2606   CALCUALTE ECC CHARACTER
  2646   POLYNOMIAL BIT TRANSLATION

```
                ;*************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;               "BYTE/SCLK COUNT NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               MM = DATA FORMAT FROM CAS REGISTER 2
                ;               NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;               BYTE-SCLK COUNT = LLL
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               LLL = AS ABOVE
                ;               MM  = AS ABOVE
                ;               NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;               TRANSITION COUNT = LLL
                ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;               EXPECTED 18 BITS =      E EEEEEE
                ;               ACTUAL 18 BITS =        A AAAAAA
                ;
                ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;               OF CAS REG 15 LOW BYTE.
                ;
                ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;               17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;               AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;               "SUBGROUP NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;*************************************************************************
```

```
1016                    ; ************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ; ************************************************************************
1035                    ; ************************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ; ************************************************************************
1329    4300                    .-      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  WMC5 - WMC 'PE' MODE RECORD FORMATTING (XMC ENABLED)
                                .SBTTL  TEST 1 - WRITE MICRO ECC, ACRC, CRC TESTS
                                ;ID     WMC5-WRITE MICRO CONTROLLER PART #5
4300                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*'PE' FORMAT BYTE ASSEMBLY TESTS
4300                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST IS DESIGNED TO TEST THE WRITE MICROCONTROLLER ECC
                                ;*LOGIC.  A KNOWN DATA PATTERN IS FED INTO THE BYTE ASSEMBLY LOGIC FROM
                                ;*THE DIAGNOSTIC DATA REGISTERS - "DDR" - TO PROVIDE AN 8 BIT WIDE
                                ;*REPLACEMENT FOR MASSBUS DATA IN IMAGE MODE.
                                ;*
                                ;*THE BYTE ASSEMBLY LOGIC WILL "BREAK UP" THE INPUT DATA (8 BITS)
                                ;*INTO THE DESIRED BYTE FORMAT TO BE WRITTEN ON TAPE.
                                ;*
                                ;*THIS TEST WILL CHECK THE OUTPUT OF THE WRITE MICRO TO BE CORRECT
                                ;*IN REGARD TO THE ECC CHARACTERS GENERATED AND THE PROPER LOCATION
                                ;*OF THESE CHARACTERS IN THE DATA STREAM.
4300                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  INIT THE TEST
                                ;*  SET FORMAT MODE TO 5 (IMAGE)
                                ;*  SET TRANSFER LENGTH TO 0 BYTES
                                ;*  BGNDO
                                ;*  :  UPDATE LENGTH OF TRANSFER BY 1 BYTE
                                ;*  :  SET DATA PATTERN INDEX TO START
                                ;*  :  RESTART WMC + SET CLOCK CONTROL TO SINGLE-STEP SYSTEM
                                ;*  :  INIT COUNTER CONTROL
                                ;*  :  CLEAR ECODE COUNTER
                                ;*  :  CLEAR PROGRAM FLAGS AND COUNTERS
                                ;*  :  SET THE DDRCTL WORD TO 'OUT'-ENABLES DATA INJECTION TO WMC
                                ;*  :  GENERATE A SET OF DATA INTO DDR
                                ;*  :  INIT SIMULATED 'ECC' BYTE AND 'CRC' BYTE
                                ;*  :  CLOCK SYSTEM THROUGH RESTART CODE (WMC ROM PROGRAM)
                                ;*  :  CALL SUBROUTINE 'LKMOD7' (OPR U-CODE) TO LOAD THE BYTE COUNTER
                                ;*  :  LOAD RESCHR WORD WITH DATA FROM MOD7 SUBROUTINE-MOD7 REMAINDER
                                ;*  :  CLEAR PAD COUNTER-NOT USED IN 'PE' DATA TRANSFERS
                                ;*  :  CALCULATE PROPER NUMBER OF 'PAD' BYTES TO EXPECT FOR CURRENT SIZE OF DATA XFR
                                ;*  :  LOAD DATACTL WORD FOR FORMAT MODE #5 (IMAGE)
                                ;*  :  START WMC ROM PGM BY SETTING WMCCTL 'ENABLE' BITS AND 'WRITE' BIT
                                ;*  :  INIT TIMEOUT COUNTER
                                ;*  :  BGNDO
                                ;*  :  :  CLOCK THE SYSTEM
                                ;*  :  :  IF WMCSTA HAS NO 'XFER' BIT SET
                                ;*  :  :  :  THEN-UPDATE TIMEOUT COUNTER
                                ;*  :  :  :  :  IF TIMEOUT DETECTED
```

```
1377                                ;*  :  :  :  :  :  THEN-REPORT LACK OF ACTIVITY IN SPECIFIED TIME
1378                                ;*  :  :  :  :  :  ELSE-CONTINUE
1379                                ;*  :  :  :  :  ENDIF
1380                                ;*  :  :  :  ELSE-GET WMCSTA DATA AND SAVE FOR LATER TESTS
1381                                ;*  :  :  :  :  BGNDO
1382                                ;*  :  :  :  :  :  CLOCK SYSTEM TO FINISH 1ST NIBBLE-'XFER' GOES AWAY
1383                                ;*  :  :  :  :  :  IF > 20 CLOCKS
1384                                ;*  :  :  :  :  :  :  THEN-REPORT FAILURE TO CLEAR 'XFER' BIT ON 1ST NIBBLE
1385                                ;*  :  :  :  :  :  :  ELSE-CONTINUE
1386                                ;*  :  :  :  :  :  ENDIF
1387                                ;*  :  :  :  :  :  DO UNTIL 'XFER' CLEARS OR TIMEOUT
1388                                ;*  :  :  :  :  ENDDO
1389                                ;*  :  :  :  BGNDO
1390                                ;*  :  :  :  :  CLOCK SYSTEM TO BEGIN 2ND NIBBLE ASSEMBLY
1391                                ;*  :  :  :  :  IF > 128 CLOCKS
1392                                ;*  :  :  :  :  :  THEN-REPORT FAILURE TO SET 'XFER' FOR 2ND NIBBLE
1393                                ;*  :  :  :  :  :  ELSE-CONTINUE
1394                                ;*  :  :  :  :  ENDIF
1395                                ;*  :  :  :  :  DO UNTIL 'XFER' SETS OR TIMEOUT
1396                                ;*  :  :  :  ENDDO
1397                                ;*  :  :  :  UPDATE COUNT OF 'ZFER' STATUS RECEIVED
1398                                ;*  :  :  :  GET PARITY BIT ASSOCIATED WITH BYTE ASSEMBLED IN WRTDAT
1399                                ;*  :  :  :  BGNDO
1400                                ;*  :  :  :  :  CLOCK SYSTEM TO END 2ND NIBBLE
1401                                ;*  :  :  :  :  IF > 20 CLOCKS
1402                                ;*  :  :  :  :  :  THEN-REPORT FAILURE TO CLEAR 'XFER' TO FINISH 2ND NIBBLE
1403                                ;*  :  :  :  :  :  ELSE-CONTINUE
1404                                ;*  :  :  :  :  ENDIF
1405                                ;*  :  :  :  ENDDO
1406                                ;*  :  :  :  CLOCK SYSTEM TWICE TO 'SET' INTERNAL STATUS FOR ASSEMBLED BYTE
1407                                ;*  :  :  :  IF WMCSTA INDICATES A SPECIAL COUNTER
1408                                ;*  :  :  :  :  THEN-DETERMINE WHICH CHARACTER
1409                                ;*  :  :  :  :  :  IF CHARACTER IS 'ACRC'
1410                                ;*  :  :  :  :  :  :  THEN-REPORT 'ACRC' SENT IN PE MODE-ILLEGAL
1411                                ;*  :  :  :  :  :  :  ELSE-CONTINUE
1412                                ;*  :  :  :  :  :  ENDIF
1413                                ;*  :  :  :  :  :  IF CHARACTER IS 'RESID'
1414                                ;*  :  :  :  :  :  :  THEN-REPORT 'RESID' PRESENTED IN PE MODE-ILLEGAL
1415                                ;*  :  :  :  :  :  :  ELSE-CONTINUE
1416                                ;*  :  :  :  :  :  ENDIF
1417                                ;*  :  :  :  :  :  EXIT DO LOOP
1418                                ;*  :  :  :  :  ELSE-CONTINUE
1419                                ;*  :  :  :  ENDIF
1420                                ;*  :  :  :  UPDATE DATA BYTE RECEIVED COUNT FOR 'ECC' CHECKING
1421                                ;*  :  :  :  IF ALL DATA BYTES RECEIVED FROM WMC
1422                                ;*  :  :  :  :  THEN-CHECK 'PAD' BYTE RECEIVED
1423                                ;*  :  :  :  :  :  IF WRTDAT DATA=0 (VALID 'PAD' CHARACTER)
1424                                ;*  :  :  :  :  :  :  THEN-CONTINUE
1425                                ;*  :  :  :  :  :  :  ELSE-REPORT INVALID 'PAD' BYTE ASSEMBLED IN 'PE' MODE
1426                                ;*  :  :  :  :  :  ENDIF
1427                                ;*  :  :  :  :  :  IF PARITY BIT ON 'PAD' BYTE IS OK
1428                                ;*  :  :  :  :  :  :  THEN-CONTINUE
1429                                ;*  :  :  :  :  :  :  ELSE-REPORT BAD PARITY ON 'PAD' BYTE
1430                                ;*  :  :  :  :  :  ENDIF
```

```
                                ;*  :  :  :  :  :  ADD 'PAD' BYTE TO SIMULATED 'ECC' BYTE
                                ;*  :  :  :  :  .  IF ALL EXPECTED 'PAD' CHARACTERS RECEIVED
                                ;*  :  :  :  :  :  :  THEN-EXIT DO LOOP
                                ;*  :  :  :  :  :  :  ELSE-WAIT FOR TRANSLATOR TO TAKE THE 'SUBGROUP' FROM WMC
                                ;*  :  :  :  :  :  :  BGNDO
                                ;*  :  :  :  :  :  :  :  CLOCK THE SYSTEM
                                ;*  :  :  :  :  :  :  :  DO UNTIL TRANSLATOR IS CLOCKED ENOUGH TIMES TO RESPOND
                                ;*  :  :  :  :  :  :  ENDDO
                                ;*  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  ELSE-CHECK DATA BYTE ASSEMBLED IN WRTDAT
                                ;*  :  :  :  :  :  IF BYTE ASSEMBLED=EXPECTED BYTE
                                ;*  :  :  :  :  :  :  THEN-CONTINUE
                                ;*  :  :  :  :  :  :  ELSE-REPORT PE BYTE ASSEMBLY FAULT
                                ;*  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  :  IF PARITY ON ASSEMBLED BYTE IS OK
                                ;*  :  :  :  :  :  :  THEN-CONTINUE
                                ;*  :  :  :  :  :  :  ELSE-REPORT BAD PARITY ON DATA BYTE ASSEMBLED
                                ;*  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  :  ADD DATA BYTE TO SIMULATED 'ECC' WORD
                                ;*  :  :  :  :  :  ADD ALSO TO SIMULATED 'CRC' WORD
                                ;*  :  :  :  :  :  IF NECESSARY TO CHECK BYTE COUNTER
                                ;*  :  :  :  :  :  :  THEN-GET BYTECT DATA
                                ;*  :  :  :  :  :  :  :  IF BYTE COUNT=0 (AS EXPECTED)
                                ;*  :  :  :  :  :  :  :  :  THEN-CONTINUE
                                ;*  :  :  :  :  :  :  :  :  ELSE-REPORT BYTE COUNTER NOT=0 WHEN EXPECTED
                                ;*  :  :  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  :  :  ELSE-CONTINUE
                                ;*  :  :  :  :  :  ENDIF
                                ;*  :  :  :  :  'EXIT DO LOOP
                                ;*  :  :  :  :  ENDIF
                                ;*  :  :  :  ENDIF
                                ;*  :  :  ENDIF
                                ;*  :  :  DO UNTIL ALL 'XFER' BYTES RECEIVED FOR THIS DATA XFR
                                ;*  :  ENDDO
                                ;*  :  DO UNTIL ALL DATA PATTERNS/TRANSFERS HAVE BEEN TESTED
                                ;*  ENDDO
                                ;*ENDTST
1468  4300                      SE
 (1)                            ;*****************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1469                            ;*WMC5 MICRO TEST 01
1470                            ;*WMC5 MICRO ERROR 01
1471                            ;*WMC5-WMC 'PE' FORMAT TEST
1472                            ;*M8959, M8958
1473                            ;*WMC ROM PROGRAM FAILED TO REQUEST NEXT 'SCLK' CYCLE IN 128 CLOCK CYCLES
1474                            ;*TEST ABORTED!
1475                            ;*
1476                            ;*WMC5 MICRO TEST 01
1477                            ;*WMC5 MICRO ERROR 02
1478                            ;*WMC5-WMC 'PE' FORMAT TEST
1479                            ;*M8959, M8958
1480                            ;*WMC ROM PGM FAILED TO SET 'XFER' BIT IN 128 CLOCK CYCLES AFTER XFR STARTED
1481                            ;*TEST ABORTED.
```

```
1482                                    ;*
1483                                    ;*WMC5 MICRO TEST 01
1484                                    ;*WMC5 MICRO ERROR 03
1485                                    ;*WMC5-WMC 'PE' FORMAT TEST
1486                                    ;*M8959, M8958
1487                                    ;*WMC FAILED TO DROP 'SCLK' BIT AFTER SEVERAL SYSTEM CLOCKS
1488                                    ;*TEST ABORTED!
1489                                    ;*
1490                                    ;*WMC5 MICRO TEST 01
1491                                    ;*WMC5 MICRO ERROR 04
1492                                    ;*WMC5-WMC 'PE' FORMAT TEST
1493                                    ;*M8959, M8958
1494                                    ;*WMC FAILED TO CLEAR 'XFER' BIT ON 1ST NIBBLE OF THIS BYTE ASSY
1495                                    ;*
1496                                    ;*WMC5 MICRO TEST 01
1497                                    ;*WMC5 MICRO ERROR 05
1498                                    ;*WMC5-WMC 'PE' FORMAT TEST
1499                                    ;*M8959, M8958
1500                                    ;*WMC FAILED TO SET 'XFER' BIT FOR 2ND NIBBLE ON THIS BYTE ASSY
1501                                    ;*BYTE-SCLK COUNT = LLL
1502                                    ;*DATA FORMAT = MM
1503                                    ;*SKIP COUNT  = NN
1504                                    ;*TEST ABORTED'
1505                                    ;*
1506                                    ;*WMC5 MICRO TEST 01
1507                                    ;*WMC5 MICRO ERROR 06
1508                                    ;*WMC5-WMC 'PE' FORMAT TEST
1509                                    ;*M8959, M8958
1510                                    ;*WMC FAILED TO CLEAR 'XFER' BIT FOR 2ND NIBBLE OF THIS BYTE ASSY
1511                                    ;*BYTE-SCLK COUNT = LLL
1512                                    ;*DATA FORMAT = MM
1513                                    ;*SKIP COUNT  = NN
1514                                    ;*TEST ABORTED!
1515                                    ;*
1516                                    ;*WMC5 MICRO TEST 01
1517                                    ;*WMC5 MICRO ERROR 07
1518                                    ;*WMC5-WMC 'PE' FORMAT TEST
1519                                    ;*M8959, M8958
1520                                    ;*WMC PRESENTED AN 'ACRC' CHARACTER IN 'PE' MODE - ILLEGAL
1521                                    ;*BYTE-SCLK COUNT = LLL
1522                                    ;*DATA FORMAT = MM
1523                                    ;*SKIP COUNT  = NN
1524                                    ;*
1525                                    ;*WMC5 MICRO TEST 01
1526                                    ;*WMC5 MICRO ERROR 10
1527                                    ;*WMC -WMC 'PE' FORMAT TEST
1528                                    ;*M8959, M8958
1529                                    ;*WMC PRESENTED 'RESID' BYTE IN 'PE' MODE - ILLEGAL
1530                                    ;*BYTE-SCLK COUNT = LLL
1531                                    ;*DATA FORMAT = MM
1532                                    ;*SKIP COUNT  = NN
1533                                    ;*
1534                                    ;*WMC5 MICRO TEST 01
1535                                    ;*WMC5 MICRO ERROR 11
```

```
1536                                    ;*WMC5-WMC 'PE' FORMAT TEST
1537                                    ;*M8959, M8958
1538                                    ;*PAD CHARACTER BYTE INCORRECT
1539                                    ;*ACTUAL = NNNN
1540                                    ;*EXPECTED = NNNN
1541                                    ;*
1542                                    ;*WMC5 MICRO TEST 01
1543                                    ;*WMC5 MICRO ERROR 12
1544                                    ;*WMC5-WMC 'PE' FORMAT TEST
1545                                    ;*M8959, M8958
1546                                    ;*PARITY BIT ON PAD SHOULD =0 AND WAS =1
1547                                    ;*ACTUAL = NNNN
1548                                    ;*
1549                                    ;*WMC5 MICRO TEST 01
1550                                    ;*WMC5 MICRO ERROR 13
1551                                    ;*WMC5-WMC 'PE' FORMAT TEST
1552                                    ;*M8959, M8958
1553                                    ;*PARITY BIT ON PAD WAS =0 AND SHOULD BE =1
1554                                    ;*ACTUAL = NNNN
1555                                    ;*
1556                                    ;*WMC5 MICRO TEST 01
1557                                    ;*WMC5 MICRO ERROR 14
1558                                    ;*WMC5-WMC 'PE' FORMAT TEST
1559                                    ;*M8959, M8958
1560                                    ;*XMC FAILED TO FINISH CLOCKING SO NEXT WMC CHAR CAN BE SENT
1561                                    ;*TEST ABORTED!
1562                                    ;*
1563                                    ;*WMC5 MICRO TEST 01
1564                                    ;*WMC5 MICRO ERROR 15
1565                                    ;*WMC5-WMC 'PE' FORMAT TEST
1566                                    ;*M8959, M8958
1567                                    ;*WMC SET 'XFER' ONLY - AFTER ALL DATA AND PAD CHARACTERS OUTPUT
1568                                    ;*
1569                                    ;*WMC5 MICRO TEST 01
1570                                    ;*WMC5 MICRO ERROR 16
1571                                    ;*WMC5-WMC 'PE' FORMAT TEST
1572                                    ;*M8959, M8958
1573                                    ;*WMC "PE" BYTE ASSEMBLY FAULT - BYTE ASSEMBLED DOESN'T MATCH EXPECTED DATA
1574                                    ;*BYTE-SCLK COUNT = LLL
1575                                    ;*DATA FORMAT = MM
1576                                    ;*SKIP COUNT  = NN
1577                                    ;*ACTUAL = NNNN
1578                                    ;*EXPECTED = NNNN
1579                                    ;*
1580                                    ;*WMC5 MICRO TEST 01
1581                                    ;*WMC5 MICRO ERROR 17
1582                                    ;*WMC5-WMC 'PE' FORMAT TEST
1583                                    ;*M8959, M8958
1584                                    ;*PARITY BIT ON DATA BYTE SHOULD -0 AND WAS =1
1585                                    ;*ACTUAL = NNNN
1586                                    ;*
1587                                    ;*WMC5 MICRO TEST 01
1588                                    ;*WMC5 MICRO ERROR 20
1589                                    ;*WMC5-WMC 'PE' FORMAT TEST
```

```
1590                                    ;*M8959, M8958
1591                                    ;*PARITY BIT ON DATA WAS =0 AND SHOULD BE =1
1592                                    ;*ACTUAL = NNNN
1593                                    ;*
1594                                    ;*WMC5 MICRO TEST 01
1595                                    ;*WMC5 MICRO ERROR 21
1596                                    ;*WMC5-WMC 'PE' FORMAT TEST
1597                                    ;*M8959, M8958
1598                                    ;*WMC BYTE COUNTER EXPECTED TO BE = 0 - ALL DATA BYTES RECEIVED FROM WMC
1599                                    ;*BYTE-SCLK COUNT = LLL
1600                                    ;*DATA FORMAT = MM
1601                                    ;*SKIP COUNT  = NN
1602                                    ;*ACTUAL = NNNN
1603                                    ;*
1604                                    ;*WMC5 MICRO TEST 01
1605                                    ;*WMC5 MICRO ERROR 22
1606                                    ;*WMC5-WMC 'PE' FORMAT TEST
1607                                    ;*M8959, M8958
1608                                    ;*SPECIAL 'PE' TERMINATING 'CRC' CHARACTER INCORRECT
1609                                    ;*ACTUAL = NNNN
1610                                    ;*EXPECTED = NNNN
1611                                    ;*
1612                                    ;*WMC5 MICRO TEST 01
1613                                    ;*WMC5 MICRO ERROR 23
1614                                    ;*WMC5-WMC 'PE' FORMAT TEST
1615                                    ;*M8959, M8958
1616                                    ;*WMC DONE STATUS NOT INDICATED AFTER LAST WMC FUNCTION PERFORMED
1617                                    ;*ACTUAL = NNNN
1618                                    ;*
1619                                    ;*WMC5 MICRO TEST 01
1620                                    ;*WMC5 MICRO ERROR 24
1621                                    ;*WMC5-WMC 'PE' FORMAT TEST
1622                                    ;*M8959, M8958
1623                                    ;*WMC PRESENTED AN 'ECC' CHAR WHEN NOT EXPECTED - ACTUAL = 'XFER' COUNT
1624                                    ;*ACTUAL = NNNN
1625                                    ;*BYTE-SCLK COUNT = LLL
1626                                    ;*DATA FORMAT = MM
1627                                    ;*SKIP COUNT  = NN
1628                                    ;*
1629                                    ;*WMC5 MICRO TEST 01
1630                                    ;*WMC5 MICRO ERROR 25
1631                                    ;*WMC5-WMC 'PE' FORMAT TEST
1632                                    ;*M8959, M8958
1633                                    ;*WMC ECC WORD ON 'BAL' OUTPUT NOT AS EXPECTED IN PE MODE
1634                                    ;*ACTUAL = NNNN
1635                                    ;*EXPECTED = NNNN
1636                                    ;*BYTE-SCLK COUNT = LLL
1637                                    ;*DATA FORMAT = MM
1638                                    ;*SKIP COUNT  = NN
1639                                    ;*
1640                                    ;*WMC5 MICRO TEST 01
1641                                    ;*WMC5 MICRO ERROR 26
1642                                    ;*WMC5-WMC 'PE' FORMAT TEST
1643                                    ;*M8959, M8958
```

```
                                        ;*PARITY BIT ON ECC SHOULD =0 AND WAS =1
                                        ;*ACTUAL = NNNN
                                        ;*
                                        ;*WMC5 MICRO TEST 01
                                        ;*WMC5 MICRO ERROR 27
                                        ;*WMC5-WMC 'PE' FORMAT TEST
                                        ;*M8959, M8958
                                        ;*PARITY BIT ON ECC WAS =0 AND SHOULD BE =1
                                        ;*ACTUAL = NNNN
                                        ;*
                                        ;*WMC5 MICRO TEST 01
                                        ;*WMC5 MICRO ERROR 28
                                        ;*WMC5-WMC 'PE' FORMAT TEST
                                        ;*M8959, M8958
                                        ;*WMC ACTUAL 'CRC' CHARACTER NOT SAME AS SIMULATED 'CRC' CHARACTER
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4300                                    $
 (1)                                    ; ***************************************************************************

4300                                    TST1:   TESTX   @1
 (1) 4300  3E  01            7.0                         MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1) 4302  CD  03  28       18.0                         CALL    TSET            ;SETUP THE TEST
                                        ;%WMC5-WMC 'PE' FORMAT TEST
                                        ;&M8959, M8958

                                        ;INIT FORMAT CODE COUNTER TO 5 - FORMAT CODE = IMAGE MODE

4305  3E  05                7.0  TST1ST: MVI     A,@5            ;LOAD THE IMAGE MODE FORMAT CODE
4307  32  4E  48           13.0           STA     FORMT           ;SAVE FORMAT CODE
430A  AF                    4.0           XRA     A
430B  32  66  48           13.0           STA     TPBCNT          ;CLEAR THE TEST PASS BYTE COUNT

                                        ;HERE TO UPDATE THE NUMBER OF BYTES TO USE THIS TEST PASS

430E  3A  66  48           13.0  TST1NBC: LDA     TPBCNT          ;GET THE BYTE COUNT
4311  3C                    4.0           INR     A
4312  32  66  48           13.0           STA     TPBCNT          ;SAVE THE NEW BYTE COUNT TO USE

                                        ;FOLLOWING IS A DUMMY HOST REQUEST FOR UNIT SELECT INFO...
                                        ;THIS IS NEEDED TO SIGNAL THE HOST THAT THE TEST IS STILL RUNNING.
                                        ;THIS WHOLE TEST TAKES ABOUT A MIN. OR SO TO RUN!

4315                                            REQ     7
 (1) 4315  CD  06  28       18.0                         CALL    REQST
 (1) 4318  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (1) 4319  00  00                                        .WORD                   ;SYSTEM '"' COUNT
 (1) 431B  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1) 431D  00                                            .BYTE                   ;DATA COMPARE FLAG IF -1
 (1) 431E  07                                            .BYTE   7               ;REQUEST CODE

                                        ;INIT COUNTER CONTROL FOR BYTE COUNTER AND ECODE COUNTER

                                        ;RESTART WRITE MICROCONTROLLER ROM PROGRAM
```

```
1689
1690  431F  3E FF          7.0  TST1L:  MVI   A,@377        ;-1 TO OFFSET NUMBER
1691  4321  32 4F 48       13.0          STA   DPNUM         ;INIT DATA PATTERN OFFSET SO 1ST WILL BE "0"
1692  4324  3E 01          7.0          MVI   A,W.RST       ;GET 'RST' BIT
1693  4326  D3 D3         10.0          OUT   WMCCTL        ;RESTART WMC
1694  4328  3E 05          7.0          MVI   A,SSTEP       ;SET THE SYSTEM TO SINGLE STEP MODE
1695  432A  32 4A 48       13.0          STA   CCTLWD        ;SAVE THE CONTROL WORD
1696  432D  D3 F0         10.0          OUT   CLKCTL        ;WRITE THE CLOCK CONTROL WORD
1697  432F  AF             4.0          XRA   A             ;FINISH THE WMC RESTART
1698  4330  D3 D3         10.0          OUT   WMCCTL
1699  4332  3E 30          7.0          MVI   A,@60
1700  4334  D3 D7         10.0          OUT   CNTCTL        ;SELECT BYTE COUNTER
1701  4336  3E 70          7.0          MVI   A,@160
1702  4338  D3 D7         10.0          OUT   CNTCTL        ;SELECT THE PAD COUNTER
1703  433A  3E B0          7.0          MVI   A,@260
1704  433C  D3 D7         10.0          OUT   CNTCTL        ;SELECT ECODE COUNTER
1705
1706                                ;CLEAR ERROR CODE COUNTER
1707  433E                        CLRECT                    ;CLEAR ECODE COUNTER
 (1)  433E  AF             4.0                XRA   A      ;CLEAR THE ACCUMULATOR
 (1)  433F  D3 D6         10.0                OUT   ERRCNT ;CLEAR BITS 7-0
 (1)  4341  D3 D6         10.0                OUT   ERRCNT ;CLEAR BITS 15-8
1708
1709  4343  32 51 48       13.0          STA   ECCCNT        ;CLEAR 'ECC' CHECK COUNTER
1710  4346  32 5D 48       13.0          STA   ECCEXP        ;CLEAR ECC EXPECTED FLAG
1711  4349  32 52 48       13.0          STA   CLKCNT        ;CLEAR THE 'SCLK' COUNTER
1712  434C  32 4C 48       13.0          STA   ABCNT         ;CLEAR THE ASSEM. BYTE COUNTER
1713  434F  32 62 48       13.0          STA   HAVCRC        ;CLEAR THE HAVE CRC FLAG
1714  4352  32 4D 48       13.0          STA   EABCNT        ;
1715  4355  32 65 48       13.0          STA   CLKCRC        ;CLEAR THE EXTRA CLOCK-TO-CRC FLAG
1716  4358  32 68 48       13.0          STA   LASTCRC       ;CLEAR THE END 'CRC' FLAG
1717  435B  32 63 48       13.0          STA   RPADCT        ;CLEAR COUNT OF RECEIVED PAD CHAR.
1718  435E  32 6B 48       13.0          STA   XFRREC        ;CLEAR COUNTER FOR 'XFER' BYTES RECEIVED
1719
1720                          ;SET DDR CONTROL TO "OUT" FOR DATA INJECTION INTO BAL LOGIC
1721
1722  4361  3E 88          7.0          MVI   A,@210        ;SET "OUT" BITS
1723  4363  D3 DB         10.0          OUT   DDRCTL        ;LOAD DDR CONTROL
1724
1725  4365  CD 6C 48       18.0          CALL  GENDAT        ;GENERATE THE 1ST SET OF DATA
1726  4368  11 B1 48       10.0          LXI   D,CDATBL      ;LOAD THE ADDRESS OF "EXPECTED" DATA TABLE
1727  436B  CD F5 48       18.0          CALL  CLECC         ;INIT THE ECC WORD - "ECCCHR"
1728  436E  CD 2B 48       18.0          CALL  CLCRC         ;INIT CRC WORD
1729
1730  4371  06 B4          7.0          MVI   B,180         ;CLOCK THE SYSTEM THRU THE RESTART AREA
1731  4373  CD E9 48       18.0  T1RST:  CALL  CLKSYS       ;CLOCK THE SYSTEM
1732  4376  05             4.0          DCR   B
1733  4377  C2 73 43       10.0          JNZ   T1RST         ;STAY HERE TILL DONE COUNTING
1734
1735  437A  3A 66 48       13.0          LDA   TPBCNT        ;GET THE BYTE COUNT TO USE
1736  437D  32 23 4F       13.0          STA   BYTEL         ;SAVE THE BYTE COUNT LOW  ALUE
1737  4380  AF             4.0          XRA   A
1738  4381  32 24 4F       13.0          STA   BYTEH         ;CLEAR THE BYTE COUNT HIGH VALUE
1739  4384  CD 46 00       18.0          CALL  LKMOD7        ;CALL MOD7 ROUTINE FROM OPR U-CODE
```

```
1740  4387  78              4.0          MOV     A,B             ;GET MOD7 REMAINDER FROM LKMOD7
1741  4388  32  69  48      13.0         STA     MOD7R           ;SAVE THE # RESIDUAL DATA BYTES EXPECTED
1742  438B  0F              4.0          RRC                     ;POSITION THE MOD7 NUMBER
1743  438C  0F              4.0          RRC
1744  438D  0F              4.0          RRC
1745  438E  32  6A  48      13.0         STA     RCHRIM          ;SAVE IT
1746  4391  D3  D1          10.0         OUT     RESCHR          ;LOAD THE RESID CHAR WORD IN WMC
1747  4393  3A  66  48      13.0         LDA     TPBCNT          ;GET THE BC AGAIN
1748  4396  90              4.0          SUB     B               ;CALC THE VALUE LOADED INTO THE BYTE COUNTER
1749  4397  3C              4.0          INR     A
1750  4398  32  4B  48      13.0         STA     IMBYTC          ;SAVE FOR LATER TESTING
1751
1752  439B  AF              4.0  T1RSTA: XRA     A
1753  439C  D3  D5          10.0         OUT     PADCNT          ;LOAD PAD COUNT 7-0
1754
1755  439E  AF              4.0  T1RSTB: XRA     A
1756  439F  D3  D5          10.0         OUT     PADCNT          ;LOAD PAD CCOUNTER 15-8
1757  43A1  3A  66  48      13.0         LDA     TPBCNT          ;GET # BYTES IN XFR
1758  43A4  E6  07          7.0          ANI     7
1759  43A6  FE  07          7.0          CPI     7               ;EXPECT 'ECC'
1760  43A8  C2  AE  43      10.0         JNZ     T1RSTD          ;NOT EXPECTING AN 'ECC' BYTE
1761  43AB  32  5D  48      13.0         STA     ECCEXP          ;SET 'ECC' BYTE EXPECTED FLAG
1762
1763  43AE  3A  69  48      13.0  T1RSTD: LDA     MOD7R           ;GET # BYTES IN 'RESID' GROUP
1764  43B1  4F              4.0          MOV     C,A             ;COPY FOR DOUBLE ADD
1765  43B2  06  00          7.0          MVI     B,0             ;TO GEN AN OFFSET INTO A TABLE
1766  43B4  21  BF  43      10.0         LXI     H,PEXPTB        ;POINT TO PAD EXPECTED TABLE
1767  43B7  09              10.0         DAD     B               ;MAKE THE OFFSET
1768  43B8  7E              7.0          MOV     A,M             ;GET THE # OF EXPECTED PADS FOR XFR
1769  43B9  32  5C  48      13.0         STA     PADEXP          ;SAVE NUMBER OF EXPECTED 'PAD' BYTES IN 'PE'
1770  43BC  C3  C7  43      10.0         JMP     T1RSTC          ;JUMP OVER THE TABLE
1771
1772  43BF  04                     PEXPTB: .BYTE   4               ;4 PADS IF NO 'RESID' GRP & EXPECT ECC
1773  43C0  03                             .BYTE   3               ; 1 BYTE IN 'RESID' GROUP- 3 PADS
1774  43C1  02                             .BYTE   2               ; 2 BYTES IN RES GRP = 2 PADS FOR PE
1775  43C2  01                             .BYTE   1               ; 3 BYTES = 1 PAD
1776  43C3  04                             .BYTE   4               ; 4 BYTES = 4 PADS
1777  43C4  03                             .BYTE   3
1778  43C5  02                             .BYTE   2
1779  43C6  04                             .BYTE   4               ;EXPECTING AN ECC CHAR SO WILL HAVE 4 PADS
1780
1781                         ;LOAD DATA CONTROL WITH SELECTED FORMAT CODE
1782
1783  43C7  3E  50          7.0  T1RSTC. MVI     A,$50           ;LOAD IMAGE FORMAT CODE
1784  43C9                       ROUT    RO2H            ;ALSO SAVE IN CAS REG 02
 (1)  43C9  D3  85          10.0                 OUT     RO2H            ;WRITE AC INTO RO2H
 (1)  43CB  7F              4.0                  MOV     A,A             ;RETRY LINK
1785  43CC  D3  D0          10.0         OUT     DATACTL         ;LOAD DATA CONTROL WORD
1786  43CE  3E  04          7.0          MVI     A,4             ;SET RECORD COUNT TO 1
1787  43D0                       ROUT    RO2L
 (1)  43D0  D3  84          10.0                 OUT     RO2L            ;WRITE AC INTO RO2L
 (1)  43D2  7F              4.0                  MOV     A,A             ;RETRY LINK
1788
1789                         ;ENABLE WRITE MICROCONTROLLER - START OF DATA XFR - WRITE
```

```
1790
1791  43D3  3E  C8        7.0  TST1A:  MVI   A,W.ENAB!X.ENAB.W.WRITE  ;SELECT 'WMC EN' + 'WRITE' (PE MODE)
1792  43D5  D3  D3       10.0          OUT   WMCCTL                    ;LOAD WRITE MICROCONTROLLER CONTROL
1793  43D7  AF            4.0          XRA   A
1794  43D8  32  50  48   13.0          STA   CLKON                     ;CLEAR 'SCLK' ON FLAG
1795
1796  43DB  06  FF        7.0          MVI   B,@377                    ;SET SINGLE STEP CLOCK 'TIMEOUT' LIMIT TO MAX POS. COUNT
1797  43DD  48            4.0          MOV   C,B                       ;COPY FOR THE 'XFER' TIMEOUT COUNTER
1798
1799  43DE  CD  74  44   18.0  TST1B:  CALL  CKSCLK                    ;ISSUE SINGLE STEP CLOCK
1800  43E1  DA  1F  43   10.0          JC    TST1L                     ;JUMP BACK IF ERROR DETECTED
1801  43E4  DB  D0       10.0          IN    WMCSTA                    ;GET WMC STATUS WORD
1802  43E6  E6  20        7.0          ANI   W.XFER                    ;SAVE ONLY 'XFER' BIT
1803  43E8  C2  8A  44   10.0          JNZ   XFRBIT                    ;JUMP IF 'XFER' IS SET - DATA READY IN WRITE DATA WORD
1804  43EB  3A  5A  48   13.0          LDA   SXFER                     ;GET THE SOFTWARE XFER FLAG
1805  43EE  A7            4.0          ANA   A
1806  43EF  CA  FB  43   10.0          JZ    TST1C                     ;JUMP IF XFER NOT SET
1807  43F2  AF            4.0          XRA   A                         ;CLEAR SOFTWARE FLAG
1808  43F3  32  5A  48   13.0          STA   SXFER                     ;FOR NEXT PASS
1809  43F6  06  7F        7.0          MVI   B,@177                    ;SETUP A WATCHDOG COUNT
1810  43F8  C3  AF  44   10.0          JMP   XFRB2                     ;PROCESS THE 'XFER' BIT DETECTED
1811
1812                                ;HERE TO SEE IF 'SCLK' TIMEOUT HAPPENED
1813
1814  43FB  05            4.0  TST1C:  DCR   B                       ;COUNT CLOCK CYCLE
1815  43FC  C2  14  44   10.0          JNZ   TST1D                     ;TIMEOUT? - JUMP IF OK
1816  43FF  3A  68  48   13.0          LDA   LASTCRC
1817  4402  A7            4.0          ANA   A
1818  4403  C2  ED  46   10.0          JNZ   QUIT                      ;EXIT IF END OF XFR
1819
1820  4406                             ERR   TST1L,TST1BC
 (1)                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4406  CD  09  28   18.0                CALL  ERLP                ;PROCESS ERROR - DO 2.3
 (1)        0001                     MSGN  =     MSGN+1                ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4409  01                       .BYTE MSGN                      ;MESSAGE NUMBER ID
 (1)  440A  00                       .BYTE
 (1)  440B  CD  15  28   18.0  TST1BC::      CALL  CKLOP               ;CHECK LOOP FUNCTION - DO 2.3
 (1)  440E  DA  1F  43   10.0                JC    TST1L               ;LOOP ADDRESS IF LOOP SPECIFIED
1821                             ;>WMC ROM PROGRAM FAILED TO REQUEST NEXT 'SCLK' CYCLE IN 128 CLOCK CYCLES
1822                             ;<TEST ABORTED!
1823  4411  C3  18  28   10.0          JMP   TSTEND                    ;TERMINATE TESTING NOW! - FATAL ERROR
1824
1825                                ;HERE TO SEE IF 'XFER' BIT TIMEOUT HAPPENED
1826
1827  4414  0D            4.0  TST1D:  DCR   C                       ;DECREMENT 'XFER' WATCHDOG COUNTER
1828  4415  C2  DE  43   10.0          JNZ   TST1B                     ;BACK TO LOOP IF OK
```

```
1830  4418                                    ERR     TST1L,TST1DC
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4418  CD   09   28       18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  441B  02                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  441C  00                                        .BYTE
 (1)  441D  CD   15   28       18.0            TST1DC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4420  DA   1F   43       10.0                    JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1831                                    ;>WMC ROM PGM FAILED TO SET 'XFER' BIT IN 128 CLOCK CYCLES AFTER XFR STARTED
1832                                    ;<TEST ABORTED!
1833  4423  C3   18   28       10.0            JMP     TSTEND
```

```
1836  4426                                  S
 (1)                                        ; ***************************************************************************
1837                                        ;SCLKBIT -- HERE WHEN A CLOCK CYCLE RESULTS IN 'SCLK' SETTING IN DBUS
1838                                        ;       CONTROL WORD.  LOAD NEXT DATA SET (18 BITS) INTO DDR REGS.
1839  4426                                  S
 (1)                                        ; ***************************************************************************
1840
1841  4426  3A  50  48      13.0  SCLKBIT: LDA     CLKON            ;SEE IF 'SCLK' WAS ON
1842  4429  A7                     4.0           ANA     A                ;SET CONDITION BITS
1843  442A  C0                    12.0           RNZ                      ;EXIT IF CLOCK WAS SET-HAVE DATA ALREADY
1844  442B  CD  8C  48      18.0           CALL    SETDDR           ;THEN LOAD THE DATA INTO THE DDR REGS
1845  442E  C5                    12.0           PUSH    B                ;SAVE REG B & C
1846  442F  06  14                 7.0           MVI     B,20             ;GET A WATCHDOG COUNT
1847  4431  0E  FF                 7.0  SCKWT:   MVI     C,@377
1848  4433  CD  E9  48      18.0  SCKWT1: CALL    CLKSYS           ;CLOCK THE SYSTEM - WAIT FOR 'SCLK' TO GO AWAY
1849  4436  DB  D0                10.0           IN      WMCSTA           ;GET THE WMC STATUS WORD
1850  4438  E6  20                 7.0           ANI     W.XFER           ;DID 'XFER' BIT SET?
1851  443A  CA  40  44      10.0           JZ      SCKWT2           ;NO - JUMP OVER
1852  443D  32  5A  48      13.0           STA     SXFER            ;YUP - SET THE FLAG
1853  4440  DB  C0                10.0  SCKWT2: IN      DBUSSTA          ;GET THE DATA BUS STATUS INFO
1854  4442  E6  02                 7.0           ANI     T.SCLK           ;SEE IF SCLK IS GONE YET
1855  4444  CA  5F  44      10.0           JZ      SCKGON           ;YUP - PROCEED
1856  4447  0D                     4.0           DCR     C                ;NO - SEE IF DONE WITH LOW COUNTER
1857  4448  C2  33  44      10.0           JNZ     SCKWT1           ;BACK TO LOOP
1858  444B  05                     4.0           DCR     B                ;YES! - DOWNCOUNT WATCHDOG
1859  444C  C2  31  44      10.0           JNZ     SCKWT            ;STAY IN LOOP TILL NO 'SCLK' OR DIE
1860  444F                                 ERR     SCKWTE,SCKWTC
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  444F  CD  09  28      18.0                    CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0003                          MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4452  03                                      .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  4453  00                                      .BYTE
 (1)  4454  CD  15  28      18.0  SCKWTC::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4457  DA  5D  44      10.0                  JC      SCKWTE           ;LOOP ADDRESS IF LOOP SPECIFIED
1861                                        ;>WMC FAILED TO DROP 'SCLK' BIT AFTER SEVERAL SYSTEM CLOCKS
1862                                        ;<TEST ABORTED.
1863  445A  C3  18  28      10.0           JMP     TSTEND           ;EXIT IF NOT LOOPING ON ERROR
1864
1865  445D  C1                    10.0  SCKWTE: POP     B
1866  445E  C9                    10.0           RET
1867  445F  C1                    10.0  SCKGON: POP     B                ;ALL OK - NOW RESET REG B & C
1868  4460  C2  66  44      10.0           JNZ     SCKB1            ;JMP IF NOT TIME TO RELOAD DDR
1869
1870  4463  CD  6C  48      18.0           CALL    GENDAT           ;GENERATE NEW DDR DATA FOR INJECTION
1871
1872  4466  06  7F                 7.0  SCKB1:  MVI     B,@177           ;GET NEW TIMEOUT COUNT
1873  4468  78                     4.0           MOV     A,B              ;COPY TO REG A
1874  4469  32  50  48      13.0           STA     CLKON            ;SET SCLK "ON" FLAG
1875  446C  3A  52  48      13.0           LDA     CLKCNT           ;GET THE CURRENT SCLK COUNT
1876  446F  3C                     4.0           INR     A                ;UPDATE IT
1877  4470  32  52  48      13.0           STA     CLKCNT           ;SAVE THE NEW COUNT
1878  4473  C9                    10.0           RET
```

```
1880  4474                                  S
 (1)                                        ; ***************************************************************************
1881                                        ;CKSCLK -- ROUTINE TO CLOCK THE SYSTEM ONCE AND THEN CHECK FOR PRESENCE
1882                                        ;       OF 'SCLK' FROM THE WMC. IF 'SCLK' IS ON, THEN LOAD NEW DATA INTO
1883                                        ;       THE DDR.
1884  4474                                  S
 (1)                                        ; ***************************************************************************
1885
1886  4474  CD  E9  48      18.0  CKSCLK: CALL    CLKSYS          ;CLOCK SYSTEM ONCE
1887  4477  DB  C0            10.0        IN      DBUSSTA         ;GET STATUS OF DATA BUS
1888  4479  E6  02             7.0        ANI     T.SCLK          ;SAVE ONLY THE 'SCLK' BIT
1889  447B  C2  83  44       10.0        JNZ     CKSC1           ;JUMP IF SCLK IS ON
1890  447E  AF                4.0        XRA     A               ;NO SCLK - CLEAR CLOCK ON FLAG
1891  447F  32  50  48       13.0        STA     CLKON
1892  4482  C9               10.0        RET                     ;EXIT
1893
1894  4483  CD  26  44       18.0  CKSC1:  CALL    SCLKBIT         ;LOAD NEW DATA INTO DDR REGS
1895  4486  DA  1F  43       10.0        JC      TST1L
1896  4489  C9               10.0        RET                     ;EXIT
1897
1898  448A                                  S
 (1)                                        ; ***************************************************************************
1899                                        ;XFRBIT -- HERE WHEN WMC STATUS INDICATES 'XFER' BIT IS SET. ISSUE 20
1900                                        ;       MORE CLOCKS THEN CHECK WRITE DATA WORD.
1901  448A                                  S
 (1)                                        ; ***************************************************************************
1902
1903  448A  DB  D0            10.0  XFRBIT: IN      WMCSTA          ;GET THE CURRENT WMC STATUS
1904  448C  32  59  48       13.0        STA     SWMCSTA         ;SAVE IT FOR THE ECC CHECK (LATER)
1905  448F  06  14             7.0        MVI     B,20            ;ALLOW UP TO 20. WMC CLOCKS TO DROP XFER
1906  4491  CD  74  44       18.0  XFRB1:  CALL    CKSCLK          ;NEED TWO MORE CLOCK CYCLES BEFORE
1907  4494  DA  1F  43       10.0        JC      TST1L           ;JUMP IF ERROR DETECTED
1908  4497  DB  D0            10.0        IN      WMCSTA          ;GET WMC STATUS
1909  4499  E6  20             7.0        ANI     W.XFER          ;SAVE THE XFER BIT
1910  449B  CA  A7  44       10.0        JZ      XFRGON          ;OK TO CONTINUE IF XFER IS GONE
1911  449E  05                4.0        DCR     B               ;DECREMENT WATCHDOG COUNTER
1912  449F  C2  91  44       10.0        JNZ     XFRB1           ;LOOP TIL =0 OR XFER IS GONE
1913  44A2                                ERR     TST1L,XFRGON
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44A2  CD  09  28       18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0004                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44A5  04                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44A6  00                                       .BYTE
 (1)  44A7  CD  15  28       18.0        XFRGON::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44AA  DA  1F  43       10.0                    JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1914                                    ;>WMC FAILED TO CLEAR 'XFER' BIT ON 1ST NIBBLE OF THIS BYTE ASSY
1915
1916
1917                                    ;HERE TO LOOK FOR 'XFER' TO SET FOR THE 2ND NIBBLE OF THIS BYTE
1918
1919  44AD  06  7F             7.0        MVI     B,@177
1920  44AF  CD  74  44       18.0  XFRB2:  CALL    CKSCLK          ;SEE IF SCLK IS SET (ALSO CLOCK)
1921  44B2  DA  1F  43       10.0        JC      TST1L           ;JUMP IF ERROR DETECTED
1922  44B5  DB  D0            10.0        IN      WMCSTA          ;GET WMC STATUS
```

```
1923  44B7  E6 20          7.0          ANI     W.XFER          ;SAVE ONLY XFER BIT
1924  44B9  C2 CE 44       10.0         JNZ     XFRB3           ;JUMP IF XFER IS SET - OK
1925  44BC  05             4.0          DCR     B               ;LOOK FOR TIMEOUT
1926  44BD  C2 AF 44       10.0         JNZ     XFRB2           ;GO BACK TO LOOP IF OK
1927  44C0                               ERR     TST1L,XFRB2C,3
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44C0  CD 09 28       18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0005                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44C3  05                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44C4  03                                .BYTE   3
 (1)  44C5  CD 3B 28       18.0         XFRB2C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44C8  DA 1F 43       10.0                 JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1928                                    ;>WMC FAILED TO SET 'XFER' BIT FOR 2ND NIBBLE ON THIS BYTE ASSY
1929                                    ;<TEST ABORTED!
1930  44CB  C3 48 28       10.0                 JMP     TSTEND
1931
1932                                    ;HERE TO LOOK FOR 'XFER' BIT TO GO AWAY TO INDICATE FULL BYTE IS AVAIL.
1933                                    ;FROM THE 'WRTDAT' WORD
1934
1935  44CE  3A 6B 48       13.0  XFRB3:  LDA     XFRREC          ;GET CURRENT COUNT OF 'XFER' BYTES
1936  44D1  3C             4.0          INR     A               ;COUNT THIS ONE
1937  44D2  32 6B 48       13.0         STA     XFRREC          ;SAVE UPDATED COUNT
1938  44D5  06 14          7.0          MVI     B,20
1939  44D7  DB E0          10.0         IN      INTSTA          ;GET THE DATA PARITY BIT
1940  44D9  32 5B 48       13.0         STA     SINTSTA         ;SAVE FOR PARITY CHECK
1941  44DC  CD 74 44       18.0  XFRB3A: CALL    CKSCLK          ;CLOCK THE SYSTEM TILL XFER BIT GOES AWAY
1942  44DF  DA 1F 43       10.0         JC      TST1L           ;JUMP IF ERROR DETECTED
1943  44E2  DB D0          10.0         IN      WMCSTA          ;GET THE WMC STATUS
1944  44E4  E6 20          7.0          ANI     W.XFER          ;SAVE ONLY THE XFER BIT
1945  44E6  CA FB 44       10.0         JZ      XFRB4           ;JUMP IF GONE - OK
1946  44E9  05             4.0          DCR     B               ;DOWNCOUNT WATCHDOG
1947  44EA  C2 DC 44       10.0         JNZ     XFRB3A          ;BACK TO LOOP IF OK
1948  44ED                               ERR     TST1L,XFRB3C,3
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44ED  CD 09 28       18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0006                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44F0  06                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44F1  03                                .BYTE   3
 (1)  44F2  CD 15 28       18.0         XFRB3C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44F5  DA 1F 43       10.0                 JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1949                                    ;>WMC FAILED TO CLEAR 'XFER' BIT FOR 2ND NIBBLE OF THIS BYTE ASSY
1950                                    ;<TEST ABORTED!
1951  44F8  C3 18 28       10.0                 JMP     TSTEND
1952                                    ;HERE TO TEST THE ASSEMBLED BYTE DATA AGAINST THE EXPECTED BYTE DATA
1953
1954  44FB  CD E9 48       18.0  XFRB4:  CALL    CLKSYS          ;ALLOW THE BYTE COUNTER TO TICK
1955  44FE  CD E9 48       18.0          CALL    CLKSYS          ;TO LATCH IT.
```

```
1957  4501                                  S
 (1)                                        ; ***************************************************************************
1958                                        ;XFRRDY -- ROUTINE TO DETERMINE WHAT CHARACTER IS ON WMC OUTPUT WHEN
1959                                        ;       'XFER' IS SET.  ALSO, CHECKS FOR ERRORS BASED ON BYTE COUNTER.
1960  4501                                  S
 (1)                                        ; ***************************************************************************
1961
1962  4501  3A  59  48      13.0  XFRRDY: LDA     SWMCSTA         ;GET WMC STATUS WORD
1963  4504  E6  1E            7.0         ANI     @36             ;SAVE JUST THE SPECIAL CHAR INFO.
1964  4506  CA  45  45        10.0         JZ      XFRB4D          ;NOTHING SPECIAL - CHECK DATA BYTE
1965
1966  4509  3A  59  48      13.0         LDA     SWMCSTA         ;GET THE STATUS AGAIN
1967  450C  E6  10            7.0         ANI     W.ECC           ;SEE IF ITS 'ECC' CHARACTER
1968  450E  C2  57  47        10.0         JNZ     CKECC           ;JUMP IF ITS 'ECC' - CHECK PE ECC CHAR
1969
1970  4511  3A  59  48      13.0         LDA     SWMCSTA         ;GET WMC STATUS WORD AGAIN
1971  4514  E6  04            7.0         ANI     W.ACRC          ;SEE IF 'ACRC' CHARACTER READY
1972  4516  CA  27  45        10.0         JZ      XFRDY2
1973  4519                                ERR     TST1L,XFRDY1,3
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4519  CD  09  28        18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  451C  07                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  451D  03                                .BYTE   3
 (1)  451E  CD  15  28        18.0         XFRDY1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4521  DA  1F  43        10.0                 JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1974                                    ;>WMC PRESENTED AN 'ACRC' CHARACTER IN 'PE' MODE - ILLFGAL
1975  4524  C3  2F  46        10.0         JMP     SCHREX
1976
1977  4527  3A  59  48      13.0  XFRDY2: LDA     SWMCSTA         ;GET STATUS AGAIN
1978  452A  E6  08            7.0         ANI     W.CRC           ;SEE IF ITS 'CRC' READY BIT
1979  452C  C2  CC  47        10.0         JNZ     CKCRC           ;GO TEST 'CRC' DATA BYTE
1980
1981  452F  3A  59  48      13.0         LDA     SWMCSTA         ;GET STATUS AGAIN
1982  4532  E6  02            7.0         ANI     W.RESID         ;SEE IF 'RESID' READY BIT (IF NOT ITS FATAL!)
1983  4534  CA  45  45        10.0         JZ      XFRB4D
1984  4537                                ERR     TST1L,XFRDY3,3
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4537  CD  09  28        18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0008                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  453A  08                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  453B  03                                .BYTE   3
 (1)  453C  CD  15  28        18.0         XFRDY3::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  453F  DA  1F  43        10.0                 JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
1985                                    ;>WMC PRESENTED 'RESID' BYTE IN 'PE' MODE - ILLEGAL
1986  4542  C3  2F  46        10.0         JMP     SCHREX
1987
1988
1989                                    ;HERE TO CHECK THE DATA BYTE JUST ASSEMBLED (COULD BE A 'PAD' BYTE)
1990
1991  4545  3A  51  48      13.0  XFRB4D: LDA     ECCCNT          ;GET 'XFER' COUNT
1992  4548  3C                4.0         INR     A               ;ADD 1
1993  4549  32  51  48        13.0         STA     ECCCNT          ;SAVE THE UPDATED COUNT
1994  454C  E5                12.0         PUSH    H               ;SAVE H&L
```

```
1995  454D  21  66  48  10.0          LXI   H,TPBCNT          ;POINT TO THE AMT OF BYTES EXPECTED
1996  4550  3A  4C  48  13.0          LDA   ABCNT             ;THEN GET # BYTES ASSEMBLED BY THE WMC
1997  4553  BE              7.0      CMP   M                 ;SEE IF DONE
1998  4554  E1             10.0      POP   H                 ;RESET REG H & L
1999  4555  DA  41  46     10.0      JC    XFRB4C            ;CONTINUE TO COMPARE DATA BYTES
2000
2001
2002                              ;HERE TO CHECK THE PAD CHARACTERS EXPECTED
2003
2004  4558  3A  5C  48  13.0  CKPAD:  LDA   PADEXP           ;GET THE PAD COUNT EXPECTED
2005  455B  47               4.0      MOV   B,A               ;COPY THE COUNT
2006
2007  455C  3A  63  48  13.0  CKPADA: LDA   RPADCT           ;GET THE RECEIVED PAD COUNT #
2008  455F  B8               4.0      CMP   B                 ;SAME AS EXPECTED (TIME FOR PHANTOM XFR)
2009  4560  CA  0E  46  10.0          JZ    CKHACK            ;GO CHECK FOR THE PHANTOM 'XFR'
2010  4563  3A  63  48  13.0          LDA   RPADCT            ;GET THE PAD CHARACTERS COUNT
2011  4566  3C               4.0      INR   A                 ;UPDATE FOR THIS CHARACTER
2012  4567  32  63  48  13.0          STA   RPADCT            ;SAVE IT
2013  456A  AF               4.0      XRA   A                 ;CLEAR THE ACCUMULATOR
2014  456B                             ROUT  EDATA             ;STORE THE EXPECTED PAD CHARACTER
 (1)  456B  D3  95            10.0                OUT   EDATA          ;WRITE AC INTO EDATA
 (1)  456D  7F                4.0                 MOV   A,A            ;RETRY LINK
2015  456E  DB  D3            10.0      IN    WRTDAT            ;GET THE ACTUAL PAD CHARACTER
2016  4570                             ROUT  ADATA             ;STORE THE ACTUAL PAD CHARACTER
 (1)  4570  D3  94            10.0                OUT   ADATA          ;WRITE AC INTO ADATA
 (1)  4572  7F                4.0                 MOV   A,A            ;RETRY LINK
2017  4573  A7               4.0      ANA   A                 ;SET THE CONDITION BITS
2018  4574  CA  7C  45  10.0          JZ    PADCNO            ;CONTINUE IF ZERO
2019  4577                             ERPB  TST1L,PADCNO
 (1)                             ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4577  CD  12  28  18.0                  CALL  ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0009                               MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  457A  09                                 .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  457B  00                                 .BYTE                 ;PRINT ROUTINE NUMBER
 (1)  457C  CD  15  28  18.0          PADCNO::       CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  457F  DA  1F  43  10.0                  JC    TST1L             ;LOOP ADDRESS IF LOOP SPECIFIED
2020                          ;>PAD CHARACTER BYTE INCORRECT
2021
```

```
2023
2024                                    ;HERE TO CHECK THE PARITY BIT ON THE PAD ASSEMBLED
2025
2026  4582  DB  D3          10.0        IN      WRTDAT          ;GET THE DATA BYTE AGAIN
2027  4584                              ROUT    ADATA           ;SAVE AS ACTUAL DATA
 (1)  4584  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4586  7F               4.0                 MOV     A,A             ;RETRY LINK
2028  4587  A7               4.0         ANA     A               ;SET THE CONDITION BITS
2029  4588  E4  9A  45       18.0        CPO     PADPO           ;CHECK FOR ODD PARITY
2030  458B  DA  1F  43       10.0        JC      TST1L           ;JUMP IF ERROR DETECTED
2031  458E  DB  D3           10.0        IN      WRTDAT          ;GET THE DATA BYTE AGAIN
2032  4590  A7               4.0         ANA     A               ;SET THE CONDITION BITS
2033  4591  EC  AE  45       18.0        CPE     PADPE           ;CHECK FOR PARITY EVEN - BIT =1
2034  4594  DA  1F  43       10.0        JC      TST1L           ;JUMP IF ERROR DETECTED
2035  4597  C3  C0  45       10.0        JMP     CKPAD1          ;ON TO THE NEXT SECTION....
2036
2037                                    ;HERE TO CHECK FOR PARITY ODD - BIT =0
2038
2039  459A  3A  5B  48       13.0  PADPO:  LDA     SINTSTA         ;GET THE PARITY BIT (IN BIT 0)
2040  459D  E6  10            7.0         ANI     WDR.P           ;SAVE THE PARITY BIT
2041  459F  CA  AD  45       10.0        JZ      PADPOE          ;EXIT IF OK
```

```
2043  45A2                                        ERRA    PADPOE,PADPOC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  45A2  CD  0F  28        18.0                        CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       000A                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45A5  0A                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45A6  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45A7  CD  15  28        18.0               PADPOC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45AA  DA  AD  45        10.0                        JC      PADPOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2044                                        ;>PARITY BIT ON PAD SHOULD =0 AND WAS =1
2045
2046  45AD  C9                10.0  PADPOE: RET
2047
2048                                        ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
2049
2050  45AE  3A  5B  48        13.0  PADPE:  LDA     SINTSTA         ;GET THE BIT
2051  45B1  E6  10             7.0          ANI     WDR.P           ;SAVE THE PARITY BIT
2052  45B3  C0                12.0          RNZ
2053  45B4                                        ERRA    PADPEE,PADPEC
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  45B4  CD  0F  28        18.0                        CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       000B                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45B7  0B                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45B8  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45B9  CD  15  28        18.0               PADPEC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45BC  DA  BF  45        10.0                        JC      PADPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
2054                                        ;>PARITY BIT ON PAD WAS =0 AND SHOULD BE =1
2055
2056  45BF  C9                10.0  PADPEE: RET
2057
2058  45C0  DB  D3            10.0  CKPAD1: IN      WRTDAT          ;GET THE PAD CHARACTER AGAIN
2059  45C2  CD  FF  48        18.0          CALL    GENECC          ;GO INCLUDE THE PAD CHARACTER IN THE ECC
2060
2061  45C5  3A  5C  48        13.0          LDA     PADEXP          ;GET THE EXPECTED COUNT OF PAD CHAR
2062  45C8  47                 4.0          MOV     B,A             ;SAVE
2063  45C9  3A  63  48        13.0          LDA     RPADCT          ;GET ACTUAL RECEIVED PAD COUNT
2064  45CC  B8                 4.0          CMP     B               ;SEE IF ALL RECEIVED
2065  45CD  C2  2F  46        10.0          JNZ     SCHREX          ;EXIT IF NOT DONE
2066
2067  45D0  06  19             7.0  XMCCK:  MVI     B,25            ;SET COUNTER FOR TRANSLATOR TO FINISH
2068
2069  45D2  0E  FF             7.0  XMCK1A: MVI     C,@377
2070
2071  45D4  CD  74  44        18.0  XMCK1:  CALL    CKSCLK          ;ISSUE SINGLE STEP CLOCK
2072  45D7  DA  1F  43        10.0          JC      TST1L           ;JUMP BACK IF ERROR DETECTED
2073  45DA  DB  D0            10.0          IN      WMCSTA          ;GET WMC STATUS WORD
2074  45DC  E6  20             7.0          ANI     W.XFER          ;SAVE ONLY 'XFER' BIT
2075  45DE  C2  8A  44        10.0          JNZ     XFRBIT          ;JUMP IF 'XFER' IS SET - DATA READY IN WRITE DATA WORD
2076  45E1  3A  5A  48        13.0          LDA     SXFER           ;GET THE SOFTWARE XFER FLAG
2077  45E4  A7                 4.0          ANA     A
2078  45E5  CA  F1  45        10.0          JZ      XMCK2           ;JUMP IF XFER NOT SET
2079  45E8  AF                 4.0          XRA     A               ;CLEAR SOFTWARE FLAG
2080  45E9  32  5A  48        13.0          STA     SXFER           ;FOR NEXT PASS
2081  45EC  06  7F             7.0          MVI     B,@177          ;SETUP A WATCHDOG COUNT
2082  45EE  C3  AF  44        10.0          JMP     XFRB2           ;PROCESS THE 'XFER' BIT DETECTED
```

```
2083
2084  45F1  0D                    4.0  XMCK2:  DCR   C               ;DOWNCOUNT FOR XMC
2085  45F2  C2  D4  45           10.0          JNZ   XMCK1           ;CLOCK AGAIN
2086
2087  45F5  05                    4.0  XMCK3:  DCR   B               ;DOWNCOUNT GROSS COUNT
2088  45F6  C2  D2  45           10.0          JNZ   XMCK1A          ;RESET OTHER COUNTER AND CONTINUE
2089
2090  45F9  3A  68  48           13.0          LDA   LASTCRC         ;TIME TO EXIT?
2091  45FC  A7                    4.0          ANA   A
2092  45FD  C2  ED  46           10.0          JNZ   QUIT            ;JUMP IF ALL DONE
2093  4600                                      ERR   TST1L,XMCK3C
 (1)                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4600  CD  09  28           18.0                CALL  ERLP              ;PROCESS ERROR - DO 2.3
 (1)        000C                              MSGN  =     MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4603  0C                              .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  4604  00                              .BYTE
 (1)  4605  CD  15  28           18.0  XMCK3C::     CALL  CKLOP             ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4608  DA  1F  43           10.0                JC    TST1L             ;LOOP ADDRESS IF LOOP SPECIFIED
2094                          ;>XMC FAILED TO FINISH CLOCKING SO NEXT WMC CHAR CAN BE SENT
2095                          ;<TEST ABORTED!
2096  460B  C3  18  28           10.0          JMP   TSTEND
2097
2098
2099                          ;HERE IF A 'XFER' DATA BYTE WAS DETECTED AND ALL DATA BYTES WERE
2100                          ;RECEIVED AND ALL PAD BYTES REVEIVED...I.E. WE ARE ABOUT TO GIVE
2101                          ;AN EXTRA CLOCK TO THE CRC GENERATOR (CRC GROUP HAS 9 CHAR).
2102
2103  460E  3A  65  48           13.0  CKHACK: LDA   CLKCRC          ;GET THE EXTRA CLOCK-TO-CRC FLAG
2104  4611  A7                    4.0          ANA   A               ;SEE IF HERE BEFORE
2105  4612  C2  24  46           10.0          JNZ   XFRERR          ;YES - THEN NO MORE PHANTOM 'XFR'S' ALLOWED
2106  4615  3E  FF                7.0          MVI   A,@377          ;NO - THEN ALLOW ONLY THE ONE
2107  4617  32  65  48           13.0          STA   CLKCRC          ;UPDATE THE RECEIVED FLAG
2108  461A  3A  51  48           13.0          LDA   ECCCNT          ;GET THE ECC CHAR. COUNTER
2109  461D  3D                    4.0          DCR   A               ;ADJUST FOR ADDITION OF THE PHANTOM 'XFR'
2110  461E  32  51  48           13.0          STA   ECCCNT          ;RESTORE ADJUSTED COUNT
2111  4621  C3  2F  46           10.0          JMP   SCHREX          ;CONTINUE TO CLOCK THE WMC
2112
2113  4624                                XFRERR: ERR   TST1L,XFB4C
 (1)                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4624  CD  09  28           18.0                CALL  ERLP              ;PROCESS ERROR - DO 2.3
 (1)        000D                              MSGN  =     MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4627  0D                              .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  4628  00                              .BYTE
 (1)  4629  CD  15  28           18.0  XFB4C::     CALL  CKLOP             ;CHECK LOOP FUNCTION - DO 2.3
 (1)  462C  DA  1F  43           10.0                JC    TST1L             ;LOOP ADDRESS IF LOOP SPECIFIED
2114                          ;>WMC SET 'XFER' ONLY - AFTER ALL DATA AND PAD CHARACTERS OUTPUT
2115
2116  462F  06  7F                7.0  SCHREX: MVI   B,@177          ;RESET WATCHDOG TIMERS
2117  4631  48                    4.0          MOV   C,B
2118  4632  3A  6B  48           13.0          LDA   XFRREC          ;GET COUNT OF RECEIVED 'XFER' CHAR
2119  4635  FE  04                7.0          CPI   4               ;TIME TO EXTEND THE COUNT?
2120  4637  DA  FB  43           10.0          JC    TST1C           ;NO - BACK TO NORMAL TIMEOUT LOOP
2121  463A  AF                    4.0          XRA   A               ;YUP
2122  463B  32  6B  48           13.0          STA   XFRREC          ;CLEAR THE COUNTER
```

```
2123  463E  C3  DC  45      10.0          JMP     XMCCK           ;ENTER SPECIAL 'HOLDING' LOOP
2124
2125                                ;GET ASSEMBLED BAL DATA BYTE FROM WRITE DATA WORDS
2126
2127  4641  3A  4C  48      13.0  XFRB4C: LDA     ABCNT           ;GET THE BYTE # UNDER TEST
2128  4644  3C                4.0          INR     A               ;ADD 1
2129  4645  32  4C  48      13.0          STA     ABCNT           ;SAVE THE BYTE NUMBER - UPDATED
2130  4648                              ROUT    RO5L
 (1)  4648  D3  8A            10.0                  OUT     RO5L            ;WRITE AC INTO RO5L
 (1)  464A  7F                4.0                  MOV     A,A             ;RETRY LINK
2131  464B  AF                4.0          XRA     A
2132  464C                              ROUT    RO5H
 (1)  464C  D3  8B            10.0                  OUT     RO5H            ;WRITE AC INTO RO5H
 (1)  464E  7F                4.0                  MOV     A,A             ;RETRY LINK
2133  464F  EB                4.0          XCHG                    ;PUT 'EXPECTED' POINTER IN H & L
2134  4650  7E                7.0          MOV     A,M             ;GET CORRECT BAL BYTE
2135  4651                              ROUT    EDATA           ;SAVE 'EXPECTED' DATA
 (1)  4651  D3  95            10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4653  7F                4.0                  MOV     A,A             ;RETRY LINK
2136  4654  DB  D3            10.0          IN      WRTDAT          ;GET ASSEMBLED DATA BYTE
2137  4656                              ROUT    ADATA           ;SAVE 'ACTUAL' DATA
 (1)  4656  D3  94            10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4658  7F                4.0                  MOV     A,A             ;RETRY LINK
2138  4659  BE                7.0          CMP     M               ;EXPECTED = ACTUAL?
2139  465A  EB                4.0          XCHG                    ;RESET H & L REGS
2140  465B  13                6.0          INX     D               ;POINT TO NEXT 'EXPECTED' ENTRY
2141  465C  CA  6A  46        10.0          JZ      TST1XC          ;JUMP OVER ERROR IF DATA BYTE OK
2142  465F  3A  4C  48        13.0          LDA     ABCNT           ;GET THE COUNT OF ASSEM. BYTES AT FAILURE
2143  4662  32  4D  48        13.0          STA     EABCNT          ;SAVE FOR THE LOOP CONTROL
2144  4665                              ERRB    DELOOP,TST1XC,3
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4665  CD  12  28      1  0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000E                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4668  0E                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4669  03                                       .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  466A  CD  15  28      18.0                TST1XC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  466D  DA  3C  47      10.0                        JC      DELOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2145                                ;>WMC "PE" BYTE ASSEMBLY FAULT - BYTE ASSEMBLED DOESN'T MATCH EXPECTED DATA
2146
2147                                ;HERE TO CHECK THE PARITY BIT ON THE DATA ASSEMBLED
2148
2149  4670  DB  D3            10.0  XFRB4P: IN      WRTDAT          ;GET BYTE ON WMC OUTPUT
2150  4672                              ROUT    ADATA           ;SAVE IT
 (1)  4672  D3  94            10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4674  7F                4.0                  MOV     A,A             ;RETRY LINK
2151  4675  A7                4.0          ANA     A
2152  4676  E4  88  46        18.0          CPO     DATPO           ;CHECK FOR ODD PARITY ON DATA
2153  4679  DA  1F  43        10.0          JC      TST1L           ;ERROR RETURN
2154
2155  467C  DB  D3            10.0          IN      WRTDAT          ;GET DATA AGAIN
2156  467E  A7                4.0          ANA     A
2157  467F  EC  9C  46        18.0          CPE     DATPE           ;CHECK FOR EVEN PARITY DATA
2158  4682  DA  1F  43        10.0          JC      TST1L           ;JUMP IF ERROR DETECTED
2159  4685  C3  AE  46        10.0          JMP     XFRB5           ;ON TO THE NEXT SECTION....
```

```
2160
2161                                        ;HERE IS THE ODD PARITY DATA CHECK
2162
2163  4688  3A 5B 48      13.0  DATPO:  LDA     SINTSTA         ;GET COPY OF INTSTA WORD
2164  468B  E6 10            7.0         ANI     WDR.P           ;SAVE THE PARITY BIT
2165  468D  CA 9B 46        10.0         JZ      DATPOE          ;EXIT IF ODD PARITY ON DATA OK
2166  4690                               ERRA    DATPOE,DATPOC
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4690  CD 0F 28      18.0                       CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       000F                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4693  0F                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4694  00                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4695  CD 15 28      18.0               DATPOC::         CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4698  DA 9B 46      10.0                        JC      DATPOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2167                                        ;>PARITY BIT ON DATA BYTE SHOULD =0 AND WAS =1
2168
2169  469B  C9               10.0  DATPOE: RET
2170
2171                                        ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
2172
2173  469C  3A 5B 48      13.0  DATPE:  LDA     SINTSTA         ;GET THE BIT
2174  469F  E6 10            7.0         ANI     WDR.P           ;SAVE THE PARITY BIT
2175  46A1  C0              12.0         RNZ
2176  46A2                               ERRA    DATPEE,DATPEC
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  46A2  CD 0F 28      18.0                       CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0010                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46A5  10                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46A6  00                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  46A7  CD 15 28      18.0               DATPEC::         CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46AA  DA AD 46      10.0                        JC      DATPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
2177                                        ;>PARITY BIT ON DATA WAS =0 AND SHOULD BE =1
2178
2179  46AD  C9               10.0  DATPEE: RET
2180
2181  46AE  DB D3            10.0  XFRB5:  IN      WRTDAT          ;GET THE DATA AGAIN
2182  46B0  CD FF 48         18.0         CALL    GENECC          ;CALCULATE THE ECC
2183  46B3  CD FE 47         18.0         CALL    GENCRC          ;ADD DATA TO CRC CHAR
2184
2185  46B6  3A 4B 48      13.0         LDA     IMBYTC          ;GET LOADED BYTE COUNT
2186  46B9  3D               4.0         DCR     A               ;TEST TO SEE IF > 1
2187  46BA  CA 2F 46        10.0         JZ      SCHREX          ;DON'T TEST BYTE COUNT =0 IF LOADED WITH 1
2188
2189  46BD  E5              12.0         PUSH    H
2190  46BE  21 4C 48         10.0         LXI     H,ABCNT         ;POINT TO # ASSEMB. BYTES
2191  46C1  3A 69 48      13.0         LDA     MOD7R           ;GET MOD7 REMAINDER
2192  46C4  47               4.0         MOV     B,A             ;TEMP STORE IT
2193  46C5  3A 66 48      13.0         LDA     TPBCNT          ;GET # EXPECTED BYTES
2194  46C8  90               4.0         SUB     B               ;CALC WHEN BYTE COUNTER SHOULD RUN OUT
2195  46C9  BE               7.0         CMP     M               ;SEE IF BYTE COUNTER SHOULD =0
2196  46CA  E1              10.0         POP     H               ;RESET REG H & L
2197  46CB  C2 2F 46        10.0         JNZ     SCHREX          ;BACK TO LOOP IF MORE BYTES TO DO
2198
2199  46CE  DB D4            10.0         IN      BYTCNT          ;DUMMY READ OF BYTE COUNTER
```

```
2200  46D0  DB  D4          10.0          IN      BYTCNT
2201
2202                                ;TIME TO CHECK THE BYTE COUNTER...
2203
2204  46D2  DB  D4          10.0          IN      BYTCNT          ;GET BYTE COUNTER 7-0
2205  46D4  47               4.0          MOV     B,A             ;TEMP STORE
2206  46D5  DB  D4          10.0          IN      BYTCNT          ;GET BYTE CNT 15-8
2207  46D7  78               4.0          MOV     A,B
2208  46D8  A7               4.0          ANA     A
2209  46D9  CA  E4  46       10.0          JZ      XFRB5C          ;JUMP IF BYTE COUNT OK
2210  46DC                                ROUT    ADATA           ,STORE THE ACTUAL BYTE COUNTER 7-0
 (1)  46DC  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46DE  7F               4.0                  MOV     A,A             ;RETRY LINK
2211  46DF                                ERRA    TST1L,XFRB5C,3
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  46DF  CD  0F  28       18.0                         CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0011                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46E2  11                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46E3  03                                           .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  46E4  CD  15  28       18.0                 XFRB5C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46E7  DA  1F  43       10.0                         JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2212                                ;>WMC BYTE COUNTER EXPECTED TO BE = 0 - ALL DATA BYTES RECEIVED FROM WMC
2213
2214
2215  46EA  C3  2F  46       10.0          JMP     SCHREX          ;BACK TO CLOCK LOOP FOR OTHER CHARACTERS
2216
2217  46ED                              S
 (1)                                  ; *************************************************************************
2218                                  ;HERE IF ALL DATA, ECC, CRC
2219  46ED                              S
 (1)                                  ; *************************************************************************
2220
2221  46ED  DB  D3          10.0  QUIT:   IN      WRTDAT          ;GET ACTUAL WMC 'CRC' CHARACTER
2222  46EF                              ROUT    ADATA
 (1)  46EF  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46F1  7F               4.0                  MOV     A,A             ;RETRY LINK
2223  46F2  21  5F  48       10.0          LXI     H,CRCCHR        ;POINT TO EXPECTED CRC CHAR (SIMULATED)
2224  46F5  BE               7.0          CMP     M               ;ACTUAL=SIMULATED?
2225  46F6  CA  02  47       10.0          JZ      QUIT1           ;JUMP IF OK
2226  46F9  7E               7.0          MOV     A,M             ;GET SIMULATED CRC CHAR.
2227  46FA                              ROUT    EDATA
 (1)  46FA  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  46FC  7F               4.0                  MOV     A,A             ;RETRY LINK
2228  46FD                              ERRB    TST1L,QUIT1
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46FD  CD  12  28       18.0                         CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0012                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4700  12                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4701  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4702  CD  15  28       18.0                 QUIT1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4705  DA  1F  43       10.0                         JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2229                                ;>SPECIAL 'PE' TERMINATING 'CRC' CHARACTER INCORRECT
2230
2231
```

```
2232
2233  4708  DB  D0          10.0  QUIT4:  IN      WMCSTA          ;GET THE WMC STATUS
2234  470A                        ROUT    ADATA           ;SAVE THE ACTUAL STATUS FOR PRINTOUT
 (1)  470A  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  470C  7F               4.0                MOV     A,A             ;RETRY LINK
2235  470D  E6  40           7.0          ANI     W.DONN          ;DONE?
2236  470F  CA  17  47       10.0          JZ      QUIT3           ;YES - EXIT THE TEST
2237  4712                        ERRA    TST1L,QUIT3
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4712  CD  0F  28       18.0                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0013                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4715  13                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4716  00                               .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4717  CD  15  28       18.0          QUIT3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  471A  DA  1F  43       10.0                JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
2238                        ;>WMC DONE STATUS NOT INDICATED AFTER LAST WMC FUNCTION PERFORMED
2239
2240
2241  471D  3A  66  48       13.0          LDA     TPBCNT          ;GET THE CURRENT BYTE COUNT USED
2242  4720  FE  26            7.0          CPI     DTEND-CDATBL    ;DONE WITH ALL COMBINATIONS?
2243  4722  C2  0E  43       10.0          JNZ     TST1NBC         ;JUMP IF NOT AT END OF ROPE...
2244
2245  4725                        ENDTST  TST1ST
 (1)                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4725                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4725  CD  06  28       18.0                CALL    REQST
 (2)  4728  00                               .BYTE                   ;DATA PATTERN NUMBER
 (2)  4729  00  00                           .WORD                   ;SYSTEM "" COUNT
 (2)  472B  00  00                           .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  472D  00                               .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  472E  07                               .BYTE   7               ;REQUEST CODE
 (1)  472F  3A  9A  4F       13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4732  3D               4.0          DCR     A               ;DOWNCOUNT
 (1)  4733  32  9A  4F       13.0          STA     ITERA           ;SAVE COUNT
 (1)  4736  F2  05  43       10.0          JP      TST1ST          ;DO TEST UNTIL TILL = 0
2246  4739  C3  18  28       10.0          JMP     TSTEND          ;ALL DONE
2247
2248                        ;HERE IS THE DATA COMPARE ERROR LOOP CONTROL ROUTINE
2249
2250                        ;WILL ENABLE THE TEST TO LOOP AT THE SPECIFIED BYTE IN ERROR
2251
2252  473C  3A  94  4F       13.0  DELOOP: LDA     LPFLG           ;SEE IF LOOP FLAG STILL SET
2253  473F  A7               4.0          ANA     A
2254  4740  C2  49  47       10.0          JNZ     DELP1           ;JUMP IF LOOPING ON ERROR
2255  4743  32  4D  48       13.0          STA     EABCNT          ;NOT LOOPING - CLEAR THE FLAG
2256  4746  C3  AE  46       10.0          JMP     XFRB5           ;AND EXIT
2257
2258  4749  3A  4C  48       13.0  DELP1:  LDA     ABCNT           ;GET THE CURRENT COUNT OF ASSEMBLED BYTES
2259  474C  47               4.0          MOV     B,A             ;TEMP STORE
2260  474D  3A  4D  48       13.0          LDA     EABCNT          ;GET THE BYTE COUNT AT THE DETECTED ERROR
2261  4750  B8               4.0          CMP     B               ;SEE IF AT THE SAME BYTE COUNT
2262  4751  CA  1F  43       10.0          JZ      TST1L           ;THEN JUMP TO RESTART TEST IF SAME
2263  4754  C3  AE  46       10.0          JMP     XFRB5           ;ELSE, CONTINUE WITH THE TEST
```

```
                                        .SBTTL  SUBROUTINE CKECC
                                        $
                                        ; ************************************************************************
                                        ;CKECC -- ROUTINE TO CHECK THE 'ECC' CHARACTER PRESENTED BY THE WMC
                                        ;       LOGIC AGAINST A CACULATED ECC BYTE. WILL REPORT AN ERROR IF
                                        ;       THE 'ECC' FLAG (IN WMCSTA WORD) COMES UP BEFORE OR AFTER THE
                                        ;       8TH ASSEMBLED BYTE FROM THE WMC OR IF AN ACTUAL ECC BYTE
                                        ;       ASSEMBLY ERROR WAS DETECTED.
                                        $
                                        ; ************************************************************************

4757  3A  51  48      13.0  CKECC:  LDA     ECCCNT          ;GET 'XFER' CYCLE COUNT
475A                         CKECC:  ROUT    ADATA
475A  D3  94           10.0          OUT     ADATA           ;WRITE AC INTO ADATA
475C  7F                4.0          MOV     A,A             ;RETRY LINK
475D  FE  07            7.0          CPI     7               ;READY FOR ECC CHAR?
475F  CA  6D  47       10.0          JZ      ECCCK           ;JUMP IF EXPECTED NOW
4762                                 ERRA    TST1L,CKECCC,3
                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
4762  CD  0F  28       18.0                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
      0014                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
4765  14                           .BYTE   MSGN            ;MESSAGE NUMBER ID
4766  03                           .BYTE   3               ;PRINT ROUTINE NUMBER
4767  CD  15  28       18.0  CKECCC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
476A  DA  1F  43       10.0                JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                     ;>WMC PRESENTED AN 'ECC' CHAR WHEN NOT EXPECTED - ACTUAL = 'XFER' COUNT


                                     ;HERE TO CHECK THE ACTUAL ECC BYTE FROM THE WMC LOGIC

476D  21  5E  48       10.0  ECCCK:  LXI     H,ECCCHR        ;POINT TO 'IMAGE' OF HARDWARE ECC WORD
4770  7E                7.0          MOV     A,M
4771                         ECCCK:  ROUT    EDATA
4771  D3  95           10.0          OUT     EDATA           ;WRITE AC INTO EDATA
4773  7F                4.0          MOV     A,A             ;RETRY LINK
4774  DB  D3           10.0          IN      WRTDAT          ;GET ACTUAL ECC WORD FROM WMC
4776                                 ROUT    ADATA
4776  D3  94           10.0          OUT     ADATA           ;WRITE AC INTO ADATA
4778  7F                4.0          MOV     A,A             ;RETRY LINK
4779  BE                7.0          CMP     M               ;IS IT OK?
477A  CA  82  47       10.0          JZ      ECCCK1          ;JUMP IF SAME-AS EXPECTED
477D                                 ERRB    TST1L,ECCCK1,3
                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
477D  CD  12  28       18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
      0015                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
4780  15                           .BYTE   MSGN            ;MESSAGE NUMBER ID
4781  03                           .BYTE   3               ;PRINT ROUTINE NUMBER
4782  CD  15  28       18.0  ECCCK1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
4785  DA  1F  43       10.0                JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                     ;>WMC ECC WORD ON 'BAL' OUTPUT NOT AS EXPECTED IN PE MODE

                                     ;HERE TO CHECK THE PARITY BIT ON THE ECC ASSEMBLED

4788  DB  D3           10.0  ECCCK2: IN      WRTDAT          ;GET THE ECC BYTE AGAIN
```

```
2297  478A                                        ROUT    ADATA           ;SAVE AS ACTUAL DATA
 (1)  478A  D3  94            10.0                       OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  478C  7F                 4.0                       MOV     A,A             ;RETRY LINK
2298  478D  A7                 4.0               ANA     A               ;SET THE CONDITION BITS
2299  478E  E4  A0  47         18.0               CPO     ECCPO           ;CHECK FOR ODD PARITY
2300  4791  DA  1F  43         10.0               JC      TST1L           ;JUMP IF ERROR DETECTED
2301  4794  DB  D3              10.0               IN      WRTDAT          ;GET THE ECC BYTE AGAIN
2302  4796  A7                 4.0               ANA     A               ;SET THE CONDITION BITS
2303  4797  EC  B4  47         18.0               CPE     ECCPE           ;CHECK FOR PARITY EVEN - BIT  1
2304  479A  DA  1F  43         10.0               JC      TST1L           ;JUMP IF ERROR DETECTED
2305  479D  C3  C6  47         10.0               JMP     CKECC1          ;ON TO THE NEXT SECTION....
2306
2307                                        ;HERE TO CHECK FOR PARITY ODD - BIT =0
2308
2309  47A0  3A  5B  48         13.0  ECCPO:  LDA     SINTSTA         ;GET THE PARITY BIT (IN BIT 0)
2310  47A3  E6  10              7.0               ANI     WDR.P           ;SAVE THE PARITY BIT
2311  47A5  CA  B3  47         10.0               JZ      ECCPOE          ;EXIT IF OK
2312  47A8                                        ERRA    ECCPOE,ECCPOC
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  47A8  CD  0F  28         18.0                       CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0016                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47AB  16                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47AC  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  47AD  CD  15  28         18.0                ECCPOC::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47B0  DA  B3  47         10.0                       JC      ECCPOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2313                                        ;>PARITY BIT ON ECC SHOULD =0 AND WAS =1
2314
2315  47B3  C9                 10.0  ECCPOE: RET
2316
2317                                        ;HERE TO CHECK FOR PARITY BIT =1 (EVEN)
2318
2319  47B4  3A  5B  48         13.0  ECCPE:  LDA     SINTSTA         ;GET THE BIT
2320  47B7  E6  10              7.0               ANI     WDR.P           ;SAVE THE PARITY BIT
2321  47B9  C0                 12.0               RNZ
2322  47BA                                        ERRA    ECCPEE,ECCPEC
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  47BA  CD  0F  28         18.0                       CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0017                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47BD  17                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47BE  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  47BF  CD  15  28         18.0                ECCPEC::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47C2  DA  C5  47         10.0                       JC      ECCPEE          ;LOOP ADDRESS IF LOOP SPECIFIED
2323                                        ;>PARITY BIT ON ECC WAS =0 AND SHOULD BE =1
2324
2325  47C5  C9                 10.0  ECCPEE: RET
2326  47C6  CD  F5  48         18.0  CKECC1: CALL    CLECC
2327  47C9  C3  2F  46         10.0               JMP     SCHREX          ;GO BACK TO CLOCK LOOP
```

```
                                        .SBTTL  SUBROUTINE CKCRC
                                        $
                                        ; ***************************************************************************
                                        ;CKCRC -- ROUTINE TO CHECK THE WMC 'CRC' DATA BYTE.  CHECKS THAT THE
                                        ;         ACTUAL WMC 'CRC' CHARACTER IS THE SAME AS THE CALCULATED 'CRC'
                                        ;         CHARACTER.
                                        ;
                                        $
                                        ; ***************************************************************************

                                CKCRC:  MVI     B,2             ;SET CRC CHAR LIMIT TO 1
                                CRC2A:  LDA     HAVCRC          ;GET CRC CHAR RECEIVED COUNT
                                        INR     A               ;ADD 1
                                        STA     HAVCRC          ;SAVE UPDATED RECEIVED COUNT
                                        CPI     1               ;1ST CRC CHARACTER?
                                        JNZ     CRC4            ;
                                        STA     LASTCRC         ;SET THE LAST CRC CHAR FLAG
                                        CALL    CRCFIN          ;FINISH CRC CHAR SIMULATION-ONLY ON 1ST CHAR.
                                CRC4:   IN      WRTDAT          ;GET ACTUAL WMC 'CRC' CHARACTER
                                        ROUT    ADATA
                                (1)             OUT     ADATA           ;WRITE AC INTO ADATA
                                (1)             MOV     A,A             ;RETRY LINK
                                        LXI     H,CRCCHR        ;POINT TO EXPECTED CRC CHAR (SIMULATED)
                                        CMP     M               ;ACTUAL=SIMULATED?
                                        JZ      CRC5            ;JUMP IF OK
                                        MOV     A,M             ;GET SIMULATED CRC CHAR.
                                        ROUT    EDATA
                                (1)             OUT     EDATA           ;WRITE AC INTO EDATA
                                (1)             MOV     A,A             ;RETRY LINK
                                        ERRB    TST1L,CRC5
                                (1)             ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
                                (1)                     CALL    ERLPB           ;PROCESS ERROR - DO 2.3
                                (1)                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
                                (1)                     .BYTE   MSGN            ;MESSAGE NUMBER ID
                                (1)                     .BYTE                   ;PRINT ROUTINE NUMBER
                                (1)             CRC5::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
                                (1)                     JC      TST1L           ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>WMC ACTUAL 'CRC' CHARACTER NOT SAME AS SIMULATED 'CRC' CHARACTER
                                        JMP     SCHREX          ;ON TO THE NEXT SECTION....

                                        $
                                        ; ***************************************************************************
                                        ;CRC -- ROUTINE TO SIMULATE THE CRC GENERATOR IN THE TM78.  CALLED
                                        ;       WITH ACCUMULATOR CONTAINING DATA BYTE TO BE ADDED TO CRCWORD
                                        $
                                        ; ***************************************************************************

                                GENCRC: CALL    PARITY          ;CALC. PARITY ON DATA IN ACCUMULATOR
                                        PUSH    PSW
                                        PUSH    B
                                        PUSH    H
                                        LXI     H,CRCCHR        ;POINT TO CRC CHAR STORAGE
                                        XRA     M               ;XOR DATA IN ACCUMULATOR WITH PREVIOUS CRC 7 TO 0
                                        INX     H               ;POINT TO CRC PARITY BIT STORAGE
                                        MOV     B,A             ;SAVE RESULT OF XOR
```

```
2368  480A  3A  64  48   13.0          LDA   PBIT          ;GET PARITY OF DATA BYTE ON CALL
2369  480D  AE              7.0          XRA   M             ;XOR WITH CRC PARITY BIT
2370  480E  1F              4.0          RAR                 ;PUT PARITY BIT IN 'C' BIT POSITION
2371  480F  78              4.0          MOV   A,B           ;GET XOR OF 7 TO 0
2372  4810  32  61  48   13.0          STA   CRCOLD        ;SAVE FOR PE 'CRC' CHAR TEST
2373  4813  1F              4.0          RAR                 ;SHIFT RIGHT
2374  4814  D2  22  48   10.0          JNC   CRC2          ;COMPLIMENT BITS 5,4,3,+2 IF 'C' IS SET
2375
2376  4817  EE  3C            7.0  CRC1:  XRI   @74           ;COMPLIMENT BITS
2377  4819  32  5F  48   13.0          STA   CRCCHR        ;SAVE THE BYTE
2378  481C  3E  01            7.0          MVI   A,1           ;AND SET THE 'P' BIT TO A 1 FOR NEXT TIME
2379  481E  77              7.0          MOV   M,A           ;SAVE THE 1
2380  481F  C3  27  48   10.0          JMP   CRC3          ;EXIT
2381
2382  4822  32  5F  48   13.0  CRC2:  STA   CRCCHR        ;SAVE CRC CHARACTER
2383  4825  AF              4.0          XRA   A             ;CLEAR THE 'P' BIT
2384  4826  77              7.0          MOV   M,A           ;FOR THE NEXT TIME
2385  4827  E1             10.0  CRC3:  POP   H
2386  4828  C1             10.0          POP   B
2387  4829  F1             10.0          POP   PSW
2388  482A  C9             10.0          RET
2389
2390                                ;INITIALIZE CRC CHARACTER STORAGE
2391
2392  482B  AF              4.0  CLCRC: XRA   A
2393  482C  32  5F  48   13.0          STA   CRCCHR
2394  482F  32  60  48   13.0          STA   CRCCHR+1
2395  4832  C9             10.0          RET
2396
2397                                ;TRANSFER COMPLETE... FINISH THE CRC GENERATION
2398
2399  4833  3A  61  48   13.0  CRCFIN: LDA   CRCOLD        ;GET CRC WORD
2400  4836  EE  D7            7.0          XRI   @327          ;COMPLIMENT BITS 7,6,4,2,1,+0
2401  4838  32  5F  48   13.0          STA   CRCCHR
2402  483B  C9             10.0          RET
```

```
2404                                        .SBTTL  PARITY CALCULATION ROUTINE
2405                                        ;THIS SUBROUTINE CALCULATES ODD PARITY ON THE CONTENTS OF THE
2406                                        ;ACCUMULATOR AND SETS OR CLEARS THE MEMORY LOCATION ''PBIT AS REQUIRED.
2407
2408  483C  F5              12.0  PARITY: PUSH   PSW             ;SAVE ACCUMULATOR
2409  483D  A7               4.0          ANA    A               ;SET CONDITION BITS
2410  483E  3E  00           7.0          MVI    A,$00           ;CLEAR ACCUMULATOR
2411  4840  E2  45  48       10.0         JPO    PAR1            ;SKIP IF PARITY ODD
2412  4843  3E  01           7.0          MVI    A,$01           ;ELSE-LOAD ACC. WITH 1
2413  4845  32  64  48       13.0 PAR1:   STA    PBIT            ;SET UP THE P BIT
2414  4848  F1               10.0         POP    PSW             ;RESTORE ACCUMULATOR
2415  4849  C9               10.0         RET                    ;RETURN
2416                                        ;HERE IS THE NECESSARY STORAGE FOR TEST PARAMETERS AND DDR DATA STORAGE
2417
2418  484A  00                    CCTLWD: .BYTE  0               ;IMAGE OF CLOCK CONTROL WORD
2419  484B  00                    IMBYTC: .BYTE  0               ;IMAGE OF BYTE COUNTER 7-0 BITS
2420
2421  484C  00                    ABCNT:  .BYTE  0               ;ASSEMBLED BYTE COUNTER
2422  484D  00                    EABCNT: .BYTE  0               ;BYTE COUNT AT A DETECTED ERROR
2423
2424  484E  00                    FORMT:  .BYTE  0               ;FORMAT CODE SELECTED FOR TEST
2425  484F  00                    DPNUM:  .BYTE  0               ;DATA PATTERN OFFSET NUMBER (0 TO 17)
2426  4850  00                    CLKON:  .BYTE  0               ;'SCLK' ON FLAG-USED TO TELL WHEN TO FILL DDR REGS
2427  4851  00                    ECCCNT: .BYTE  0               ;COUNT BY 7 FOR ECC CHARACTER CHECK
2428  4852  00                    CLKCNT: .BYTE  0               ;COUNT OF SCLKS GENERATED
2429
2430  4853  00                    SDDRA:  .BYTE  0               ;DDRA DATA ACTUALLY LOADED
2431  4854  00                    SDDRB:  .BYTE  0               ;DDRB DATA ACTUALLY LOADED
2432  4855  00                    SDDRC:  .BYTE  0               ;DDRC DATA ACTUALLY LOADED
2433
2434  4856  00                    SDDRAT: .BYTE  0               ;DDRA DATA FROM PGM TABLE
2435  4857  00                    SDDRBT: .BYTE  0               ;DDRB DATA FROM PGM TABLE
2436  4858  00                    SDDRCT: .BYTE  0               ;DDRC DATA FROM PGM TABLE
2437
2438  4859  00                    SWMCSTA: .BYTE 0               ;COPY OF THE WMC STATUS AT 'XFER' BIT TIME
2439  485A  00                    SXFER:  .BYTE  0               ;SOFTWARE 'XFER' FLAG
2440  485B  00                    SINTSTA: .BYTE 0               ;SOFTWARE COPY OF INTSTA WORD
2441  485C  00                    PADEXP: .BYTE  0               ;
2442
2443  485D  00                    ECCEXP: .BYTE  0               ;
2444  485E  00                    ECCCHR: .BYTE  0               ;ECC CHARACTER CALCULATED BY THE SOFTWARE
2445  485F  00  00                CRCCHR: .WORD  0               ;IMAGE OF THE CRC CHARACTER
2446  4861  00                    CRCOLD: .BYTE  0               ;CRC CHAR TO USE IN PE TEST (INTERMED. STEP)
2447  4862  00                    HAVCRC: .BYTE  0               ;RECEIVED 'CRC' CHAR
2448  4863  00                    RPADCT: .BYTE  0               ;# RECEIVED PAD CHAR
2449  4864  00                    PBIT:   .BYTE  0
2450  4865  00                    CLKCRC: .BYTE  0               ;EXTRA CLOCK-TO-CRC FLAG
2451  4866  00                    TPBCNT: .BYTE  0               ;TEST PASS BYTE COUNTER
2452  4867  00                    PDCRC:  .BYTE  0               ;PAD/CRC FLAG IF SET, WMC WILL ISSUE A PAD
2453  4868  00                    LASTCRC: .BYTE 0               ;LAST CRC BYTE EXPECTED FLAG
2454  4869  00                    MOD7R:  .BYTE  0               ;IMAGE OF RESID GROUP CHAR EXPECTED
2455  486A  00                    RCHRIM: .BYTE  0               ;IMAGE OF WHAT LOADED INTO RESCHR WORD
2456  486B  00                    XFRREC: .BYTE  0               ;COUNT OF RECEIVED 'XFER' CHARACTERS
```

```
                                        .SBTTL  SUBROUTINE GENDAT
                                        $
                                        ; ************************************************************************
                                        ;GENDAT -- ROUTINE TO GENERATE DDR REG DATA FOR NEXT "SCLK" CYCLE.
                                        ;       DATA IS LOADED IN DDR REGS AND A COPY IS STORED IN 'SDDRAC', +
                                        ;       'SDDRCC' FOR REFERENCE.
                                        ;
                                        $
                                        ; ************************************************************************

                                12.0  GENDAT: PUSH    PSW             ;SAVE PSW + A
                                12.0          PUSH    B               ;SAVE B + C
                                12.0          PUSH    H               ;SAVE H + L
                                13.0          LDA     DPNUM           ;GET THE LAST PATTERN # USED
                                 4.0          INR     A               ;POINT TO NEXT
                                13.0          STA     DPNUM           ;SAVE NUMBER
                                 4.0          MOV     C,A
                                 7.0          MVI     B,0             ;B + C HAVE OFFSET
                                10.0          LXI     H,CDATBL        ;POINT TO DATA TABLE
                                10.0          DAD     B               ;ADD IN OFFSET TO H + L

                                 7.0          MOV     A,M             ;GET DATA BYTE FOR DDR 'A'
                                13.0          STA     SDDRAT          ;SAVE DDR 'A' IMAGE
                                 4.0          XRA     A               ;
                                13.0          STA     SDDRBT          ;SAVE DDR 'B' IMAGE
                                13.0          STA     SDDRCT          ;SAVE DDR 'C' IMAGE

                                10.0  GENOUT: POP     H               ;RESTORE REGS
                                10.0          POP     B
                                10.0          POP     PSW
                                10.0          RET
```

```
                                        .SBTTL  SUBROUTINE SETDDR
                                        $
                                        ; ************************************************************************
                                        ;SETDDR -- ROUTINE TO TRANSLATE THE DESIRED DATA FROM A 1-FOR-1 BIT
                                        ;       POSITION TO THE DDR TO BE USED AS DATA TO EITHER THE TM78 OR
                                        ;       THE MASSBUS. WMC "LEFT" FLAG SET INDICATES THAT THIS ROUTINE IS NOT
                                        ;       NECESSARY, THE DATA IS IN THE CORRECT FORMAT ALREADY. IF WMC
                                        ;       "LEFT" FLAG IS NOT SET, THE DATA MUST BE TRANSLATED TO A COMPATABLE
                                        ;       DDR FORMAT.
                                        $
                                        ; ************************************************************************

                                        ;HERE TO TRANSLATE THE DATA INTO AN ACCEPTABLE FORMAT FOR DDR INJECTION

488C  3A 56 48        13.0  SETDDP: LDA     SDDRAT          ;GET DDRA TEMP
488F  E6 3F            7.0          ANI     @77             ;STRIP 2 BITS OFF
4891  07                4.0          RLC
4892  07                4.0          RLC                     ;LEFT JUSTIFY
4893  47                4.0          MOV     B,A             ;TEMP SAVE
4894  3A 58 48        13.0          LDA     SDDRCT          ;GET DDRC TEMP
4897  E6 03            7.0          ANI     @3              ;SAVE 2 BITS
4899  B0                4.0          ORA     B
489A  32 53 48        13.0          STA     SDDRA           ;SAVE DDRA DESIRED
489D  D3 D8           10.0          OUT     DDRA            ;LOAD THE REAL DDRA

489F  3A 56 48        13.0          LDA     SDDRAT          ;GET DDRA AGAIN
48A2  E6 C0            7.0          ANI     @300            ;SAVE 2 BITS
48A4  07                4.0          RLC
48A5  07                4.0          RLC
48A6  32 54 48        13.0          STA     SDDRB           ;SAVE DDRB DATA
48A9  D3 D9           10.0          OUT     DDRB            ;LOAD THE REAL DDRB

48AB  3A 55 48        13.0          LDA     SDDRC
48AE  D3 DA           10.0          OUT     DDRC
48B0  C9              10.0          RET                     ;EXIT

48B1  F7                    CDATBL: .BYTE   @367            ;1ST SET OF DATA HAS KNOWN RESULTS
48B2  FB                            .BYTE   @373
48B3  00                            .BYTE   @0
48B4  23                            .BYTE   @43
48B5  88                            .BYTE   @210
48B6  C0                            .BYTE   @300
48B7  63                            .BYTE   @143            ;EXPECTED 'ECC' IS @311
                                                            ;EXPECTED INTERMEDIATE 'CRC' IS @142

48B8  12                            .BYTE   $12             ;HEX 12
48B9  23                            .BYTE   $23             ;HEX 23
48BA  34                            .BYTE   $34             ;HEX 34
48BB  45                            .BYTE   $45             ;HEX 45
48BC  56                            .BYTE   $56             ;HEX 56
48BD  67                            .BYTE   $67             ;HEX 67
48BE  78                            .BYTE   $78             ;HEX 78
48BF  89                            .BYTE   $89             ;HEX 89
48C0  9A                            .BYTE   $9A             ;HEX 9A
```

```
2539  48C1  AB                              .BYTE   $AB             ;HEX AB
2540  48C2  BC                              .BYTE   $BC             ;HEX BC
2541  48C3  CD                              .BYTE   $CD             ;HEX CD
2542  48C4  DE                              .BYTE   $DE             ;HEX DE
2543  48C5  EF                              .BYTE   $EF             ;HEX EF
2544  48C6  F0                              .BYTE   $F0             ;HEX F0
2545  48C7  01                              .BYTE   $01             ;HEX 01
2546  48C8  02                              .BYTE   $02             ;FLOATING 1'S
2547  48C9  04                              .BYTE   $04
2548  48CA  08                              .BYTE   $08
2549  48CB  10                              .BYTE   $10
2550  48CC  20                              .BYTE   $20
2551  48CD  40                              .BYTE   $40
2552  48CE  80                              .BYTE   $80
2553  48CF  FE                              .BYTE   $FE             ;FLOATING 0'S
2554  48D0  FD                              .BYTE   $FD
2555  48D1  FB                              .BYTE   $FB
2556  48D2  F7                              .BYTE   $F7
2557  48D3  EF                              .BYTE   $EF
2558  48D4  DF                              .BYTE   $DF
2559  48D5  BF                              .BYTE   $BF
2560  48D6  7F                              .BYTE   $7F
2561  48D7                          DTEND:                          ;END OF THE DATA TABLE
2562
2563                                ;ROUTINE TO SHIFT CONTENTS OF ACCUMULATOR RIGHT OR LEFT EITHER 2 OR 4 TIMES
2564
2565  48D7  E6  FC          7.0   SHR4:   ANI     @374              ;STRIP OFF 2 BITS
2566  48D9  0F              4.0           RRC                       ;SHIFT RIGHT 2 TIMES
2567  48DA  0F              4.0           RRC
2568  48DB  E6  FC          7.0   SHR2:   ANI     @374              ;STRIP OFF 2 BITS
2569  48DD  0F              4.0           RRC                       ;SHIFT RIGHT 2 TIMES
2570  48DE  0F              4.0           RRC
2571  48DF  C9             10.0           RET
2572
2573  48E0  E6  3F          7.0   SHL4:   ANI     @77               ;SHIFT LEFT TWICE
2574  48E2  07              4.0           RLC
2575  48E3  07              4.0           RLC
2576  48E4  E6  3F          7.0   SHL2:   ANI     @77
2577  48E6  07              4.0           RLC                       ;SHIFT LEFT TWICE
2578  48E7  07              4.0           RLC
2579  48E8  C9             10.0           RET
2580  48E9                                S
 (1)                                    ; ***************************************************************************
2581                                ;CLKSYS -- ROUTINE TO CLOCK THE SYSTEM 1 TIME BY WRITING CLOCK CONTROL
2582                                ;       'CLK' BIT TO A '1' THEN A '0'.
2583  48E9                          ;
 (1)                                    ; ***************************************************************************
2584
2585  48E9  3A  4A  48      13.0   CLKSYS: LDA     CCTLWD            ;GET SOFTWARE CLOCK CONTROL IMAGE
2586  48EC  F6  40           7.0           ORI     SSCLK             ;ADD IN 'CLK' BIT
2587  48EE  D3  F0          10.0           OUT     CLKCTL            ;LOAD CLOCK CONTROL
2588  48F0  E6  3F           7.0           ANI     @77               ;STRIP OFF CLOCK BIT
2589  48F2  D3  F0          10.0           OUT     CLKCTL            ;LOAD CLOCK CONTROL WORD
2590  48F4  C9              10.0           RET                       ;EXIT - CLOCK CYCLE COMPLETE
```

```
                                        .SBTTL  CLEAR ECC SUBROUTINE
                                        $
                                        ; ************************************************************************
                                        ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION "ECCCHR" -
                                        ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
                                        ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
                                        $
                                        ; ************************************************************************

                                C'ECC:  PUSH    PSW             ;SAVE THE ACCUMULATOR
                                        XRA     A               ;CLEAR THE ACCUMULATOR
                                        STA     ECCCHR          ;CLEAR THE ECC CHARACTER
                                        STA     ECCCNT          ;CLEAR THE COUNT OF ASSEMBLED BYTES
                                        POP     PSW             ;RESTORE THE ACCUMULATOR
                                        RET                     ;RETURN TO USER
```

```
                                        .SBTTL  CALCUALTE ECC CHARACTER
 48FF                                   $
  (1)                                   ; ************************************************************************
                                        ;GENECC --THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
                                        ;THE CONTENTS OF "ECCCHR" TO UPDATE THE CONTENTS OF "ECCCHR" ACCORDING
                                        ;TO THE ANSI STANDARD ECC POLYNOMIAL.
 48FF                                   $
  (1)                                   ; ************************************************************************

 48FF F5              12.0  GENECC: PUSH    PSW             ;SAVE THE ACCUMULATOR
 4900 C5              12.0          PUSH    B               ;SAVE B&C
 4901 D5              12.0          PUSH    D               ;SAVE D&E
 4902 E5              12.0          PUSH    H               ;SAVE H&L
 4903 21 5E 48        10.0          LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
 4906 AE               7.0          XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
 4907 5F               4.0          MOV     E,A             ;SAVE XOR RESULT IN E
 4908 E6 10            7.0          ANI     $10             ;IS BIT #4 OF RESULT SET
 490A 7B               4.0          MOV     A,E             ;RESTORE XOR RESULT FROM B
 490B CA 10 49        10.0          JZ      ECC1            ;CONTINUE IF BIT #4 RESET
 490E EE 23            7.0          XRI     $23             ;ELSE-XOR WITH 23
 4910 5F               4.0  ECC1:   MOV     E,A             ;STORE THE ECC RESULT IN E
 4911 AF               4.0          XRA     A               ;CLEAR A
 4912 4F               4.0          MOV     C,A             ;CLEAR THE TRANSLATE RESULT
 4913 21 22 49        10.0          LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
 4916 CD 2A 49        18.0          CALL    TRANS           ;TRANSLATE THE BITS
 4919 79               4.0          MOV     A,C             ;GET THE TRANSLATED RESULT
 491A 32 5E 48        13.0          STA     ECCCHR          ;STORE RESULT
 491D E1              10.0          POP     H               ;RESTORE H&L
 491E D1              10.0          POP     D               ;RESTORE D&E
 491F C1              10.0          POP     B               ;RESTORE B&C
 4920 F1              10.0          POP     PSW             ;RESTORE ACCUM.
 4921 C9              10.0          RET

 4922 08               ECCTBL: $08                    ;BIT 0 = POSITION 3
 4923 20                       $20                    ;BIT 1 = POSITION 5
 4924 02                       $02                    ;BIT 2 = POSITION 1
 4925 40                       $40                    ;BIT 3 = POSITION 6
 4926 80                       $80                    ;BIT 4 = POSITION 7
 4927 01                       $01                    ;BIT 5 = POSITION 0
 4928 10                       $10                    ;BIT 6 = POSITION 4
 4929 04                       $04                    ;BIT 7 = POSITION 2
```

```
                                        .SBTTL  POLYNOMIAL BIT TRANSLATION

                                        ;
                                        ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
                                        ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
                                        ;
492A  06  01            7.0  TRANS:  MVI   B,1          ;INIT 'B' TO BIT POSITION 0
492C  7B                 4.0  TRANS1: MOV   A,E          ;GET CHAR TO BE TRANSLATED
492D  A0                 4.0          ANA   B            ;SEE IF BIT POSITION IN 'B' IS SET
492E  CA  34  49        10.0          JZ    TRANS2       ;DO NEXT BIT POSITION IF NOT SET
4931  79                 4.0          MOV   A,C          ;GET PREVIOUS RESULT OF 'OR'
4932  B6                 7.0          ORA   M            ;'OR' IN NEW POSITION
4933  4F                 4.0          MOV   C,A          ;SAVE RESULT

4934  78                 4.0  TRANS2: MOV   A,B
4935  07                 4.0          RLC                ;POSITION MASK TO NEXT BIT
4936  47                 4.0          MOV   B,A
4937  D8                12.0          RC                 ;EXIT WHEN ALL POSITIONS DONE
4938  23                 6.0          INX   H            ;POINT TO NEXT TABLE ENTRY
4939  C3  2C  49        10.0          JMP   TRANS1       ;PROCESS NEXT BIT
      0000                    .END
```

```
A      =%0007          ABCNT   484C          ADATA = 0094          AMTIEP= 0001
AMTIE7= 0002           ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000         BADST = 0090          BIT0  = 0001
BIT1  = 0002           BIT15 = 8000          BIT2  = 0004          BIT3  = 0008
BIT4  = 0010           BIT5  = 0020          BIT6  = 0040          BIT7  = 0080
BIT8  = 0100           BIT9  = 0200          BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60           BRKXCT= 4F00          BSAVE   4F9C          BYTCNT= 00D4
BYTEH   4F24           BYTEL   4F23          C      =%0001         CASCT   4F21
CASCTL= 00A0           CASSTA= 00A0          CATTH = 0089          CATTL = 0088
CBUSST= 00A1           CBYTH = 008B          CBYTL = 008A          CCTLWD  484A
CDATBL  48B1           CDG1H = 0087          CDG1L = 0086          CDG2H = 0093
CDG2L = 0092           CDG3H = 0095          CDG3L = 0094          CDVTH = 008D
CDVTL = 008C           CHPTIE= 0028          CH0TIE= 0020          CH1TIE= 0021
CH2TIE= 0022           CH3TIE= 0023          CH4TIE= 0024          CH5TIE= 0025
CH6TIE= 0026           CH7TIE= 0027          CKCRC   47CC          CKECC   4757
CKECCC  4767  G        CKECC1  47C6          CKHACK  460E          CKLOP = 2815
CKPAD   4558           CKPADA  455C          CKPAD1  45C0          CKSCLK  4474
CKSC1   4483           CLCRC   482B          CLECC   48F5          CLKCNT  4852
CLKCRC  4865           CLKCTL= 00F0          CLKON   4850          CLKSYS  48E9
CLOCK   4F26           CMC0H = 0099          CMC0L = 0098          CMC1H = 009B
CMC1L = 009A           CMC2H = 009D          CMC2L = 009C          CMC3H = 009F
CMC3L = 009E           CMINH = 0097          CMINL = 0096          CNTCTL= 00D7
CRCCHR  485F           CRCFIN  4833          CRCOLD  4861          CRCWRD= 0018
CRC1    4817           CRC2    4822          CRC2A   47CE          CRC3    4827
CRC4    47E0           CRC5    47F5  G       CSAVE   4F9D          CSRLH = 0091
CSRLL = 0090           CTCH  = 0085          CTCL  = 0084          CTSTH = 008F
CTSTL = 008E           CXCTH = 0081          CXCTL = 0080          CXINH = 0083
CXINL = 0082           C.    = 0001          C.AVAI= 0080          C.DP  = 0008
C.DSE = 0010           C.DTU = 0003          C.DVA = 0008          C.FAIL= 00FC
C.FMT = 0070           C.FNCT= 003E          C.GO  = 0001          C.INTC= 00FE
C.MAIN= 0020           C.NSA = 0080          C.RCT = 00FC          C.SER = 0080
C.SHR = 0040           C.SKPC= 000F          C.TAPE= 0040          C.WCS = 0002
D      =%0002          DATACT= 00D0          DATPE   469C          DATPEC  46A7  G
DATPEE  46AD           DATPO   4688          DATPOC  4695  G       DATPOE  469B
DBUS    4F28           DBUSCT= 00C0          DBUSST= 00C0          DDRA  = 00D8
DDRAIN= 0010           DDRB  = 00D9          DDRBIN= 0002          DDRC  = 00DA
DDRCIN= 0001           DDRCO = 0088          DDRCTL= 00DB          DELCOP  473C
DELP1   4749           DIAFLG  4F22          DIAGPG= 4300          DIAGRM= 4F90
DIARD = 000B           DONE1 = 0045          DONINT= 0010          DPNUM   484F
DSAVE   4F9E           DSE   = 0006          DTEND   48D7          D.ATH0= 0001
D.ATH1= 0002           D.EOTD= 0010          D.LAGC= 0020          D.NOTW= 0040
D.NTHR= 0004           D.TACH= 0008          D.WR4 = 0080          E      =%0003
EABCNT  484D           ECCBAD= 0042          ECCCHR  485E          ECCCK   476D
ECCCK1  4782  G        ECCCK2  4788          ECCCNT  4851          ECCCOR= 0019
ECCEXP  485D           ECCOK = 0041          ECCPE   47B4          ECCPEC  47BF  G
ECCPEE  47C5           ECCPO   47A0          ECCPOC  47AD  G       ECCPOE  47B3
ECCSTA= 001A           ECCTBL  4922          ECCTST= 000E          ECC1    4910
EDATA = 0095           EOTCLR= 0003          ERFLG   4F93          ERLP  = 2809
ERLPA = 280F           ERLPB = 2812          ERLPE = 280C          ERNUM   4F90
ERRCNT= 00D6           ESAVE   4F9F          E.ACRC= 0010          E.AMT = 0020
E.CDP = 0080           E.CRC = 0080          E.PNTR= 0008          E.RPE = 0040
E.STEC= 0001           E.TTEC= 0002          E.UNC = 0004          FIFORD= 006A
FORMAT  4F25           FORMT   484E          FWDTST= 0061          GCRID = 0089
GCRSET= 0002           GENCRC  47FE          GENDAT  486C          GENECC  48FF
```

```
GENOUT  4888          GOODTM= 0092          4     =%0004          HAVCRC  4862
HLSAVE  4FA0          IE    = 0008          IMBYTC  484B          INTSTA= 00E0
ITERA   4F9A          I.PWR = 0020          I.RMPE= 0040          I5.5  = 0010
I6.5  = 0020          I7.5  = 0040          KCALL = 005F          KCLR  = 007B
KDEP  = 003F          KENAB = 0078          KEXAM = 003E          KEYBRD= 00C8
KEY1  = 0078          KEY10 = 006D          KEY11 = 006E          KEY12 = 006F
KEY13 = 005C          KEY14 = 005D          KEY15 = 005E          KEY16 = 005F
KEY17 = 003C          KEY18 = 003D          KEY19 = 003E          KEY2  = 0079
KEY20 = 003F          KEY3  = 007A          KEY4  = 007B          KEY5  = 0074
KEY6  = 0075          KEY7  = 0076          KEY8  = 0077          KEY9  = 006C
KINTA = 006F          KLDAD = 003D          KN0   = 003C          KN1   = 005C
KN2   = 005D          KN3   = 005E          KN4   = 006C          KN5   = 006D
KN6   = 006E          KN7   = 0074          KN8   = 0075          KN9   = 0076
KU2   = 0079          KU3   = 007A          KU8   = 0077          L     =%0005
LASTCR  4868          LBLANK= 000F          LCE   = 000B          LCH   = 000C
LCL   = 000D          LCP   = 000E          LC0   = 0000          LC1   = 0001
LC2   = 0002          LC3   = 0003          LC4   = 0004          LC5   = 0005
LC6   = 0006          LC7   = 0007          LC8   = 0008          LC9   = 0009
LDLEDA= 00CA          LDLEDB= 00CB          LDLEDC= 00CC          LDLEDD= 00CD
LDLEDE= 00CE          LDLEDF= 00CF          LKDIAG= 2800          LKKBD = 004C
LKKEY = 0049          LKLWMG= 0058          LKLWMP= 0055          LKLWPG= 0052
LKLWPP= 004F          LKMOD7= 0046          LKOPR = 0046          LPFLG   4F94
LPNUM   4F92          M     =%0006          MBSEL = 00E0          MB.A  = 0008
MB.B  = 0004          MEMTOP= 4FFF          MINUS = 000A          MM    = 8000
MOD7R   4869          MSE   = 0008          MSGN  = 0018          MTACLR= 0000
MT.ARA= 0020          MT.CPE= 0080          MT.DSE= 0001          MT.FWD= 0040
MT.INH= 0008          MT.LWR= 0004          MT.MOT= 0002          MT.NWT= 0080
MT.PEC= 0040          MT.PSB= 0004          MT.PS0= 0001          MT.PS1= 0002
MT.REV= 0020          MT.WRT= 0010          MT.Z  = 0008          M.ATA = 0080
M.CAPE= 0020          M.CONT= 0080          M.DEM = 0020          M.EBL = 0010
M.EXC = 0008          M.FAIL= 0008          M.ILR = 0010          M.INIT= 0010
M.OCC = 0020          M.ONLI= 0001          M.PE  = 0040          M.PORT= 0080
M.RDEN= 0002          M.RDPE= 0008          M.RUN = 0004          M.SCLK= 0001
M.TRA = 0040          M.UNIT= 0007          M.WCLK= 0040          M.WCLN= 0080
M.WREN= 0080          M5.5  = 0001          M6.5  = 0002          M7.5  = 0004
NOTCAP= 0088          OKAY  = 00FF          OPRRAM= 4300          OPSTRT= 0058
OPVER = 0040          PADCNT= 00D5          PADCN0  457C  G       PADCRC= 0080
PADEXP  485C          PADPE   45AE          PADPEC  45B9  G       PADPEE  45BF
PADPO   459A          PADPOC  45A7  G       PADPOE  45AD          PARITY  483C
PAR1    4845          PBIT    4864          PDCRC   4867          PDIAG = 0048
PEID  = 008A          PENAB = 004C          PESET = 0001          PEXPTB  43BF
PL    = 00B1          PRDD  = 004C          PRENF = 009C          PS    = 00B2
PSTAT = 0048          PSW   =%0009          P.AMTP= 0001          P.BCTC= 0040
P.CMDP= 0020          P.INTE= 0080          P.LCS = 0040          P.LWR = 0020
P.RDP = 0002          P.RPEN= 0004          P.RPST= 0002          P.RPOE= 0020
P.RP1E= 0010          P.RP2E= 0020          P.RP3E= 0010          P.SING= 0080
P.STAT= 0002          P.STPE= 0080          P.TACH= 0008          P.TUPR= 0010
P.WCSP= 0004          P.WDS = 0040          P.WFLP= 0001          P.WPEN= 0010
P.WPOE= 0008          P.WP1E= 0004          P.WP2E= 0008          P.WP3E= 0004
P.XVOK= 0002          QUE   = 281B          QUEM  = 281E          QUIT    46ED
QUIT1   4702  G       QUIT3   4717  G       QUIT4   4708          RAMT  = 0010
RARA  = 0006          RARAI = 0004          RCHBD0= 0048          RCHBD1= 0047
RCHOK = 0046          RCHRIM  486A          RCHTST= 000C          RCLRT = 000D
RCMD  = 000B          RCMLP = 0003          RCONT = 0080          RDATA = 0017
```

```
RDCLK = 0010    RDON  = 0011    READG = 0007    REND  = 0014
REQST = 2806    RESCHR= 00D1    REVTST= 0064    REWIND= 0004
RFIFOL= 0008    RGCLK = 0002    RGCRI = 0003    RIBG  = 0001
RILL  = 0012    RINST = 000C    RMCTST= 0008    RMK2  = 0013
RNOP  = 0000    RPADCT  4863    RPATH = 0001    RPBAD = 0044
RPCHI = 0001    RPCLK = 0003    RPCTL = 0009    RPEI  = 0002
RPFAIL= 0000    RPF1  = 009D    RPF2  = 009E    RPOK  = 0043
RPOSTN= 0016    RPSTA = 0015    RRCMT = 000A    RSTAT = 0002
RTIEB = 000A    RTIER = 0030    RTM   = 0005    RUNKI = 0009
RUPTST= 005E    RWDUNL= 0005    R.AMT = 0001    R.BOP = 0008
R.DATA= 0040    R.DON = 0002    R.DRDY= 0010    R.END = 0010
R.ILL = 0004    R.JVOK= 0004    R.MK2 = 0008    R.PLOD= 0008
R.PLOO= 0010    R.PL01= 0020    R.POST= 0020    R.STNM= 0002
R.STOP= 0004    R.STPC= 0001    R.TBJN= 0080    R.TSTD= 0040
R.VOK = 0080    R00H  = 0081    R00L  = 0080    R01H  = 0083
R01L  = 0082    R02H  = 0085    R02L  = 0084    R03H  = 0087
R03L  = 0086    R04H  = 0089    R04L  = 0088    R05H  = 008B
R05L  = 008A    R06H  = 008D    R06L  = 008C    R07H  = 008F
R07L  = 008E    R10H  = 0091    R10L  = 0090    R11H  = 0093
R11L  = 0092    R12H  = 0095    R12L  = 0094    R13H  = 0097
R13L  = 0096    R14H  = 0099    R14L  = 0098    R15H  = 009B
R15L  = 009A    R16H  = 009D    R16L  = 009C    R17H  = 009F
R17L  = 009E    R7.5  = 0010    SCHREX  462F    SCKB1   4466
SCKGON  445F    SCKWT   4431    SCKWTC  4454  G SCKWTE  445D
SCKWT1  4433    SCKWT2  4440    SCLKBI  4426    SDDRA   4853
SDDRAT  4856    SDDRB   4854    SDDRBT  4857    SDDRC   4855
SDDRCT  4858    SELCLR= 0000    SETATA= 00A1    SETDDR  488C
SHL2    48E4    SHL4    48E0    SHR2    48DB    SHR4    48D7
SID   = 0080    SINTST  485B    SOD   = 0080    SOE   = 0040
SP    =%0008    SSCLK = 0040    SSTEP = 0005    STACK = 4FFF
STATRM= 4F20    STPCT   4F20    STRSP = 5000    SWMCST  4859
SXFER   485A    TADR00= 0080    TADR01= 0081    TADR02= 0082
TADR03= 0083    TADR04= 0084    TADR05= 0085    TADR06= 0086
TADR07= 0087    TADR10= 0088    TADR11= 0089    TADR12= 008A
TADR13= 008B    TAMT  = 0044    TASEL = 0080    TCMD  = 0040
TC.FWD= 0040    TC.INH= 0008    TC.LWR= 0004    TC.REV= 0020
TC.WRT= 0010    TEMP    4F99    TMF   = 0099    TMRDY = 0040
TPBCNT  4866    TRANS   492A    TRANS1  492C    TRANS2  4934
TRKENA= 00D2    TSET  = 2803    TSTEND= 2818    TSTS  = 0040
TST1    4300    TST1A   43D3    TST1B   43DE    TST1BC  440B  G
TST1C   43FB    TST1D   4414    TST1DC  441D  G TST1L   431F
TST1NB  430E    TST1ST  4305    TST1XC  466A  G TUSEL0= 00D1
TUSEL1= 00D2    TU78  = 0010    T.ATH0= 0001    T.ATH1= 0002
T.BOT = 0004    T.EOT = 0002    T.FPT = 0001    T.NTHR= 0004
T.ONL = 0020    T.PES = 0008    T.PSBJ= 0020    T.PSOJ= 0008
T.PS1J= 0010    T.RDY = 0080    T.RDYO= 0040    T.RWD = 0010
T.SCLK= 0002    T1RST   4373    T1RSTA  439B    T1RSTB  439E
T1RSTC  43C7    T1RSTD  43AE    UIBG  = 00A1    VALFC   4F98
VALTB   4F95    VELTST= 005B    WDR.P = 0010    WMCCTL= 00D3
WMCERR= 00DA    WMCSTA= 00D0    WRTCLK= 0000    WRTDAT= 00D3
W.ACRC= 0004    W.CRC = 0008    W.DIAG= 0002    W.DONN= 0040
W.ECC = 0010    W.ENAB= 0080    W.ERR = 0020    W.FMT = 0070
W.GCR = 0010    W.LEFT= 0004    W.ONES= 0020    W.RESI= 0002
W.REV = 0004    W.ROME= 0010    W.RST = 0001    W.SKIP= 000F
```

```
W.WRIT= 0008            W.XFER= 0020            X     =%000A            XFB4C   4629  G
XFRB1T  448A            XFRB1   4491            XFRB2   44AF            XFRB2C  44C5  G
XFRB3   44CE            XFRB3A  44DC            XFRB3C  44F2  G         XFRB4   44FB
XFRB4C  4641            XFRB4D  4545            XFRB4P  4670            XFRB5   46AE
XFRB5C  46E4  G         XFRDY1  451E  G         XFRDY2  4527            XFRDY3  453C  G
XFRERR  4624            XFRGON  44A7  G         XFRRDY  4501            XFRREC  486B
XMCCK   45D0            XMCK1   45D4            XMCK1A  45D2            XMCK2   45F1
XMCK3   45F5            XMCK3C  4605  G         X.DONN= 0080            X.ENAB= 0040
X.PEPE= 0002            X.ROME= 0001            X.WCLK= 0001            Y     =%000B
.     = 493C


 ERRORS DETECTED:  0

 *WMC5.A78/PTP,WMC5=NLIST,PARAM,MACRO,LIST,WMC5
 RUN-TIME: 5 7 0 SECONDS
 CORE USED:  10K
```

3 "MEMORY MAP" EXTENDED ADDRESS BIT
19 READ MICRO CONTROLLER CONTROL & STATUS WORDS
139 TRANSPORT/PORT COMMAND + STATUS WORDS
380 COMMON ADDRESS SPACE REGISTERS
444 MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485 MASSBUS CONTROL WORD DEFINITIONS
557 WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709 KEYBOARD/DISPLAY CONTROL + STATUS WORD
797 MISCELLANEOUS VARIABLE DEFINITIONS
801 RAM ADDRESS DEFINITIONS
819 LINKAGE TABLE ADDRESSES
842 BIT DEFINITIONS
856 RIM AND SIM INSTRUCTION BIT DEFINITIONS
877 INTERRUPTABLE INSTRUCTION MACROS
924 DIAGNOSTIC VARIABLES
945 COMMON VARIABLE STORAGE
959 MACRO CALL DEFINITIONS
1047 TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101 8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1333 TEST 1 - LOOP CMD. TO STATUS - TU PORT #0
1407 TEST 2 - LOOP CMD. TO STATUS - TU PORT #1
1460 TEST 3 - LOOP CMD. TO STATUS - TU PORT #2
1513 TEST 4 - LOOP CMD. TO STATUS - TU PORT #3
1566 TEST 5 - WCS PARITY INT. TEST - TU PORT #0
1623 TEST 6 - WCS PARITY INT. - TU PORT #1
1680 TEST 7 - WCS PARITY INT. - TU PORT #2
1736 TEST 10 - WCS PARITY INT. - TU PORT #3
1793 TEST 11 - LOOP AMTIE TO AMTIE - TU PORT #0
1838 TEST 12 - LOOP AMTIE TO AMTIE - TU PORT #1
1882 TEST 13 - LOOP AMTIE TO AMTIE - TU PORT #2
1926 TEST 14 - LOOP AMTIE TO AMTIE - TU PORT #3
1970 TEST 15 - LOOP "W/R ENA" TO "W/R ENA" - TU PORT #0
2054 TEST 16 - LOOP "W/R ENA" TO "W/R ENA" - TU PORT #1
2138 TEST 17 - LOOP "W/R ENA" TO "W/R ENA" - TU PORT #2
2242 TEST 20 - LOOP "W/R ENA" TO "W/R ENA" - TU PORT #3
2344 TEST 21 - LOOP "TACH" TO "TACH" - TU PORT #0
2385 TEST 22 - LOOP "TACH" TO "TACH" - TU PORT #1
2426 TEST 23 - LOOP "TACH" TO "TACH" - TU PORT #2
2467 TEST 24 - LOOP "TACH" TO "TACH" - TU PORT #3
2511 SUBROUTINE CLEAR ALL TU PORTS
2555 SUBROUTINE CONTR1 - LOOP COMMAND TO STATUS TESTS 1 - 4
2627 SUBROUTINE CONTR2 - LOOP AMTIE TO AMTIE TESTS 11 - 14
2704 SUBROUTINE CONTR3 - WMC PARITY GENERATOR
2802 SUBROUTINE CONTR4 - LOOP TACH TO TACH
2850 PROGRAM VARIABLES

```
                ;*************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;               "BYTE/SCLK COUNT NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               MM = DATA FORMAT FROM CAS REGISTER 2
                ;               NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;               BYTE-SCLK COUNT = LLL
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               LLL = AS ABOVE
                ;               MM  = AS ABOVE
                ;               NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;               TRANSITION COUNT = LLL
                ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;               EXPECTED 18 BITS =      E EEFEEE
                ;               ACTUAL 18 BITS =        A AAAAAA
                ;
                ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;               OF CAS REG 15 LOW BYTE.
                ;
                ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;               17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;               AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;               "SUBGROUP NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;*************************************************************************
```

```
1016                    ;*****************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ;*****************************************************************
1035                    ;*****************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ;*****************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  TUP1 - TAPE UNIT PORT TEST PART #1
                                .SBTTL  TEST  1 - LOOP CMD. TO STATUS - TU PORT #0
                                ;ID     TUP1-TAPE UNIT PORT CONTROLLER PART #1
 4300                           ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP COMMAND TO STATUS-TU PORT #0
 4300                           SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE DATA PATH BETWEEN THE TU PORT #0 COMMAND AND
                                ;*STATUS REGISTER AND VERIFIES THE OPERATIONALITY OF THE COMMAND AND
                                ;*STATUS BITS.  THE ABSENCE OF BIT COUPLING IS ALSO TESTED.
 4300                           SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #0 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    LOAD TU PORT #0 SELECT CODE
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    CALL SUBROUTINE CONTR1
                                ;*ENDTST
 4300                           SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP1 MICRO TEST 01
                                ;*TUP1 MICRO ERROR 01
                                ;*TUP1-LOOP CMD. TO STATUS-TU PORT #0
                                ;*M8955'S, M8954, M8960
                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*TUP1 MICRO TEST 01
                                ;*TUP1 MICRO ERROR 14
                                ;*TUP1-LOOP CMD. TO STATUS-TU PORT #0
                                ;*M8955'S, M8954, M8960
                                ;*DATA WRITTEN TO CMD. NOT = DATA READ AT STATUS
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 01
                                ;*TUP1 MICRO ERROR 15
                                ;*TUP1-LOOP CMD. TO STATUS-TU PORT #0
                                ;*M8955'S, M8954, M8960
                                ;*PTY. BIT NOT = 1 FOR DATA
                                ;*ACTUAL = NNNN
                                ;*
```

```
1374                                    ;*TUP1 MICRO TEST 01
1375                                    ;*TUP1 MICRO ERROR 16
1376                                    ;*TUP1-LOOP CMD. TO STATUS-TU PORT #0
1377                                    ;*M8955'S, M8954, M8960
1378                                    ;*PTY. BIT NOT = 0 FOR DATA
1379                                    ;*ACTUAL = NNNN
1380  4300                              $
 (1)                                    ; ************************************************************************
1381
1382  4300                              TEST1:  TESTX   $1              ;SET UP THE TEST NUMBER
 (1)  4300  3E  01           7.0                        MVI     A,$1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28      18.0                        CALL    TSET            ;SETUP THE TEST
1383                                    ;%TUP1-LOOP CMD. TO STATUS-TU PORT #0
1384                                    ;&M8955'S, M8954, M8960
1385  4305                                      REQ     @7,0,0,0,0
 (1)  4305  CD  06  28      18.0                        CALL    REQST
 (1)  4308  00                                          .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4309  00  00                                      .WORD   0               ;SYSTEM "0" COUNT
 (1)  430B  00  00                                      .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                          .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1)  430E  07                                          .BYTE   @7              ;REQUEST CODE
1386  430F                                      RIN     R12L            ;GET THE UNITS DESIRED
 (1)  430F  DB  94          10.0                        IN      R12L            ;READ R12L INTO AC
 (1)  4311  7F               4.0                        MOV     A,A             ;RETRY LINK
1387  4312  32  49  49      13.0                STA     UNITMP          ;STORE IN MEMORY
1388  4315  A7                4.0                ANA     A               ;SET THE CONDITION CODE
1389  4316  C2  27  43      10.0                JNZ     TST01X          ;GO TEST THE SPECIFIED UNITS
1390                                                                    ;NO UNITS SPECIFIED
1391  4319                                      ERR     EXIT,DUMMY
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28      18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0001                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                          .BYTE
 (1)  431E  CD  15  28      18.0                DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  92  47      10.0                        JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1392                                    ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1393                                    ;>FATAL ERROR - TEST ABORTED
1394  4324  C3  92  47      10.0                JMP     EXIT            ;EXIT THE TEST
1395
1396  4327  3A  49  49      13.0  TST01X: LDA     UNITMP          ;GET THE UNIT MAP
1397  432A  E6  01           7.0          ANI     @001            ;TEST UNIT 0?
1398  432C  CA  4A  43      10.0          JZ      TEST2           ;NO-CHECK FOR TEST2
1399                                                              ;YES-RUN THE TEST
1400  432F  AF                4.0          XRA     A               ;GET THE PORT # FOR TEST
1401  4330  CD  98  47      18.0          CALL    CLEAR           ;CLEAR ALL PORTS
1402  4333  CD  C3  47      18.0          CALL    CONTR1          ;DO THE TEST
1403
1404  4336                              ENDTST  TST01X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4336                                  REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4336  CD  06  28      18.0                        CALL    REQST
 (2)  4339  00                                          .BYTE                   ;DATA PATTERN NUMBER
 (2)  433A  00  00                                      .WORD                   ;SYSTEM "" COUNT
```

```
   (2)  433C  00  00                                    .WORD                ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
   (2)  433E  00                                        .BYTE                ;DATA COMPARE FLAG IF =1
   (2)  433F  07                                        .BYTE   7            ;REQUEST CODE
   (1)  4340  3A  9A  4F        13.0          LDA      ITERA                ;GET ITERATION COUNT
   (1)  4343  3D                 4.0          DCR      A                    ;DOWNCOUNT
   (1)  4344  32  9A  4F        13.0          STA      ITERA                ;SAVE COUNT
   (1)  4347  F2  27  43        10.0          JP       TST01X               ;DO TEST UNTIL TILL = 0
  1405
```

```
1407                                .SBTTL  TEST  2 - LOOP CMD. TO STATUS - TU PORT #1
1408  434A                          ST
 (1)                                ; ***************************************************************************
 (1)                                ;*TEST TITLE
 (1)                                ;*---- -----
1409                                ;*LOOP COMMAND TO STATUS-TU PORT #1
1410  434A                          SD
 (1)                                ; ***************************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
1411                                ;*THIS TEST CHECKS THE DATA PATH BETWEEN THE TU PORT #1 COMMAND AND
1412                                ;*STATUS REGISTER AND VERIFIES THE OPERATIONALITY OF THE COMMAND AND
1413                                ;*STATUS BITS.  THE ABSENCE OF BIT COUPLING IS ALSO TESTED.
1414  434A                          SP
 (1)                                ; ***************************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
1415                                ;*BGNTST
1416                                ;*    IF TU PORT #1 NOT SELECTED BY USER
1417                                ;*    :   THEN-NEXT TEST
1418                                ;*    :   ELSE-CONTINUE
1419                                ;*    ENDIF
1420                                ;*    LOAD TU PORT #1 SELECT CODE
1421                                ;*    CALL SUBROUTINE CLEAR
1422                                ;*    CALL SUBROUTINE CONTR1
1423                                ;*ENDTST
1424  434A                          SE
 (1)                                ; ***************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1425                                ;*TUP1 MICRO TEST 02
1426                                ;*TUP1 MICRO ERROR 14
1427                                ;*TUP1-LOOP CMD. TO STATUS-TU PORT #1
1428                                ;*M8955'S, M8954, M8960
1429                                ;*DATA WRITTEN TO CMD. NOT = DATA READ AT STATUS
1430                                ;*ACTUAL = NNNN
1431                                ;*EXPECTED = NNNN
1432                                ;*
1433                                ;*TUP1 MICRO TEST 02
1434                                ;*TUP1 MICRO ERROR 15
1435                                ;*TUP1-LOOP CMD. TO STATUS-TU PORT #1
1436                                ;*M8955'S, M8954, M8960
1437                                ;*PTY. BIT NOT = 1 FOR DATA
1438                                ;*ACTUAL = NNNN
1439                                ;*
1440                                ;*TUP1 MICRO TEST 02
1441                                ;*TUP1 MICRO ERROR 16
1442                                ;*TUP1-LOOP CMD. TO STATUS-TU PORT #1
1443                                ;*M8955'S, M8954, M8960
1444                                ;*PTY. BIT NOT = 0 FOR DATA
1445                                ;*ACTUAL = NNNN
1446  434A                          S
 (1)                                ; ***************************************************************************
1447
```

```
1448  434A                                TEST2:  TESTX   $2              ;SET UP THE TEST NUMBER
 (1)  434A  3E  02              7.0                      MVI     A,$2            ;DEFINE THE TEST NUMBER
 (1)  434C  CD  03  28         18.0                      CALL    TSET            ;SETUP THE TEST
1449                                ;%TUP1-LOOP CMD. TO STATUS-TU PORT #1
1450                                ;@M8955'S, M8954, M8960
1451  434F  3A  49  49         13.0  TST02X: LDA     UNITMP          ;GET THE UNIT MAP
1452  4352  E6  02              7.0          ANI     @002            ;TEST UNIT 1?
1453  4354  CA  73  43         10.0          JZ      TEST3           ;NO-CHECK FOR TEST3
1454  4357  3E  01              7.0          MVI     A,@01           ;LOAD UNIT 1 SELECT CODE
1455  4359  CD  98  47         18.0          CALL    CLEAR           ;CLEAR ALL PORTS
1456  435C  CD  C3  47         18.0          CALL    CONTR1          ;TEST THIS PORT
1457  435F                                  ENDTST  TST02X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  435F                                  REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  435F  CD  06  28         18.0                      CALL    REQST
 (2)  4362  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  4363  00  00                                       .WORD                   ;SYSTEM '*' COUNT
 (2)  4365  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4367  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4368  07                                           .BYTE   7               ;REQUEST CODE
 (1)  4369  3A  9A  4F         13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  436C  3D                  4.0          DCR     A               ;.DOWNCOUNT
 (1)  436D  32  9A  4F         13.0          STA     ITERA           ;SAVE COUNT
 (1)  4370  F2  4F  43         10.0          JP      TST02X          ;DO TEST UNTIL TILL = 0
1458
```

```
                                .SBTTL  TEST  3 - LOOP CMD. TO STATUS - TU PORT #2
1461  4373                      ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1462                            ;*LOOP COMMAND TO STATUS-TU PORT #2
1463  4373                      SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1464                            ;*THIS TEST CHECKS THE DATA PATH BETWEEN THE TU PORT #2 COMMAND AND
1465                            ;*STATUS REGISTER AND VERIFIES THE OPERATIONALITY OF THE COMMAND AND
1466                            ;*STATUS BITS.  THE ABSENCE OF BIT COUPLING IS ALSO TESTED.
1467  4373                      SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1468                            ;*BGNTST
1469                            ;*    IF TU PORT #2 NOT SELECTED BY USER
1470                            ;*    :   THEN-NEXT TEST
1471                            ;*    :   ELSE-CONTINUE
1472                            ;*    ENDIF
1473                            ;*    LOAD TU PORT #2 SELECT CODE
1474                            ;*    CALL SUBROUTINE CLEAR
1475                            ;*    CALL SUBROUTINE CONTR1
1476                            ;*ENDTST
1477  4373                      SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1478                            ;*TUP1 MICRO TEST 03
1479                            ;*TUP1 MICRO ERROR 14
1480                            ;*TUP1-LOOP CMD. TO STATUS-TU PORT #2
1481                            ;*M8955'S, M8954, M8960
1482                            ;*DATA WRITTEN TO CMD. NOT = DATA READ AT STATUS
1483                            ;*ACTUAL = NNNN
1484                            ;*EXPECTED = NNNN
1485                            ;*
1486                            ;*TUP1 MICRO TEST 03
1487                            ;*TUP1 MICRO ERROR 15
1488                            ;*TUP1-LOOP CMD. TO STATUS-TU PORT #2
1489                            ;*M8955'S, M8954, M8960
1490                            ;*PTY. BIT NOT = 1 FOR DATA
1491                            ;*ACTUAL = NNNN
1492                            ;*
1493                            ;*TUP1 MICRO TEST 03
1494                            ;*TUP1 MICRO ERROR 16
1495                            ;*TUP1-LOOP CMD. TO STATUS-TU PORT #2
1496                            ;*M8955'S, M8954, M8960
1497                            ;*PTY. BIT NOT = 0 FOR DATA
1498                            ;*ACTUAL = NNNN
1499  4373                      S
 (1)                            ; ************************************************************************
1500
```

```
1501  4373                                  TEST3:  TESTX   $3               ;SET UP THE TEST NUMBER
 (1)  4373  3E  03                7.0                    MVI     A,$3             ;DEFINE THE TEST NUMBER
 (1)  4375  CD  03  28           18.0                    CALL    TSET             ;SETUP THE TEST
1502                                    ;%TUP1-LOOP CMD. TO STATUS-TU PORT #2
1503                                    ;&M8955'S, M8954, M8960
1504  4378  3A  49  49           13.0  TST03X: LDA     UNITMP           ;GET THE UNIT MAP
1505  437B  E6  04                7.0           ANI     @004             ;TEST FOR UNIT 2
1506  437D  CA  9C  43            10.0           JZ      TEST4            ;NO-CHECK FOR TEST4
1507  4380  3E  02                7.0           MVI     A,@02            ;GET PORT SELECT #
1508  4382  CD  98  47            18.0           CALL    CLEAR            ;CLEAR ALL PORTS
1509  4385  CD  C3  47            18.0           CALL    CONTR1           ;TEST THIS PORT
1510  4388                                       ENDTST  TST03X
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4388                                     REQ     7                ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4388  CD  06  28           18.0                    CALL    REQST
 (2)  438B  00                                           .BYTE                    ;DATA PATTERN NUMBER
 (2)  438C  00  00                                       .WORD                    ;SYSTEM '"' COUNT
 (2)  438E  00  00                                       .WORD                    ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4390  00                                           .BYTE                    ;DATA COMPARE FLAG IF =1
 (2)  4391  07                                           .BYTE   7                ;REQUEST CODE
 (1)  4392  3A  9A  4F           13.0           LDA     ITERA            ;GET ITERATION COUNT
 (1)  4395  3D                    4.0           DCR     A                ;DOWNCOUNT
 (1)  4396  32  9A  4F           13.0           STA     ITERA            ;SAVE COUNT
 (1)  4399  F2  78  43           10.0           JP      TST03X           ;DO TEST UNTIL TILL = 0
1511
```

```
                                .SBTTL  TEST  4 - LOOP CMD. TO STATUS - TU PORT #3
439C                            ST
                                ;*****************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP COMMAND TO STATUS-TU PORT #3
439C                            SD
                                ;*****************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE DATA PATH BETWEEN THE TU PORT #3 COMMAND AND
                                ;*STATUS REGISTER AND VERIFIES THE OPERATIONALITY OF THE COMMAND AND
                                ;*STATUS BITS.  THE ABSENCE OF BIT COUPLING IS ALSO TESTED.
439C                            SP
                                ;*****************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #3 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    LOAD TU PORT #3 SELECT CODE
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    CALL SUBROUTINE CONTR1
                                ;*ENDTST
439C                            SE
                                ;*****************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP1 MICRO TEST 04
                                ;*TUP1 MICRO ERROR 14
                                ;*TUP1-LOOP CMD. TO STATUS-TU PORT #3
                                ;*M8955'S, M8954, M8960
                                ;*DATA WRITTEN TO CMD. NOT = DATA READ AT STATUS
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 04
                                ;*TUP1 MICRO ERROR 15
                                ;*TUP1-LOOP CMD. TO STATUS-TU PORT #3
                                ;*M8955'S, M8954, M8960
                                ;*PTY. BIT NOT = 1 FOR DATA
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 04
                                ;*TUP1 MICRO ERROR 16
                                ;*TUP1-LOOP CMD. TO STATUS-TU PORT #3
                                ;*M8955'S, M8954, M8960
                                ;*PTY. BIT NOT = 0 FOR DATA
                                ;*ACTUAL = NNNN
439C                            S
                                ;*****************************************************************
```

```
1554  439C                                  TEST4:  TESTX   $4               ;SET UP THE TEST NUMBER
 (1)  439C  3E  04              7.0                      MVI     A,$4        ;DEFINE THE TEST NUMBER
 (1)  439E  CD  03  28         18.0                      CALL    TSET        ;SETUP THE TEST
1555                                    ;%TUP1-LOOP CMD. TO STATUS-TU PORT #3
1556                                    ;&M8955'S, M8954, M8960
1557  43A1  3A  49  49         13.0  TST04X: LDA     UNITMP                  ;GET THE UNIT MAP
1558  43A4  E6  08                7.0          ANI     @010                  ;TEST FOR UNIT 3?
1559  43A6  CA  C5  43          10.0          JZ      TEST5                 ;NO-CHECK FOR TEST5
1560  43A9  CD  98  47          18.0          CALL    CLEAR                 ;CLEAN UP THE PORTS
1561  43AC  3E  03                7.0          MVI     A,@03                 ;
1562  43AE  CD  C3  47          18.0          CALL    CONTR1                ;
1563  43B1                                    ENDTST  TST04X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43B1                                      REQ     7                     ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43B1  CD  06  28         18.0                      CALL    REQST
 (2)  43B4  00                                           .BYTE                 ;DATA PATTERN NUMBER
 (2)  43B5  00  00                                       .WORD                 ;SYSTEM '"' COUNT
 (2)  43B7  00  00                                       .WORD                 ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43B9  00                                           .BYTE                 ;DATA COMPARE FLAG IF =1
 (2)  43BA  07                                           .BYTE   7             ;REQUEST CODE
 (1)  43BB  3A  9A  4F         13.0          LDA     ITERA                   ;GET ITERATION COUNT
 (1)  43BE  3D                 4.0          DCR     A                       ;DOWNCOUNT
 (1)  43BF  32  9A  4F         13.0          STA     ITERA                   ;SAVE COUNT
 (1)  43C2  F2  A1  43         10.0          JP      TST04X                  ;DO TEST UNTIL TILL = 0
1564
```

```
                                .SBTTL  TEST  5 - WCS PARITY INT. TEST - TU PORT #0
 43C5                           ST
  (1)                           ; ***************************************************************
  (1)                           ;*TEST TITLE
  (1)                           ;*---- -----
                                ;*WCS PARITY ERROR INTERRUPT TEST-TU PORT #0
 43C5                           SD
  (1)                           ; ***************************************************************
  (1)                           ;*DESCRIPTION
  (1)                           ;*-----------
                                ;*THIS TEST CHECKS THE PROPER OPERATION OF THE TU PORT #0 PARITY ERROR
                                ;*INTERRUPT LOGIC.
 43C5                           SP
  (1)                           ; ***************************************************************
  (1)                           ;*PROCEDURE
  (1)                           ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #0 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    LOAD TU PORT #0 SELECT CODE
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    CALL SUBROUTINE CONTR3
                                ;*ENDTST
 43C5                           SE
  (1)                           ; ***************************************************************
  (1)                           ;*ERRORS
  (1)                           ;*------
                                ;*TUP1 MICRO TEST 05
                                ;*TUP1 MICRO ERROR 21
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 0
                                ;*M8955'S, M8954
                                ;*STATUS PARITY ERROR BIT NOT SET ON DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 05
                                ;*TUP1 MICRO ERROR 22
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 0
                                ;*M8955'S, M8954
                                ;*NO INTERRUPT WHEN ONE EXPECTED ON DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 05
                                ;*TUP1 MICRO ERROR 23
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 0
                                ;*M8955'S, M8954
                                ;*STATUS PARITY ERROR BIT NOT EXPECTED FOR DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 05
                                ;*TUP1 MICRO ERROR 24
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 0
                                ;*M8955'S, M8954
```

```
1608                                    ;*INTER. WHEN NONE EXP.
1609                                    ;*ACTUAL = NNNN
1610  43C5                              $
 (1)                                    ; ************************************************************************
1611
1612  43C5                              TEST5:  TESTX   @05             ;INITIALIZE THE TEST
 (1)  43C5  3E  05           7.0                        MVI     A,@05           ;DEFINE THE TEST NUMBER
 (1)  43C7  CD  03  28      18.0                        CALL    TSET            ;SETUP THE TEST
1613                                    ;%TUP1-WCS PTY. ERROR INTER.-TU PORT # 0
1614                                    ;@M8955'S, M8954
1615  43CA  3A  49  49      13.0        TST05X: LDA     UNITMP          ;GET THE UNIT MAP
1616  43CD  E6  01           7.0                ANI     @001            ;TEST FOR UNIT 0?
1617  43CF  CA  ED  43      10.0                JZ      TEST6
1618  43D2  AF                4.0                XRA     A               ;GET PORT 0 SELECT CODE
1619  43D3  CD  98  47      18.0                CALL    CLEAR           ;CLEAN UP THE PORTS
1620  43D6  CD  84  48      18.0                CALL    CONTR3          ;ISSUE THE TEST
1621  43D9                                      ENDTST  TST05X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43D9                                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43D9  CD  06  28      18.0                        CALL    REQST
 (2)  43DC  00                                          .BYTE                   ;DATA PATTERN NUMBER
 (2)  43DD  00  00                                      .WORD                   ;SYSTEM '"' COUNT
 (2)  43DF  00  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43E1  00                                          .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  43E2  07                                          .BYTE   7               ;REQUEST CODE
 (1)  43E3  3A  9A  4F      13.0                LDA     ITERA           ;GET ITERATION COUNT
 (1)  43E6  3D                4.0                DCR     A               ;DOWNCOUNT
 (1)  43E7  32  9A  4F      13.0                STA     ITERA           ;SAVE COUNT
 (1)  43EA  F2  CA  43      10.0                JP      TST05X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST  6 - WCS PARITY INT. - TU PORT #1
43ED                            ST
 (1)                            ;*****************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*WCS PARITY ERROR INTERRUPT TEST-TU PORT #1
43ED                            SD
 (1)                            ;*****************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE PROPER OPERATION OF THE TU PORT #1 PARITY ERROR
                                ;*INTERRUPT LOGIC.
43ED                            SP
 (1)                            ;*****************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #1 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    LOAD TU PORT #1 SELECT CODE
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    CALL SUBROUTINE CONTR3
                                ;*ENDTST
43ED                            SE
 (1)                            ;*****************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*TUP1 MICRO TEST 06
                                ;*TUP1 MICRO ERROR 21
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 1
                                ;*M8955'S, M8954
                                ;*STATUS PARITY ERROR BIT NOT SET ON DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 06
                                ;*TUP1 MICRO ERROR 22
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 1
                                ;*M8955'S, M8954
                                ;*NO INTERRUPT WHEN ONE EXPECTED ON DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 06
                                ;*TUP1 MICRO ERROR 23
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 1
                                ;*M8955'S, M8954
                                ;*STATUS PARITY ERROR BIT NOT EXPECTED FOR DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 06
                                ;*TUP1 MICRO ERROR 24
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 1
                                ;*M8955'S, M8954
```

```
1665                                    ;*INTER. WHEN NONE EXP.
1666                                    ;*ACTUAL = NNNN
1667  43ED                              $
 (1)                                    ; ************************************************************************
1668
1669  43ED                              TEST6:  TESTX   @06             ;INITIALIZE THE TEST
 (1)  43ED  3E  06          7.0                      MVI     A,@06       ;DEFINE THE TEST NUMBER
 (1)  43EF  CD  03  28     18.0                      CALL    TSET        ;SETUP THE TEST
1670                                    ;%TUP1-WCS PTY. ERROR INTER.-TU PORT # 1
1671                                    ;@M8955'S, M8954
1672  43F2  3A  49  49     13.0        TST06X: LDA     UNITMP          ;GET THE UNIT MAP
1673  43F5  E6  02           7.0                ANI     @002            ;TEST FOR UNIT 1?
1674  43F7  CA  16  44      10.0                JZ      TEST7
1675  43FA  3E  01           7.0                MVI     A,1             ;GET PORT #
1676  43FC  CD  98  47      18.0                CALL    CLEAR           ;CLEAN UP THE PORTS
1677  43FF  CD  84  48      18.0                CALL    CONTR3          ;DO THE TEST
1678  4402                                      ENDTST  TST06X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4402                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4402  CD  06  28     18.0                      CALL    REQST
 (2)  4405  00                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  4406  00  00                                  .WORD                   ;SYSTEM '"' COUNT
 (2)  4408  00  00                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  440A  00                                      .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  440B  07                                      .BYTE   7               ;REQUEST CODE
 (1)  440C  3A  9A  4F     13.0                LDA     ITERA           ;GET ITERATION COUNT
 (1)  440F  3D              4.0                DCR     A               ;DOWNCOUNT
 (1)  4410  32  9A  4F     13.0                STA     ITERA           ;SAVE COUNT
 (1)  4413  F2  F2  43     10.0                JP      TST06X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST  7 - WCS PARITY INT. - TU PORT #2
4416                            ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*WCS PARITY ERROR INTERRUPT TEST-TU PORT #2
4416                            SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE PROPER OPERATION OF THE TU PORT #2 PARITY ERROR
                                ;*INTERRUPT LOGIC.
4416                            ;
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #2 NOT SELECTED BY USER
                                ;*     :   THEN-NEXT TEST
                                ;*     :   ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     LOAD TU PORT #2 SELECT CODE #
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     CALL SUBROUTINE CONTR3
                                ;*ENDTST
4416                            SE
 (1)                            ;*********************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*TUP1 MICRO TEST 07
                                ;*TUP1 MICRO ERROR 21
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 2
                                ;*M8955'S, M8954
                                ;*STATUS PARITY ERROR BIT NOT SET ON DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 07
                                ;*TUP1 MICRO ERROR 22
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 2
                                ;*M8955'S, M8954
                                ;*NO INTERRUPT WHEN ONE EXPECTED ON DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 07
                                ;*TUP1 MICRO ERROR 23
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 2
                                ;*M8955'S, M8954
                                ;*STATUS PARITY ERROR BIT NOT EXPECTED FOR DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 07
                                ;*TUP1 MICRO ERROR 24
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 2
                                ;*INTER. WHEN NONE EXP.
```

```
1722                                    ;*ACTUAL = NNNN
1723  4416                              S
 (1)                                    ; ***************************************************************************
1724
1725  4416                              TEST7:  TESTX   @07              ;INITIALIZE THE TEST
 (1)  4416  3E  07              7.0                     MVI     A,@07           ;DEFINE THE TEST NUMBER
 (1)  4418  CD  03  28         18.0                     CALL    TSET            ;SETUP THE TEST
1726                                    ;%TUP1-WCS PTY. ERROR INTER.-TU PORT # 2
1727                                    ;&M8955'S, M8954
1728  441B  3A  49  49         13.0    TST07X: LDA     UNITMP          ;GET THE UNIT MAP
1729  441E  E6  04                   7.0             ANI     @004            ;TEST FOR UNIT 2?
1730  4420  CA  3F  44         10.0             JZ      TEST10
1731  4423  3E  02                   7.0             MVI     A,2             ;GET PORT #
1732  4425  CD  98  47         18.0             CALL    CLEAR           ;CLEAN UP THE PORTS
1733  4428  CD  84  48         18.0             CALL    CONTR3          ;DO THE TEST
1734  442B                             ENDTST  TST07X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  442B                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  442B  CD  06  28         18.0                     CALL    REQST
 (2)  442E  00                                          .BYTE                   ;DATA PATTERN NUMBER
 (2)  442F  00  00                                      .WORD                   ;SYSTEM "" COUNT
 (2)  4431  00  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4433  00                                          .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  4434  07                                          .BYTE   7               ;REQUEST CODE
 (1)  4435  3A  9A  4F         13.0             LDA     ITERA           ;GET ITERATION COUNT
 (1)  4438  3D                   4.0             DCR     A               ;DOWNCOUNT
 (1)  4439  32  9A  4F         13.0             STA     ITERA           ;SAVE COUNT
 (1)  443C  F2  1B  44         10.0             JP      TST07X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 10 - WCS PARITY INT. - TU PORT #3
443F                            ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*WCS PARITY ERROR INTERRUPT TEST-TU PORT #3
443F                            SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE PROPER OPERATION OF THE TU PORT #3 PARITY ERROR
                                ;*INTERRUPT LOGIC.
443F                            SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #3 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    LOAD TU PORT #3 SELECT CODE
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    CALL SUBROUTINE CONTR3
                                :*ENDTST
443F                            SE
 (1)                            ;*********************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*TUP1 MICRO TEST 10
                                ;*TUP1 MICRO ERROR 21
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 3
                                ;*M8955'S, M8954
                                ;*STATUS PARITY ERROR BIT NOT SET ON DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 10
                                ;*TUP1 MICRO ERROR 22
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 3
                                ;*M8955'S, M8954
                                ;*NO INTERRUPT WHEN ONE EXPECTED ON DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 10
                                ;*TUP1 MICRO ERROR 23
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 3
                                ;*M8955'S, M8954
                                ;*STATUS PARITY ERROR BIT NOT EXPECTED FOR DATA BYTE (ACTUAL)
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 10
                                ;*TUP1 MICRO ERROR 24
                                ;*TUP1-WCS PTY. ERROR INTER.-TU PORT # 3
                                ;*M8955'S, M8954
```

```
                                        ;*INTER. WHEN NONE EXP.
                                        ;*ACTUAL = NNNN
443F                                    $
 (1)                                    ; ************************************************************************
                                        
                                        TEST10: TESTX   @010            ;INITIALIZE THE TEST
 (1)  443F  3E  08          7.0                         MVI     A,@010      ;DEFINE THE TEST NUMBER
 (1)  4441  CD  03  28     18.0                         CALL    TSET        ;SETUP THE TEST
                                        ;%TUP1-WCS PTY. ERROR INTER.-TU PORT # 3
                                        ;&M8955'S, M8954
4444  3A  49  49           13.0 TST10X: LDA     UNITMP          ;GET THE UNIT MAP
4447  E6  08            7.0          ANI     @010            ;TEST FOR UNIT 3?
4449  CA  68  44         10.0          JZ      TEST11
444C  3E  03            7.0          MVI     A,3             ;GET THE PORT #
444E  CD  98  47         18.0         CALL    CLEAR           ;CLEAN UP THE PORTS
4451  CD  84  48         18.0         CALL    CONTR3          ;DO THE TEST
4454                                   ENDTST  TST10X
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4454                             REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4454  CD  06  29     18.0                         CALL    REQST
 (2)  4457  00                                          .BYTE                   ;DATA PATTERN NUMBER
 (2)  4458  00  00                                      .WORD                   ;SYSTEM '*' COUNT
 (2)  445A  00  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  445C  C0                                          .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  445D  07                                          .BYTE   7               ;REQUEST CODE
 (1)  445E  3A  9A  4F     13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4461  3D             4.0          DCR     A               ;DOWNCOUNT
 (1)  4462  32  9A  4F     13.0          STA     ITERA           ;SAVE COUNT
 (1)  4465  F2  44  44     10.0          JP      TST10X          ;DO TEST UNTIL TILL = 0
```

```
                                        .SBTTL  TEST 11 - LOOP AMTIE TO AMTIE - TU PORT #0
4468                                    ST
 (1)                                    ; ***************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
                                        ;*LOOP AMTIE TO AMTIE TEST-TU PORT #0
4468                                    SD
 (1)                                    ; ***************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS TEST CHECKS THE INTEGRITY OF THE AMTIE LINES THROUGH THE TU PORT
                                        ;*#0 AMTIE REGISTER.
4468                                    SP
 (1)                                    ; ***************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNTST
                                        ;*    IF TU PORT #0 NOT SELECTED BY USER
                                        ;*    :   THEN-NEXT TEST
                                        ;*    :   ELSE-CONTINUE
                                        ;*    ENDIF
                                        ;*    CALL SUBROUTINE CLEAR
                                        ;*    CALL SUBROUTINE CONTR2
                                        ;*ENDTST
4468                                    SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
                                        ;*TUP1 MICRO TEST 11
                                        ;*TUP1 MICRO ERROR 17
                                        ;*TUP1-LOOP AMTIE TO AMTIE-TU PORT #0
                                        ;*M8955'S, M8954
                                        ;*DATA WRITTEN TO AMTIE NOT = DATA READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*TUP1 MICRO TEST 11
                                        ;*TUP1 MICRO ERROR 20
                                        ;*TUP1-LOOP AMTIE TO AMTIE-TU PORT #0
                                        ;*M8955'S, M8954
                                        ;*VALUE WRITTEN TO AMTIE PTY. NOT = VALUE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4468                                    S
 (1)                                    ; ***************************************************************************

4468                                    TEST11: TESTX   @11             ;SET UP THE TEST NUMBER
 (1) 4468  3E  09          7.0                       MVI     A,@11       ;DEFINE THE TEST NUMBER
 (1) 446A  CD  03  28     18.0                       CALL    TSET        ;SETUP THE TEST
                                        ;%TUP1-LOOP AMTIE TO AMTIE-TU PORT #0
                                        ;&M8955'S, M8954
446D  ?A  49  49          13.0  TST11X: LDA     UNITMP          ;GET THE UNIT MAP
4470  .6  01               7.0         ANI     @001            ;TEST UNIT 0?
4472  CA  91  44          10.0         JZ      TEST12          ;NO-CHECK FOR TEST12
```

```
1832                                                                    ;YES-RUN THE TEST
1833  4475  3E  00          7.0           MVI     A,@00
1834  4477  CD  98  47      18.0          CALL    CLEAR                 ;CLEAN UP THE PORTS
1835  447A  CD  27  48      18.0          CALL    CONTR2                ;
1836  447D                                ENDTST  TST11X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  447D                          REQ     7                         ;FAKE CALL TO KEEP TEST ALIVE
 (2)  447D  CD  06  28      18.0                                CALL    REQST
 (2)  4480  00                                                  .BYTE                   ;DATA PATTERN NUMBER
 (2)  4481  00  00                                              .WORD                   ;SYSTEM "" COUNT
 (2)  4483  00  00                                              .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4485  00                                                  .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4486  07                                                  .BYTE   7               ;REQUEST CODE
 (1)  4487  3A  4A  4F      13.0          LDA     ITERA                 ;GET ITERATION COUNT
 (1)  448A  3D               4.0          DCR     A                     ;DOWNCOUNT
 (1)  448B  32  9A  4F      13.0          STA     ITERA                 ;SAVE COUNT
 (1)  448E  F2  6D  44      10.0          JP      TST11X                ;DO TEST UNTIL TILL = 0
```

```
                                        .SBTTL  TEST 12 - LOOP AMTIE TO AMTIE - TU PORT #1
 4491                                   ST
  (1)                                   ; ************************************************************************
  (1)                                   ;*TEST TITLE
  (1)                                   ;*---- -----
                                        ;*LOOP AMTIE TO AMTIE TEST-TU PORT #1
 4491                                   SD
  (1)                                   ; ************************************************************************
  (1)                                   ;*DESCRIPTION
  (1)                                   ;*-----------
                                        ;*THIS TEST CHECKS THE INTEGRITY OF THE AMTIE LINES THROUGH THE TU PORT
                                        ;*#1 AMTIE REGISTER.
 4491                                   SP
  (1)                                   ; ************************************************************************
  (1)                                   ;*PROCEDURE
  (1)                                   ;*---------
                                        ;*BGNTST
                                        ;*    IF TU PORT #1 NOT SELECTED BY USER
                                        ;*    :   THEN-NEXT TEST
                                        ;*    :   ELSE-CONTINUE
                                        ;*    ENDIF
                                        ;*    CALL SUBROUTINE CLEAR
                                        ;*    CALL SUBROUTINE CONTR2
                                        ;*ENDTST
 4491                                   SE
  (1)                                   ; ************************************************************************
  (1)                                   ;*ERRORS
  (1)                                   ;*------
                                        ;*TUP1 MICRO TEST 12
                                        ;*TUP1 MICRO ERROR 17
                                        ;*TUP1-LOOP AMTIE TO AMTIE-TU PORT #1
                                        ;*M8955'S, M8954
                                        ;*DATA WRITTEN TO AMTIE NOT = DATA READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*TUP1 MICRO TEST 12
                                        ;*TUP1 MICRO ERROR 20
                                        ;*TUP1-LOOP AMTIE TO AMTIE-TU PORT #1
                                        ;*M8955'S, M8954
                                        ;*VALUE WRITTEN TO AMTIE PTY. NOT = VALUE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
 4491                                   S
  (1)                                   ; ************************************************************************

 4491                                   TEST12: TESTX   @12             ;SET UP THE TEST NUMBER
  (1)  4491  3E  0A          7.0                      MVI     A,@12       ;DEFINE THE TEST NUMBER
  (1)  4493  CD  03  28     18.0                      CALL    TSET        ;SETUP THE TEST
                                        ;%TUP1-LOOP AMTIE TO AMTIE-TU PORT #1
                                        ;&M8955'S, M8954
 4496  3A  49  49          13.0  TST12X: LDA     UNITMP          ;GET THE UNIT MAP
 4499  E6  02               7.0         ANI     @002            ;TEST UNIT 1?
 449B  CA  BA  44          10.0         JZ      TEST13          ;NO-CHECK FOR TEST13
```

```
1877  449E  3E  01        7.0        MVI   A,@01          ;
1878  44A0  CD  98  47    18.0       CALL  CLEAR          ;CLEAN UP PORTS
1879  44A3  CD  27  48    18.0       CALL  CONTR2         ;
1880  44A6                          ENDTST TST12X
 (1)                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44A6                          REQ   7              ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44A6  CD  06  28    18.0                    CALL    REQST
 (2)  44A9  00                                    .BYTE                 ;DATA PATTERN NUMBER
 (2)  44AA  00  00                                .WORD                 ;SYSTEM "" COUNT
 (2)  44AC  00  00                                .WORD                 ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  44AE  00                                    .BYTE                 ;DATA COMPARE FLAG IF =1
 (2)  44AF  07                                    .BYTE   7             ;REQUEST CODE
 (1)  44B0  3A  9A  4F    13.0       LDA   ITERA          ;GET ITERATION COUNT
 (1)  44B3  3D        4.0        DCR   A              ;DOWNCOUNT
 (1)  44B4  32  9A  4F    13.0       STA   ITERA          ;SAVE COUNT
 (1)  44B7  F2  96  44    10.0       JP    TST12X         ;DO TEST UNTIL TILL = 0
```

```
                                        .SBTTL  TEST 13 - LOOP AMTIE TO AMTIE - TU PORT #2
44BA                                    ST
 (1)                                    ; ***************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
                                        ;*LOOP AMTIE TO AMTIE TEST-TU PORT #2
44BA                                    SD
 (1)                                    ; ***************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS TEST CHECKS THE INTEGRITY OF THE AMTIE LINES THROUGH THE TU PORT
                                        ;*#2 AMTIE REGISTER.
44BA                                    SP
 (1)                                    ; ***************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNTST
                                        ;*    IF TU PORT #2 NOT SELECTED BY USER
                                        ;*    :   THEN-NEXT TEST
                                        ;*    :   ELSE-CONTINUE
                                        ;*    ENDIF
                                        ;*    CALL SUBROUTINE CLEAR
                                        ;*    CALL SUBROUTINE CONTR2
                                        ;*ENDTST
44BA                                    SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
                                        ;*------
                                        ;*TUP1 MICRO TEST 13
                                        ;*TUP1 MICRO ERROR 17
                                        ;*TUP1-LOOP AMTIE TO AMTIE-TU PORT #2
                                        ;*M8955'S, M8954
                                        ;*DATA WRITTEN TO AMTIE NOT = DATA READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*TUP1 MICRO TEST 13
                                        ;*TUP1 MICRO ERROR 20
                                        ;*TUP1-LOOP AMTIE TO AMTIE-TU PORT #2
                                        ;*M8955'S, M8954
                                        ;*VALUE WRITTEN TO AMTIE PTY. NOT = VALUE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
44BA                                    S
 (1)                                    ; ***  ***************************************************************************

44BA                                    TEST13: TESTX   @13             ;SET UP THE TEST NUMBER
44BA  3E  0B            7.0                             MVI     A,@13           ;DEFINE THE TEST NUMBER
 (1)  44BC  CD  03  28  18.0                            CALL    TSET            ;SETUP THE TEST
                                        ;%TUP1-LOOP AMTIE TO AMTIE-TU PORT #2
                                        ;&M8955'S, M8954
44BF  3A  49  49        13.0    TST13X: LDA     UNITMP          ;GET THE UNIT MAP
44C2  E6  04             7.0            ANI     @004            ;TEST FOR UNIT 2
44C4  CA  E3  44        10.0            JZ      TEST14          ;NO-CHECK FOR TEST14
```

```
 1921  44C7  3E  02          7.0          MVI     A,@02           ;
 1922  44C9  CD  98  47      18.0          CALL    CLEAR           ;CLEAN UP THE PORTS
 1923  44CC  CD  27  48      18.0          CALL    CONTR2          ;
 1924  44CF                               ENDTST  TST13X
  (1)                       ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  44CF                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  44CF  CD  06  28      18.0                         CALL    REQST
  (2)  44D2  00                                           .BYTE                   ;DATA PATTERN NUMBER
  (2)  44D3  00  00                                       .WORD                   ;SYSTEM '*' COUNT
  (2)  44D5  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  44D7  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  44D8  07                                           .BYTE   7               ;REQUEST CODE
  (1)  44D9  3A  9A  4F      13.0          LDA     ITERA           ;GET ITERATION COUNT
  (1)  44DC  3D               4.0          DCR     A               ;DOWNCOUNT
  (1)  44DD  32  9A  4F      13.0          STA     ITERA           ;SAVE COUNT
  (1)  44E0  F2  BF  44      10.0          JP      TST13X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 14 - LOOP AMTIE TO AMTIE - TU PORT #3
44E3                            $T
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*LOOP AMTIE TO AMTIE TEST-TU PORT #3
44E3                            $D
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE AMTIE LINES THROUGH THE TU PORT
                                ;*#3 AMTIE REGISTER.
44E3                            $P
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #3 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    CALL SUBROUTINE CONTR2
                                ;*ENDTST
44E3                            $E
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*TUP1 MICRO TEST 14
                                ;*TUP1 MICRO ERROR 17
                                ;*TUP1-LOOP AMTIE TO AMTIE-TU PORT #3
                                ;*M8955'S, M8954
                                ;*DATA WRITTEN TO AMTIE NOT = DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 14
                                ;*TUP1 MICRO ERROR 20
                                ;*TUP1-LOOP AMTIE TO AMTIE-TU PORT #3
                                ;*M8955'S, M8954
                                ;*VALUE WRITTEN TO AMTIE PTY. NOT = VALUE READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
44E3                            $
 (1)                            ; ************************************************************************

                                TEST14: TESTX   @014            ;SET UP THE TEST NUMBER
44E3  3E  0C            7.0                     MVI     A,@014          ;DEFINE THE TEST NUMBER
44E5  CD  03  28       18.0                     CALL    TSET            ;SETUP THE TEST
                                ;%TUP1-LOOP AMTIE TO AMTIE-TU PORT #3
                                ;&M8955'S, M8954
44E8  3A  49  49        13.0    TST14X: LDA     UNITMP          ;GET THE UNIT MAP
44EB  E6  08             7.0            ANI     @010            ;TEST FOR UNIT 3?
44ED  CA  0C  45        10.0            JZ      TEST15          ;NO-CHECK FOR TEST15
```

```
1965  44F0  3E  03        7.0        MVI     A,@03
1966  44F2  CD  98  47   18.0        CALL    CLEAR          ;CLEAN UP THE PORTS
1967  44F5  CD  27  48   18.0        CALL    CONTR2
1968  44F8                           ENDTST  TST14X
 (1)                           ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44F8                     REQ     7                  ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44F8  CD  06  28   18.0                      CALL    REQST
 (2)  44FB  00                                     .BYTE                  ;DATA PATTERN NUMBER
 (2)  44FC  00  00                                 .WORD                  ;SYSTEM "" COUNT
 (2)  44FE  00  00                                 .WORD                  ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4500  00                                     .BYTE                  ;DATA COMPARE FLAG IF =1
 (2)  4501  07                                     .BYTE   7              ;REQUEST CODE
 (1)  4502  3A  9A  4F   13.0        LDA     ITERA          ;GET ITERATION COUNT
 (1)  4505  3D            4.0        DCR     A              ;DOWNCOUNT
 (1)  4506  32  9A  4F   13.0        STA     ITERA          ;SAVE COUNT
 (1)  4509  F2  E8  44   10.0        JP      TST14X         ;DO TEST UNTIL TIEL = 0
```

```
                                .SBTTL  TEST 15 - LOOP "W/R ENA" TO "W/R ENA" - TU PORT #0
                                ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP W/R PATH ENABLE TO W/R PATH ENABLE TEST-TU PORT #0
                                SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE WRITE PATH AND READ PATH BITS TO BE
                                ;*SET AND RESET.
                                SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #0 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT TU PORT 0
                                ;*    SET THE WRITE PATH ENABLE BITS FOR TU PORT #0
                                 *    INPUT THE WRITE PATH ENABLE BITS FOR TU PORT #0
                                ;*    IF WRITE PATH ENABLE BIT SET
                                ;*    :   THEN-CONTINUE
                                ;*    :   ELSE-ERROR
                                ;*    ENDIF
                                ;*    RESET THE WRITE PATH ENABLE AND SET READ PATH ENABLE BITS FOR TU PORT #0
                                ;*    INPUT THE READ PATH ENABLE BIT FOR TU PORT #0
                                ;*    IF READ PATH ENABLE BITS RESET
                                ;*    :   THEN-CONTINUE
                                ;*    :   ELSE-ERROR
                                ;*    ENDIF
                                ;*ENDTST
                                SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP1 MICRO TEST 15
                                ;*TUP1 MICRO ERROR 02
                                ;*TUP1-LOOP "W/R ENA" TO "W/R ENA"-TU PORT #0
                                ;*M8955'S
                                ;*EXPECTED PORT 0 WRITE PATH ENAB SET & READ PATH ENAB CLEARED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 15
                                ;*TUP1 MICRO ERROR 03
                                ;*TUP1-LOOP "W/R ENA" TO "W/R ENA"-TU PORT #0
                                ;*M8955'S
                                ;*EXPECTED PORT 0 WRITE PATH ENAB CLEARED & READ PATH ENAB SET
                                ;*ACTUAL = NNNN
```

```
2012                                    ;*EXPECTED = NNNN
2013  450C                              S
 (1)                                    ; ************************************************************************
2014
2015  450C                              TES'15: TESTX   @15              ;INITIALIZE THE TEST
 (1)  450C  3E  0D            7.0                       MVI     A,@15         ;DEFINE THE TEST NUMBER
 (1)  450E  CD  03  28       18.0                       CALL    TSET          ;SETUP THE TEST
2016                                    ;%TUP1-LOOP "W/R ENA" TO "W/R ENA"-TU PORT #0
2017                                    ;&M8955'S
2018
2019  4511  3A  49  49        13.0  TST15X: LDA     UNITMP          ;GET THE UNIT MAP
2020  4514  E6  01             7.0          ANI     @01             ;TEST FOR UNIT 0
2021  4516  CA  7A  45        10.0          JZ      TEST16          ;NO-GO CHECK FOR TEST 16
2022                                       ;
2023  4519  CD  98  47        18.0          CALL    CLEAR           ;CLEAN UP THE PORTS
2024  451C  DB  E0             10.0          IN      INTSTA
2025  451E  E6  80              7.0          ANI     BIT7
2026  4520  C6  00              7.0          ADI     0
2027  4522  D3  E0             10.0          OUT     MBSEL           ;SELECT PORT 0
2028  4524  3E  10              7.0  DPEN00: MVI     A,P.WPEN        ;LOAD THE WRITE DATA PATH ENABLE BIT
2029  4526  D3  4C             10.0          OUT     PENAB           ;SET THE DATA PATH ENABLE BITS
2030  4528  00              4.0          NOP
2031  4529  DB  D1             10.0          IN      TUSEL0          ;GET THE ENABLE STATUS WORD
2032  452B  E6  3F              7.0          ANI     $3F             ;STRIP OUT THE DUAL TU PORT BIT
2033  452D  FE  08              7.0          CPI     P.WPOEN!0       ;SEE IF PROPER BITS SET
2034  452F  CA  3F  45        10.0          JZ      DPEN01          ;JUMP IF OK
2035  4532                                   ROUT    ADATA
 (1)  4532  D3  94             10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4534  7F              4.0                  MOV     A,A             ;RETRY LINK
2036  4535  3E  08              7.0          MVI     A,P.WPOEN!0
2037  4537                                   ROUT    EDATA
 (1)  4537  D3  95             10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4539  7F              4.0                  MOV     A,A             ;RETRY LINK
2038  453A                                   ERRB    DPEN00,DPEN01
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  453A  CD  12  28        18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0002                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  453D  02                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  453E  00                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  453F  CD  15  28        18.0  DPEN01::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4542  DA  24  45        10.0                  JC      DPEN00          ;LOOP ADDRESS IF LOOP SPECIFIED
2039                                    ;>EXPECTED PORT 0 WRITE PATH ENAB SET & READ PATH ENAB CLEARED
2040  4545  3E  04              7.0  DPEN02: MVI     A,P.RPEN        ;CLEAR WRITE PATH ENABLE & SET READ PATH ENABLE
2041  4547  D3  4C             10.0          OUT     PENAB           ;CLEAR THE DATA PATH ENABLE BITS
2042  4549  00              4.0          NOP
2043  454A  DB  D1             10.0          IN      TUSEL0          ;GET THE ENABLE STATUS WORD
2044  454C  E6  3F              7.0          ANI     $3F             ;STRIP OUT THE DUAL TU PORT BIT
2045  454E  FE  20              7.0          CPI     P.RPOEN!0        ;SEE IF PROPER BITS CLEAR
2046  4550  CA  60  45        10.0          JZ      DPEN03          ;JUMP IF OK
2047  4553                                   ROUT    ADATA
 (1)  4553  D3  94             10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4555  7F              4.0                  MOV     A,A             ;RETRY LINK
2048  4556  3E  20              7.0          MVI     A,P.RPOEN!0
2049  4558                                   ROUT    EDATA
```

```
  (1)  4558  D3  95          10.0                OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  455A  7F               4.0                MOV     A,A             ;RETRY LINK
 2050  455B                                 ERRB    DPEN02,DPEN03
  (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  455B  CD  12  28      18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0003                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  455E  03                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  455F  00                                      .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  4560  CD  15  28      18.0                DPEN03::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  4563  DA  45  45      10.0                    JC      DPEN02          ;LOOP ADDRESS IF LOOP SPECIFIED
 2051                                 ;>EXPECTED PORT 0 WRITE PATH ENAB CLEARED & READ PATH ENAB SET
 2052  4566                                 ENDTST  TST15X
  (1)                                     ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  4566                                 REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
  (2)  4566  CD  06  28      18.0                    CALL    REQST
  (2)  4569  00                                      .BYTE                   ;DATA PATTERN NUMBER
  (2)  456A  00  00                                  .WORD                   ;SYSTEM "" COUNT
  (2)  456C  00  00                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  456E  00                                      .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  456F  07                                      .BYTE   7               ;REQUEST CODE
  (1)  4570  3A  9A  4F      13.0                LDA     ITERA           ;GET ITERATION COUNT
  (1)  4573  3D               4.0                DCR     A               ;DOWNCOUNT
  (1)  4574  32  9A  4F      13.0                STA     ITERA           ;SAVE COUNT
  (1)  4577  F2  11  45      10.0                JP      TST15X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 16 - LOOP ''W/R ENA'' TO ''W/R ENA'' - TU PORT #1
457A                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*LOOP W/R PATH ENABLE TO W/R PATH ENABLE TEST-TU PORT #1
457A                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE WRITE PATH AND READ PATH BITS TO BE
                                ;*SET AND RESET.
457A                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #1 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT TU PORT #1
                                ;*    SET THE WRITE PATH ENABLE BITS FOR TU PORT #1
                                ;*    INPUT THE WRITE PATH ENABLE BITS FOR TU PORT #1
                                ;*    IF WRITE PATH ENABLE BIT SET
                                ;*    :   THEN-CONTINUE
                                ;*    :   ELSE-ERROR
                                ;*    ENDIF
                                ;*    RESET THE WRITE PATH ENABLE AND SET READ PATH ENABLE BITS FOR TU PORT #1
                                ;*    INPUT THE READ PATH ENABLE BIT FOR TU PORT #1
                                ;*    IF READ PATH ENABLE BITS RESET
                                ;*    :   THEN-CONTINUE
                                ;*    :   ELSE-ERROR
                                ;*    ENDIF
                                ;*ENDTST
457A                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*TUP1 MICRO TEST 16
                                ;*TUP1 MICRO ERROR 04
                                ;*TUP1-LOOP ''W/R ENA'' TO ''W/R ENA''-TU PORT #1
                                ;*M8955'S
                                ;*EXPECTED PORT 1 WRITE PATH ENAB SET & READ PATH ENAB CLEARED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 16
                                ;*TUP1 MICRO ERROR 05
                                ;*TUP1-LOOP ''W/R ENA'' TO ''W/R ENA''-TU PORT #1
                                ;*M8955'S
                                ;*EXPECTED PORT 1 WRITE PATH ENAB CLEAR & READ PATH ENAB SET
                                ;*ACTUAL = NNNN
```

```
2096                                    ;*EXPECTED = NNNN
2097  457A                              $
 (1)                                    ; ************************************************************************
2098
2099  457A                              TEST16: TESTX   @16              ;INITIALIZE THE TEST
 (1)  457A  3E  0E           7.0                        MVI     A,@16        ;DEFINE THE TEST NUMBER
 (1)  457C  CD  03  28      18.0                        CALL    TSET         ;SETUP THE TEST
2100                                    ;%TUP1-LOOP "W/R ENA" TO "W/R ENA"-TU PORT #1
2101                                    ;&M8955'S
2102
2103  457F  3A  49  49      13.0  TST16X: LDA     UNITMP           ;GET THE UNIT MAP
2104  4582  E6  02           7.0         ANI     @02              ;TEST FOR UNIT 1
2105  4584  CA  E8  45      10.0         JZ      TEST17           ;NO-GO CHECK FOR TEST 17
2106                                                               ;
2107  4587  CD  98  47      18.0         CALL    CLEAR            ;CLEAN UP THE PORTS
2108  458A  DB  E0          10.0         IN      INTSTA
2109  458C  E6  80           7.0         ANI     BIT7
2110  458E  C6  01           7.0         ADI     1
2111  4590  D3  E0          10.0         OUT     MBSEL            ;SELECT PORT 1
2112  4592  3E  10           7.0  DPEN10: MVI     A,P.WPEN         ;LOAD THE DATA PATH ENABLE BITS
2113  4594  D3  4C          10.0         OUT     PENAB            ;SET THE DATA PATH ENABLE BITS
2114  4596  00              4.0         NOP
2115  4597  DB  D1          10.0         IN      TUSELO           ;GET THE ENABLE STATUS WORD
2116  4599  E6  3F           7.0         ANI     $3F              ;STRIP OUT THE DUAL TU PORT BIT
2117  459B  FE  05           7.0         CPI     P.WP1EN!1        ;SEE IF PROPER BITS SET
2118  459D  CA  AD  45      10.0         JZ      DPEN11           ;JUMP IF OK
2119  45A0                              ROUT    ADATA
 (1)  45A0  D3  94          10.0                        OUT     ADATA        ;WRITE AC INTO ADATA
 (1)  45A2  7F              4.0                         MOV     A,A          ;RETRY LINK
2120  45A3  3E  05           7.0         MVI     A,P.WP1EN!1
2121  45A5                              ROUT    EDATA
 (1)  45A5  D3  95          10.0                        OUT     EDATA        ;WRITE AC INTO EDATA
 (1)  45A7  7F              4.0                         MOV     A,A          ;RETRY LINK
2122  45A8                              ERRB    DPEN10,DPEN11
 (1)                                    ,FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  45A8  CD  12  28      18.0                        CALL    ERLPB        ;PROCESS ERROR - DO 2.3
 (1)        0004                                      MSGN    =       MSGN+1       ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45AB  04                                        .BYTE   MSGN         ;MESSAGE NUMBER ID
 (1)  45AC  00                                        .BYTE                ;PRINT ROUTINE NUMBER
 (1)  45AD  CD  15  28      18.0         DPEN11::                CALL    CKLOP        ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45B0  DA  92  45      10.0                                 JC      DPEN10       ;LOOP ADDRESS IF LOOP SPECIFIED
2123                                    ;>EXPECTED PORT 1 WRITE PATH ENAB SET & READ PATH ENAB CLEARED
2124  45B3  3E  04           7.0  DPEN12: MVI     A,P.RPEN         ;CLEAR WRITE PATH ENAB. & SET REAR PATH
2125  45B5  D3  4C          10.0         OUT     PENAB            ;CLEAR THE DATA PATH ENABLE BITS
2126  45B7  00              4.0         NOP
2127  45B8  DB  D1          10.0         IN      TUSELO           ;GET THE ENABLE STATUS WORD
2128  45BA  E6  3F           7.0         ANI     $3F              ;STRIP OUT THE DUAL TU PORT BIT
2129  45BC  FE  11           7.0         CPI     P.RP1EN!1         ;SEE IF PROPER BITS CLEAR
2130  45BE  CA  CE  45      10.0         JZ      DPEN13           ;JUMP IF OK
2131  45C1                              ROUT    ADATA
 (1)  45C1  D3  94          10.0                        OUT     ADATA        ;WRITE AC INTO ADATA
 (1)  45C3  7F              4.0                         MOV     A,A          ;RETRY LINK
2132  45C4  3E  11           7.0         MVI     A,P.RP1EN!1
2133  45C6                              ROUT    EDATA
```

```
  (1)  45C6  D3  95          10.0                OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  45C8  7F               4.0                MOV     A,A             ;RETRY LINK
2134   45C9                                ERRB    DPEN12,DPEN13
  (1)                                       ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  45C9  CD  12  28      18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0005                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  45CC  05                              .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  45CD  00                              .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  45CE  CD  15  28      18.0            DPEN13::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  45D1  DA  B3  45      10.0                    JC      DPEN12          ;LOOP ADDRESS IF LOOP SPECIFIED
2135                                ;>EXPECTED PORT 1 WRITE PATH ENAB CLEAR & READ PATH ENAB SET
2136   45D4                                ENDTST  TST16X
  (1)                               ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  45D4                                 REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  45D4  CD  06  28      18.0                    CALL    REQST
  (2)  45D7  00                              .BYTE                   ;DATA PATTERN NUMBER
  (2)  45D8  00  00                          .WORD                   ;SYSTEM "" COUNT
  (2)  45DA  00  00                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  45DC  00                              .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  45DD  07                              .BYTE   7               ;REQUEST CODE
  (1)  45DE  3A  9A  4F      13.0                LDA     ITERA           ;GET ITERATION COUNT
  (1)  45E1  3D               4.0                DCR     A               ;DOWNCOUNT
  (1)  45E2  32  9A  4F      13.0                STA     ITERA           ;SAVE COUNT
  (1)  45E5  F2  7F  45      10.0                JP      TST16X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 17 - LOOP "W/R ENA" TO "W/R ENA" - TU PORT #2
45E8                            ST
 (1)                            ;*****************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*LOOP W/R PATH ENABLE TO W/R PATH ENABLE TEST-TU PORT #2
45E8                            SD
 (1)                            ;*****************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE WRITE PATH AND READ PATH BITS TO BE
                                ;*SET AND RESET.
45E8                            SP
 (1)                            ;*****************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #2 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT TU PORT #2
                                ;*    INPUT THE SINGLE TU PORT BIT
                                ;*    IF SINGLE PORT SET
                                ;*    :   THEN-ERROR
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    SET THE WRITE PATH ENABLE BITS FOR TU PORT #2
                                ;*    INPUT THE WRITE PATH ENABLE BITS FOR TU PORT #2
                                ;*    IF WRITE PATH ENABLE BIT SET
                                ;*    :   THEN-CONTINUE
                                ;*    :   ELSE-ERROR
                                ;*    ENDIF
                                ;*    RESET THE WRITE PATH ENABLE AND SET READ PATH ENABLE BITS FOR TU PORT #2
                                ;*    INPUT THE READ PATH ENABLE BITS FOR TU PORT #2
                                ;*    IF READ PATH ENABLE BIT RESET
                                ;*    :   THEN-CONTINUE
                                ;*    :   ELSE-ERROR
                                ;*    ENDIF
                                ;*ENDTST
45E8                            SE
 (1)                            ;*****************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*TUP1 MICRO TEST 17
                                ;*TUP1 MICRO ERROR 06
                                ;*TUP1-LOOP "W/R ENA" TO "W/R ENA" - TU PORT 2
                                ;*M8955'S
                                ;*'TUSEL0' STATUS INDICATES ONLY PORT 0 & 1 EXIST
                                ;*BIT 7 (217(8)) MUST BE =0 TO USE PORT 2
                                ;*ACTUAL = NNNN
                                ;*
                                ;*TUP1 MICRO TEST 17
```

```
                                        ;*TUP1 MICRO ERROR 07
                                        ;*TUP1-LOOP 'W/R ENA" TO "W/R ENA" - TU PORT 2
                                        ;*M8955'S
                                        ;*EXPECTED PORT 2 WRITE PATH ENAB SET & READ PATH ENAB CLEARED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*TUP1 MICRO TEST 17
                                        ;*TUP1 MICRO ERROR 10
                                        ;*TUP1-LOOP 'W/R ENA" TO "W/R ENA" - TU PORT 2
                                        ;*M8955'S
                                        ;*EXPECTED PORT 2 WRITE PATH ENAB CLEAR & READ PATH ENAB SET
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN

45E8                                    S
 (1)                                    ; ************************************************************************************

45E8                                    TEST17: TESTX   @17             ;INITIALIZE THE TEST
 (1) 45E8  3E  0F           7.0                         MVI     A,@17       ;DEFINE THE TEST NUMBER
 (1) 45EA  CD  03  28      18.0                         CALL    TSET        ;SETUP THE TEST
                                        ;%TUP1-LOOP 'W/R ENA" TO "W/R ENA"-TU PORT #2
                                        ;&M8955'S

45ED  3A  49  49          13.0  TST17X: LDA     UNITMP          ;GET THE UNIT MAP
45F0  E6  04               7.0        ANI     @4              ;TEST FOR UNIT 2
45F2  CA  6B  46          10.0        JZ      TEST20          ;NO-GO CHECK FOR TEST 20
45F5  CD  98  47          18.0        CALL    CLEAR           ;CLEAN UP THE PORTS
45F8  DB  E0              10.0        IN      INTSTA
45FA  E6  80               7.0        ANI     BIT7
45FC  C6  02               7.0        ADI     2
45FE  D3  E0              10.0        OUT     MBSEL           ;SELECT PORT 2
4600  DB  D1              10.0        IN      TUSEL0          ;GET THE SELECT STATUS WORD 0
4602                                  ROUT    ADATA           ;SAVE IN CASE OF ERROR
 (1) 4602  D3  94          10.0                OUT     ADATA       ;WRITE AC INTO ADATA
 (1) 4604  7F               4.0                MOV     A,A         ;RETRY LINK
4605  E6  80               7.0        ANI     P.SINGLE        ;SAVE THE SINGLE PORT STATUS BIT
4607  CA  0F  46          10.0        JZ      TST17P          ;JUMP IF BIT IS CLEARED (MULTI PORT)
460A                                  ERRA    TST17X,TST17P
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1) 460A  CD  0F  28      18.0                CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)     0006                         MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 460D  06                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 460E  00                         .BYTE                   ;PRINT ROUTINE NUMBER
 (1) 460F  CD  15  28      18.0        TST17P::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1) 4612  DA  ED  45      10.0                JC      TST17X          ;LOOP ADDRESS IF LOOP SPECIFIED
                                  ;>'TUSEL0' STATUS INDICATES ONLY PORT 0 & 1 EXIST
                                  ;>BIT 7 (200(8)) MUST BE =0 TO USE PORT 2
4615  3E  10               7.0  DPEN20: MVI     A,P.WPEN        ;LOAD THE DATA PATH ENABLE BITS
4617  D3  4C              10.0        OUT     PENAB           ;SET THE DATA PATH ENABLE BITS
4619  00                   4.0        NOP
461A  DB  D2              10.0        IN      TUSEL1          ;GET THE ENABLE STATUS WORD
461C  E6  3F               7.0        ANI     $3F             ;STRIP OUT THE DUAL TU PORT BIT
461E  FE  0A               7.0        CPI     P.WP2EN.2       ;SEE IF PROPER BITS SET
```

```
2222  4620  CA  30  46   10.0          JZ      DPEN21          ;JUMP IF OK
2223  4623                             ROUT    ADATA
 (1)  4623  D3  94        10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4625  7F             4.0                 MOV     A,A             ;RETRY LINK
2224  4626  3E  0A         7.0          MVI     A,P.WP2EN!2
2225  4628                             ROUT    EDATA
 (1)  4628  D3  95        10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  462A  7F             4.0                 MOV     A,A             ;RETRY LINK
2226  462B                             ERRB    DPEN20,DPEN21
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  462B  CD  12  28    18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0007                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  462E  07                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  462F  00                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4630  CD  15  28    18.0          DPEN21::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4633  DA  15  46    10.0                JC      DPEN20          ;LOOP ADDRESS IF LOOP SPECIFIED
2227                           ;>EXPECTED PORT 2 WRITE PATH ENAB SET & READ PATH ENAB CLEARED
2228  4636  3E  04         7.0  DPEN22  MVI     A,P.RPEN        ;SET READ PATH ENABLE
2229  4638  D3  4C        10.0          OUT     PENAB           ;CLEAR THE DATA PATH ENABLE BITS
2230  463A  00             4.0          NOP
2231  463B  DB  D2        10.0          IN      TUSEL1          ;GET THE ENABLE STATUS WORD
2232  463D  E6  3F         7.0          ANI     $3F             ;STRIP OUT THE DUAL TU PORT BIT
2233  463F  FE  22         7.0          CPI     P.RP2EN!2        ;SEE IF PROPER BITS CLEAR
2234  4641  CA  51  46    10.0          JZ      DPEN23          ;JUMP IF OK
2235  4644                             ROUT    ADATA
 (1)  4644  D3  94        10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4646  7F             4.0                 MOV     A,A             ;RETRY LINK
2236  4647  3E  22         7.0          MVI     A,P.RP2EN!2
2237  4649                             ROUT    EDATA
 (1)  4649  D3  95        10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  464B  7F             4.0                 MOV     A,A             ;RETRY LINK
2238  464C                             ERRB    DPEN22,DPEN23
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  464C  CD  12  28    18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0008                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  464F  08                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4650  00                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4651  CD  15  28    18.0          DPEN23::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4654  DA  36  46    10.0                JC      DPEN22          ;LOOP ADDRESS IF LOOP SPECIFIED
2239                           ;>EXPECTED PORT 2 WRITE PATH ENAB CLEAR & READ PATH ENAB SET
2240  4657                             ENDTST  TST17X
 (1)                                 ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4657                           REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4657  CD  06  28    18.0                CALL    REQST
 (2)  465A  00                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  465B  00  00                       .WORD                   ;SYSTEM "" COUNT
 (2)  465D  00  00                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SK'D
 (2)  465F  C0                           .BYTE                   ;DATA COMPARE FLAG IF  1
 (2)  4660  07                           .BYTE   7               ;REQUEST CODE
 (1)  4661  3A  9A  4F    13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4664  3D             4.0          DCR     A               ;DOWNCOUNT
 (1)  4665  32  9A  4F    13.0          STA     ITERA           ;SAVE COUNT
 (1)  4668  F2  ED  45    10.0          JP      TST17X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 20 - LOOP "W/R ENA" TO "W/R ENA" - TU PORT #3
466B                            ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*LOOP W/R PATH ENABLE TO W/R PATH ENABLE TEST-TU PORT #3
466B                            SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE WRITE PATH AND READ PATH ENABLE
                                ;*BITS TO BE SET AND RESET.
466B                            SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #3 NOT SELECTED BY USER
                                ;*     :   THEN-NEXT TEST
                                ;*     :   ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT TU PORT #3
                                ;*     INPUT THE SINGLE TU PORT BIT
                                ;*     IF SINBLE PORT SET
                                ;*     :   THEN-ERROR
                                ;*     :   ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     SET THE WRITE PATH ENABLE BITS FOR TU PORT #3
                                ;*     INPUT THE WRITE PATH ENABLE BITS FOR TU PORT #3
                                ;*     IF WRITE PATH ENABLE BIT SET
                                ;*     :   THEN-CONTINUE
                                ;*     :   ELSE-ERROR
                                ;*     ENDIF
                                ;*     RESET THE WRITE PATH ENABLE AND SET READ PATH ENABLE BITS FOR TU PORT #3
                                ;*     INPUT THE READ PATH ENABLE BIT FOR TU PORT #3
                                ;*         IF READ PATH ENABLE BITS RESET
                                ;*         :   THEN-CONTINUE
                                ;*         :   ELSE-ERROR
                                ;*         ENDIF
                                ;*ENDTST
466B                            SE
 (1)                            ;*********************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*TUP1 MICRO TEST 2?
                                ;*TUP1 MICRO ERROR 11
                                ;*TUP1-LOOP "W/R ENA" TO "W/R ENA"-TU PORT #3
                                ;*M8955'S
                                ;*'TUSEL0' STATUS INDICATES ONLY PORT 0 & 1 EXIST
                                ;*BIT 7 (220(8)) MUST BE =0 TO USE PORT 3
                                ;*
                                ;*TUP1 MICRO TEST 20
                                ;*TUP1 MICRO ERROR 12
```

```
                                        ;*TUP1-LOOP "W/R ENA" TO "W/R ENA"-TU PORT #3
                                        ;*M8955'S
                                        ;*EXPECTED PORT 3 WRITE PATH ENAB SET & READ PATH ENAB CLEARED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*TUP1 MICRO TEST 20
                                        ;*TUP1 MICRO ERROR 13
                                        ;*TUP1-LOOP "W/R ENA" TO "W/R ENA"-TU PORT #3
                                        ;*M8955'S
                                        ;*EXPECTED PORT 3 WRITE PATH ENAB CLEAR & READ PATH ENAB SET
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
466B                                    $
 (1)                                    ; ************************************************************************
                                        TEST20: TESTX   @20             ;INITIALIZE THE TEST
 (1) 466B  3E  10          7.0                   MVI     A,@20           ;DEFINE THE TEST NUMBER
 (1) 466D  CD  03  28     18.0                   CALL    TSET            ;SETUP THE TEST
                                        ;%TUP1-LOOP "W/R ENA" TO "W/R ENA"-TU PORT #3
                                        ;&M8955'S

4670  3A  49  49          13.0  TST20X: LDA     UNITMP          ;GET THE UNIT MAP
4673  E6  08                7.0         ANI     @10             ;TEST FOR UNIT 3
4675  CA  EE  46           10.0         JZ      TEST21          ;NO-GO CHECK FOR TEST 21
4678  CD  98  47           18.0         CALL    CLEAR           ;CLEAN UP THE PORTS
467B  DB  E0               10.0         IN      INTSTA
467D  E6  80                7.0         ANI     BIT7
467F  C6  03                7.0         ADI     3
4681  D3  E0               10.0         OUT     MBSEL           ;SELECT PORT 3
4683  DB  D1               10.0         IN      TUSEL0          ;GET THE SELECT STATUS WORD 0
4685                                    ROUT    ADATA           ;SAVE IN CASE OF ERROR
 (1) 4685  D3  94           10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1) 4687  7F                4.0                 MOV     A,A             ;RETRY LINK
4688  E6  80                7.0         ANI     P.SINGLE        ;SAVE THE SINGLE PORT STATUS BIT
468A  CA  92  46           10.0         JZ      TST20P          ;JUMP IF BIT IS CLEARED (MULTI PORT)
468D                                    ERR     TST20X,TST20P
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1) 468D  CD  09  28      18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0009                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 4690  09                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 4691  00                                 .BYTE
 (1) 4692  CD  15  28      18.0         TST20P::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1) 4695  DA  70  46      10.0                 JC      TST20X          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>'TUSEL0' STATUS INDICATES ONLY PORT 0 & 1 EXIST
                                        ;>BIT 7 (200(8)) MUST BE =0 TO USE PORT 3
4698  3E  10                7.0  DPEN30: MVI     A,P.WPEN        ;LOAD THE DATA PATH ENABLE BITS
469A  D3  4C               10.0         OUT     PENAB           ;SET THE DATA PATH ENABLE BITS
469C  00                    4.0         NOP
469D  DB  D2               10.0         IN      TUSEL1          ;GET THE ENABLE STATUS WORD
469F  E6  3F                7.0         ANI     $3F             ;STRIP OUT THE DUAL TU PORT BIT
46A1  FE  07                7.0         CPI     P.WP3EN!3       ;SEE IF PROPER BITS SET
46A3  CA  B3  46           10.0         JZ      DPEN31          ;JUMP IF OK
46A6                                    ROUT    ADATA
```

```
  (1)  46A6  D3  94          10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  46A8  7F               4.0                 MOV     A,A             ;RETRY LINK
2326   46A9  3E  07           7.0         MVI     A,P.WP3EN.3
2327   46AB                               ROUT    EDATA
  (1)  46AB  D3  95          10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  46AD  7F               4.0                 MOV     A,A             ;RETRY LINK
2328   46AE                               ERRB    DPEN30,DPEN31
  (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  46AE  CD  12  28      18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)        C00A                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  46B1  0A                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  46B2  00                                 .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  46B3  CD  15  28      18.0         DPEN31::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  46B6  DA  98  46      10.0                 JC      DPEN30          ;LOOP ADDRESS IF LOOP SPECIFIED
2329                           ;>EXPECTED PORT 3 WRITE PATH ENAB SET & READ PATH ENAB CLEARED
2330   46B9  3E  04           7.0  DPEN32: MVI     A,P.RPEN        ;SET READ PATH ENABLE
2331   46BB  D3  4C          10.0         OUT     PENAB           ;CLEAR THE DATA PATH ENABLE BITS
2332   46BD  00               4.0         NOP
2333   46BE  DB  D2          10.0         IN      TUSEL1          ;GET THE ENABLE STATUS WORD
2334   46C0  E6  3F           7.0         ANI     $3F             ;STRIP OUT THE DUAL TU PORT BIT
2335   46C2  FE  13           7.0         CPI     P.RP3EN'3       ;SEE IF PROPER BITS CLEAR
2336   46C4  CA  D4  46      10.0         JZ      DPEN33          ;JUMP IF OK
2337   46C7                               ROUT    ADATA
  (1)  46C7  D3  94          10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  46C9  7F               4.0                 MOV     A,A             ;RETRY LINK
2338   46CA  3E  13           7.0         MVI     A,P.RP3EN.3
2339   46CC                               ROUT    EDATA
  (1)  46CC  D3  95          10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  46CE  7F               4.0                 MOV     A,A             ;RETRY LINK
2340   46CF                               ERRB    DPEN32,DPEN33
  (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  46CF  CD  12  28      18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)        000B                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  46D2  0B                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  46D3  00                                 .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  46D4  CD  15  28      18.0         DPEN33::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  46D7  DA  B9  46      10.0                 JC      DPEN32          ;LOOP ADDRESS IF LOOP SPECIFIED
2341                           ;>EXPECTED PORT 3 WRITE PATH ENAB CLEAR & READ PATH ENAB SET
2342   46DA                               ENDTST  TST20X
  (1)                          ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  46DA                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  46DA  CD  06  28      18.0                 CALL    REQST
  (2)  46DD  00                                 .BYTE                   ;DATA PATTERN NUMBER
  (2)  46DE  00  00                             .WORD                   ;SYSTEM "" COUNT
  (2)  46E0  00  00                             .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  46E2  00                                 .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  46E3  07                                 .BYTE   7               ;REQUEST CODE
  (1)  46E4  3A  9A  4F      13.0                 LDA     ITERA           ;GET ITERATION COUNT
  (1)  46E7  3D               4.0                 DCR     A               ;DOWNCOUNT
  (1)  46E8  32  9A  4F      13.0                 STA     ITERA           ;SAVE COUNT
  (1)  46EB  F2  70  46      10.0                 JP      TST20X          ;DO TEST UNTIL TILL = 0
```

```
                                        .SBTTL  TEST 21 - LOOP "TACH" TO "TACH" - TU PORT #0
46EE                                    ST
 (1)                                    ; ***************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
                                        ;*LOOP TACH TO TACH TEST - TU PORT 0
46EE                                    SD
 (1)                                    ; ***************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS TEST CHECKS THE ABILITY OF THE TACH BIT TO SET AND RESET.
46EE                                    SP
 (1)                                    ; ***************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNTST
                                        ;*      IF TU PORT 0 NOT SELECTED BY USER
                                        ;*      :   THEN-NEXT TEST
                                        ;*      :   ELSE-CONTINUE
                                        ;*      ENDIF
                                        ;*      CALL SUBROUTINE CLEAR
                                        ;*      GET PORT 0 SELECT CODE
                                        ;*      CALL SUBROUTINE CONTR4
                                        ;*ENDTST
46EE                                    SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
                                        ;*TUP1 MICRO TEST 21
                                        ;*TUP1 MICRO ERROR 25
                                        ;*TUP1-LOOP "TACH" TO "TACH"-TU PORT #0
                                        ;*M8955'S
                                        ;*"TACH" DID NOT SET
                                        ;*
                                        ;*TUP1 MICRO TEST 21
                                        ;*TUP1 MICRO ERROR 26
                                        ;*TUP1-LOOP "TACH" TO "TACH"-TU PORT #0
                                        ;*M8955'S
                                        ;*"TACH" DID NOT RESET
46EE                                    S
 (1)                                    ; ***************************************************************************

46EE                                    TEST21: TESTX   @21             ;INITIALIZE THE TEST
 (1) 46EE  3E  11          7.0                       MVI     A,@21       ;DEFINE THE TEST NUMBER
 (1) 46F0  CD  03  28     18.0                       CALL    TSET        ;SETUP THE TEST
                                        ;%TUP1-LOOP "TACH" TO "TACH"-TU PORT #0
                                        ;&M8955'S
46F3  3A  49  49          13.0  TST21X: LDA     UNITMP          ;GET THE UNIT MAP
46F6  E6  01               7.0         ANI     @01             ;TEST FOR UNIT 0
46F8  CA  17  47          10.0         JZ      TEST22          ;NO-GO CHECK FOR TEST 22
                                                                ;
46FB  CD  98  47          18.0         CALL    CLEAR           ;CLEAN UP THE PORTS
46FE  3E  00               7.0         MVI     A,0             ;GET PORT 0 CODE
4700  CD  12  49          18.0         CALL    CONTR4          ;TEST THIS PORT
```

```
2383  4703                                    ENDTST  TST21X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4703                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4703  CD  06  28      18.0                              CALL    REQST
 (2)  4706  00                                                .BYTE                   ;DATA PATTERN NUMBER
 (2)  4707  00  00                                            .WORD                   ;SYSTEM '"' COUNT
 (2)  4709  00  00                                            .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  470B  00                                                .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  470C  07                                                .BYTE   7               ;REQUEST CODE
 (1)  470D  3A  9A  4F      13.0            LDA     ITERA               ;GET ITERATION COUNT
 (1)  4710  3D               4.0            DCR     A                   ;DOWNCOUNT
 (1)  4711  32  9A  4F      13.0            STA     ITERA               ;SAVE COUNT
 (1)  4714  F2  F3  46      10.0            JP      TST21X              ;DO TEST UNTIL TILL = 0
```

```
                                        .SBTTL  TEST 22 - LOOP "TACH" TO "TACH" - TU PORT #1
                                        $T
                                        ; ************************************************************************
                                        ;*TEST TITLE
                                        ;*---- -----
                                        ;*LOOP TACH TO TACH TEST-TU PORT #1
                                        $D
                                        ; ************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS TEST CHECKS THE ABILITY OF THE TACH BIT TO SET AND RESET.
                                        $P
                                        ; ************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNTST
                                        ;*    IF TU PORT #1 NOT SELECTED BY USER
                                        ;*    :   THEN-NEXT TEST
                                        ;*    :   ELSE-CONTINUE
                                        ;*    ENDIF
                                        ;*    CALL SUBROUTINE CLEAR
                                        ;*    GET PORT 1 SELECT CODE
                                        ;*    CALL SUBROUTINE CONTR4
                                        ;*ENDTST
                                        $E
                                        ; ************************************************************************
                                        ;*ERRORS
                                        ;*------
                                        ;*TUP1 MICRO TEST 22
                                        ;*TUP1 MICRO ERROR 25
                                        ;*TUP1-LOOP "TACH" TO "TACH"-TU PORT #1
                                        ;*M8955'S
                                        ;*"TACH" DID NOT SET
                                        ;*
                                        ;*TUP1 MICRO TEST 22
                                        ;*TUP1 MICRO ERROR 26
                                        ;*TUP1-LOOP "TACH" TO "TACH"-TU PORT #1
                                        ;*M8955'S
                                        ;*"TACH" DID NOT RESET
                                        $S
                                        ; ************************************************************************

                                        TEST22: TESTX   @22             ;INITIALIZE THE TEST
 (1)  4717  3E  12          7.0                  MVI     A,@22           ;DEFINE THE TEST NUMBER
 (1)  4719  CD  03  28     18.0                  CALL    TSET            ;SETUP THE TEST
                                        ;%TUP1-LOOP "TACH" TO "TACH"-TU PORT #1
                                        ;&M8955'S
 4717  471C  3A  49  49    13.0  TST22X: LDA     UNITMP          ;GET THE UNIT MAP
 4718  471F  E6  02          7.0         ANI     @02             ;TEST FOR UNIT 1
 4719  4721  CA  40  47     10.0         JZ      TEST23          ;NO-GO CHECK FOR TEST 23
 4720                                               :
 4721  4724  CD  98  47     18.0         CALL    CLEAR           ;CLEAN UP THE PORTS
 4722  4727  3E  01          7.0         MVI     A,1             ;GET PORT 1 SELECT CODE
 4723  4729  CD  12  49     18.0         CALL    CONTR4          ;DO THE TEST
```

```
2424  472C                                    ENDTST  TST22X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  472C                                    REQ     7                  ;FAKE CALL TO KEEP TEST ALIVE
 (2)  472C  CD  06  28        18.0                                CALL    REQST
 (2)  472F  00                                                    .BYTE                      ;DATA PATTERN NUMBER
 (2)  4730  00  00                                                .WORD                      ;SYSTEM "" COUNT
 (2)  4732  00  00                                                .WORD                      ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4734  00                                                    .BYTE                      ;DATA COMPARE FLAG IF =1
 (2)  4735  07                                                    .BYTE   7                  ;REQUEST CODE
 (1)  4736  3A  9A  4F        13.0           LDA     ITERA               ;GET ITERATION COUNT
 (1)  4739  3D                 4.0           DCR     A                   ;DOWNCOUNT
 (1)  473A  32  9A  4F        13.0           STA     ITERA               ;SAVE COUNT
 (1)  473D  F2  1C  47        10.0           JP      TST22X              ;DO TEST UNTIL TILL = 0
```

```
                                        .SBTTL  TEST 23 - LOOP "TACH" TO "TACH" - TU PORT #2
4740                                    ST
 (1)                                    ; ***************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
                                        ;*LOOP TACH TO TACH TEST-TU PORT #2
4740                                    SD
 (1)                                    ; ***************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS TEST CHECKS THE ABILITY OF THE TACH BIT TO SET AND RESET.
4740                                    SP
 (1)                                    ; ***************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNTST
                                        ;*     IF TU PORT #2 NOT SELECTED BY USER
                                        ;*     :   THEN-NEXT TEST
                                        ;*     :   ELSE-CONTINUE
                                        ;*     ENDIF
                                        ;*     CALL SUBROUTINE CLEAR
                                        ;*     GET PORT 2 SELECT CODE
                                        ;*     CALL SUBROUTINE CONTR4
                                        ;*ENDTST
4740                                    SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
                                        ;*------
                                        ;*TUP1 MICRO TEST 23
                                        ;*TUP1 MICRO ERROR 25
                                        ;*TUP1-LOOP "TACH" TO "TACH"-TU PORT #2
                                        ;*M8955'S
                                        ;*"TACH" DID NOT SET
                                        ;*
                                        ;*TUP1 MICRO TEST 23
                                        ;*TUP1 MICRO ERROR 26
                                        ;*TUP1-LOOP "TACH" TO "TACH"-TU PORT #2
                                        ;*M8955'S
                                        ;*"TACH" DID NOT RESET
4740                                    S
 (1)                                    ; ***************************************************************************

4740                                    TEST23: TESTX   @23             ;INITIALIZE THE TEST
 (1) 4740  3E  13          7.0                       MVI     A,@23       ;DEFINE THE TEST NUMBER
 (1) 4742  CD  03  28     18.0                       CALL    TSET        ;SETUP THE TEST
                                        ;%TUP1-LOOP "TACH" TO "TACH"-TU PORT #2
                                        ;&M8955'S
4745  3A  49  49           13.0  TST23X: LDA     UNITMP          ;GET THE UNIT MAP
4748  E6  04                    7.0          ANI     @04             ;TEST FOR UNIT 2
474A  CA  69  47           10.0           JZ      TEST24          ;NO-GO CHECK FOR TEST 24
                                                                  ;
474D  CD  98  47           18.0           CALL    CLEAR           ;CLEAN UP THE PORTS
4750  3E  02                    7.0          MVI     A,2             ;GET PORT 2 SELECT CODE
4752  CD  12  49           18.0           CALL    CONTR4          ;DO THE TEST
```

```
2465  4755                                  ENDTST  TST23X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4755                                  REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4755  CD   06   28        18.0                        CALL    REQST
 (2)  4758  00                              .BYTE                   ;DATA PATTERN NUMBER
 (2)  4759  00   00                         .WORD                   ;SYSTEM "" COUNT
 (2)  475B  00   00                         .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  475D  00                              .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  475E  07                              .BYTE   7               ;REQUEST CODE
 (1)  475F  3A   9A   4F        13.0        LDA     ITERA           ;GET ITERATION COUNT
 (1)  4762  3D                   4.0        DCR     A               ;DOWNCOUNT
 (1)  4763  32   9A   4F        13.0        STA     ITERA           ;SAVE COUNT
 (1)  4766  F2   45   47        10.0        JP      TST23X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 24 - LOOP ''TACH'' TO ''TACH'' - TU PORT #3
                                ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP TACH TO TACH TEST-TU PORT #3
                                SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE TACH BIT TO SET AND RESET.
                                SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #3 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    GET PORT 3 SELECT CODE
                                ;*    CALL SUBROUTINE CONTR4
                                ;*ENDTST
                                SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP1 MICRO TEST 24
                                ;*TUP1 MICRO ERROR 25
                                ;*TUP1-LOOP ''TACH'' TO ''TACH''-TU PORT #3
                                ;*M8955'S
                                ;*''TACH'' DID NOT SET
                                ;*
                                ;*TUP1 MICRO TEST 24
                                ;*TUP1 MICRO ERROR 26
                                ;*TUP1-LOOP ''TACH'' TO ''TACH''-TU PORT #3
                                ;*M8955'S
                                ;*''TACH'' DID NOT RESET
                                S
                                ; ***************************************************************************

4769                            TEST24: TESTX   @24             ;INITIALIZE THE TEST
4769  3E  14          7.0                       MVI     A,@24         ;DEFINE THE TEST NUMBER
476B  CD  03  28       18.0                      CALL    TSET          ;SETUP THE TEST
                                ;%TUP1-LOOP ''TACH'' TO ''TACH''-TU PORT #3
                                ;&M8955'S
476E  3A  49  49       13.0     TST24X: LDA     UNITMP          ;GET THE UNIT MAP
4771  E6  08             7.0             ANI     @010            ;TEST FOR UNIT 3
4773  CA  92  47        10.0             JZ      EXIT            ;EXIT THE TEST IF NOT SET
                                        ;
4776  CD  98  47        18.0             CALL    CLEAR           ;CLEAN UP THE PORTS
4779  3E  03             7.0             MVI     A,3             ;GET PORT 3 SELECT CODE
477B  CD  12  49        18.0             CALL    CONTR4          ;DO THE TEST
```

```
2506  477E                                    ENDTST  TST24X
 (1)                                ;TFST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  477E                                   REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  477E  CD  06  28      18.0                              CALL    REQST
 (2)  4781  00                                                .BYTE                   ;DATA PATTERN NUMBER
 (2)  4782  00  00                                            .WORD                   ;SYSTEM "" COUNT
 (2)  4784  00  00                                            .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4786  00                                                .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4787  07                                                .BYTE   7               ;REQUEST CODE
 (1)  4788  3A  9A  4F      13.0            LDA     ITERA           ;GET ITERATION COUNT
 (1)  478B  3D               4.0            DCR     A               ;DOWNCOUNT
 (1)  478C  32  9A  4F      13.0            STA     ITERA           ;SAVE COUNT
 (1)  478F  F2  6E  47      10.0            JP      TST24X          ;DO TEST UNTIL TILL = 0
2507
2508  4792  CD  98  47      18.0  EXIT:     CALL    CLEAR           ;LEAVE THE PORTS CLEAN
2509  4795  C3  18  28      10.0            JMP     TSTEND          ;END OF TEST
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
4798                                    $SUB
 (1)                                    ; ***************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
4798                                    $D
 (1)                                    ; ***************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
4798                                    $P
 (1)                                    ; ***************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNSUB
                                        ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*    CLEAR PORT SELECT FOR TRANSPORT
                                        ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*    CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
4798                                    $
 (1)                                    ; ***************************************************************************
4798  F5                  12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
4799  C5                  12.0          PUSH    B
479A  06  00                7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
479C  DB  E0               10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
479E  E6  80                7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
47A0  B0                    4.0          ORA     B               ;ADD IN THE SELECTED PORT #
47A1  D3  E0               10.0          OUT     MBSEL           ;RESET TO THIS PORT
47A3  3E  80                7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
47A5  D3  40               10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
47A7  AF                    4.0          XRA     A               ;CLEAR TU COMMAND A
47A8  D3  40               10.0          OUT     TCMD            ;
47AA  3E  81                7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
47AC  D3  40               10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
47AE  3E  00                7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
47B0  D3  40               10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
47B2  AF                    4.0          XRA     A
47B3  D3  44               10.0          OUT     TAMT            ;CLEAR AMTIE WORD
47B5  D3  48               10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
47B7  D3  4C               10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
47B9  04                    4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
47BA  78                    4.0          MOV     A,B
47BB  FE  04                7.0          CPI     4               ;DONE?
47BD  C2  9C  47           10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
47C0  C1                   10.0          POP     B               ;RESET B & C
47C1  F1                   10.0          POP     PSW             ;ALL DONE
47C2  C9                   10.0          RET                     ;EXIT
```

```
                                        .SBTTL  SUBROUTINE CONTR1 - LOOP COMMAND TO STATUS TESTS 1 - 4
47C3                                    $SUB
 (1)                                    ; ***************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
                                        ;*SUBROUTINE CONTROL 1 - LOOP COMMAND TO STATUS
47C3                                    $D
 (1)                                    ; ***************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS SUBROUTINE WILL ESTABLISH THE LOOP WRITE/READ CONDITION ON THE
                                        ;*SPECIFIED PORT AND TRANSFER COMMANDS (0-377(8)) CHECKING EACH ONE FOR
47C3                                    ;*PROPER TRANSFER TO THE STATUS REGISTER.
                                        $P
 (1)                                    ; ***************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNSUB
                                        ;*     CLEAR THE COMMAND DATA
                                        ;*     BGNDO
                                        ;*     :    ISSUE THE CONTENTS OF COMMAND DATA TO TU PORT COMMAND BYTE
                                        ;*     :    WAIT
                                        ;*     :    INPUT THE TU PORT STATUS BYTE
                                        ;*     :         IF STATUS=COMMAND WRITTEN
                                        ;*     :         :    THEN-CONTINUE
                                        ;*     :         :    ELSE-ERROR
                                        ;*     :         ENDIF
                                        ;*     :    CALCULATE PARITY ON THE CONTENTS OF COMMAND DATA
                                        ;*     :         IF CALCULATED PARTY=THE TU PORT STATUS PARITY
                                        ;*     :         :    THEN-CONTINUE
                                        ;*     :         :    ELSE-ERROR
                                        ;*     :         ENDIF
                                        ;*     :    INCREMENT THE TU PORT COMMAND DATA
                                        ;*     :    DO UNTIL THE TU PORT COMMAND DATA-ZERO
                                        ;*     ENDDO
                                        ;*ENDSUB
47C3                                    $
 (1)                                    ; ***************************************************************************
47C3  47                        4.0     CONTR1: MOV     B,A
47C4  DB  E0                   10.0             IN      INTSTA          ;GET THE SELECT DATA
47C6  E6  80                    7.0             ANI     BIT7            ;SAVE JUST THE MASSBUS SELECT BIT
47C8  B0                        4.0             ORA     B               ;ADD IN THE SELECTED PORT #
47C9  D3  E0                   10.0             OUT     MBSEL           ;SELECT IT
47CB  3E  40                    7.0             MVI     A,P.LCS         ;LOAD THE LOOP CMD. TO STAT. CMD BIT
47CD  D3  48                   10.0             OUT     PDIAG           ;SET LOOP MODE ON THE PORT
47CF  21  4A  49               10.0             LXI     H,DATA          ;LOAD ADDRESS OF DATA
47D2  AF                        4.0             XRA     A
47D3  77                        7.0             MOV     M,A             ;STORE IN DATA
47D4  7E                        7.0     LP01:   MOV     A,M             ;GET THE DATA
47D5  47                        4.0             MOV     B,A             ;COPY TO B
47D6  D3  40                   10.0             OUT     TCMD            ;STORE IN THE CMD. REGISTER
47D8  00                        4.0             NOP                     ;
47D9  DB  40                   10.0             IN      TSTS            ;READ THE STATUS
47DB  B8                        4.0             CMP     B               ;COMPARE ACTUAL AND EXP.
```

```
2599  47DC  CA  EB  47      10.0          JZ      LP02            ;EQUAL - CONTINUE
2600  47DF                                ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  47DF  D3  94            10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  47E1  7F                 4.0                 MOV     A,A             ;RETRY LINK
2601  47E2  7E                 7.0          MOV     A,M             ;GET EXP. DATA
2602  47E3                                ROUT    EDATA           ;STORE EXP. DATA
 (1)  47E3  D3  95            10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  47E5  7F                 4.0                 MOV     A,A             ;RETRY LINK
2603  47E6                                ERRB    LP01,LP02,0
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  47E6  CD  12  28        18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      000C                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47E9  0C                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47EA  00                                    .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  47EB  CD  15  28        18.0          LP02::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47EE  DA  D4  47        10.0                 JC      LP01            ;LOOP ADDRESS IF LOOP SPECIFIED
2604                          ;>DATA WRITTEN TO CMD. NOT = DATA READ AT STATUS
2605  47F1  7E                 7.0          MOV     A,M             ;GET THE DATA
2606  47F2                                ROUT    ADATA           ;STORE THE DATA IN CAS
 (1)  47F2  D3  94            10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  47F4  7F                 4.0                 MOV     A,A             ;RETRY LINK
2607  47F5  A7                 4.0          ANA     A               ;SET THE CONDITION CODE
2608  47F6  E2  0E  48         10.0          JPO     UNITOO          ;ODD NUMBER OF ONES
2609  47F9  DB  48             10.0          IN      PSTAT           ;ELSE - EVEN NUMBER OF ONES
2610  47FB  E6  04             7.0          ANI     P.WCSP          ;PTY. BIT SHOULD BE SET
2611  47FD  C2  05  48         10.0          JNZ     LP03            ;PTY. CORRECT - CONTINUE TEST
2612  4800                                ERRA    LP01,LP03
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4800  CD  0F  28        18.0                 CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)      000D                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4803  0D                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4804  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4805  CD  15  28        18.0          LP03::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4808  DA  D4  47        10.0                 JC      LP01            ;LOOP ADDRESS IF LOOP SPECIFIED
2613                          ;>PTY. BIT NOT = 1 FOR DATA
2614  480B  C3  20  48         10.0          JMP     UNITOX          ;CONTINUE TEST
2615                                                  ;
2616  480E  DB  48             10.0  UNITOO: IN      PSTAT           ;PTY. BIT SHOULD BE RESET
2617  4810  E6  04             7.0          ANI     P.WCSP          ;
2618  4812  CA  1A  48         10.0          JZ      LP04            ;CONTINUE IF CORRECT
2619  4815                                ERRA    LP01,LP04
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4815  CD  0F  28        18.0                 CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)      000E                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4818  0E                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4819  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  481A  CD  15  28        18.0          LP04::  CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  481D  DA  D4  47        10.0                 JC      LP01            ;LOOP ADDRESS IF LOOP SPECIFIED
2620                          ;>PTY. BIT NOT - 0 FOR DATA
2621  4820  7E                 7.0  UNITOX: MOV     A,M             ;GET DATA
2622  4821  3C                 4.0          INR     A               ;UPDATE THE DATA BYTE
2623  4822  77                 7.0          MOV     M,A             ;STORE THE UPDATED DATA
2624  4823  C2  D4  47         10.0          JNZ     LP01            ;CONTINUE UNTIL DONE
2625  4826  C9                 10.0          RET
```

```
                                        .SBTTL  SUBROUTINE CONTR2 - LOOP AMTIE TO AMTIE TESTS 1' - 14
                                        SSUB
                                        ; ************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*SUBROUTINE CONTROL 2 - LOOP AMTIE TO AMTIE
                                        SD
                                        ; ************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE WILL TRANSFER DATA (0-377(8)) ACROSS THE TU PORT AMTIE
                                        ;*LINES AS WELL AS TEST THE AMTIE PARITY BIT (BY USING THE LSB OF THE
                                        ;*DATA AS AMTIE PARITY).
                                        SP
                                        ; ************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*    BGNDO
                                        ;*    :   ISSUE THE CONTENTS OF AMTIE DATA TO TU PORT #0 AMTIE DATA REGISTER
                                        ;*    :   ISSUE THE LSB OF THE AMTIE DATA REGISTER TO TU PORT AMTIE PARITY REGISTER
                                        ;*    :   INPUT THE AMTIE DATA REGISTER
                                        ;*    :       IF AMTIE DATA READ=AMTIE DATA
                                        ;*    :         THEN-CONTINUE
                                        ;*    :         ELSE-ERROR
                                        ;*    :       ENDIF
                                        ;*    :   INPUT THE AMTIE PARITY BIT
                                        ;*    :       IF THE AMTIE PARITY READ=AMTIE PARITY WRITTEN
                                        ;*    :         THEN-CONTINUE
                                        ;*    :         ELSE-ERROR
                                        ;*    :       ENDIF
                                        ;*    :   CLEAR TU PORT LOOP W/R BIT
                                        ;*    :   INCREMENT THE AMTIE DATA
                                        ;*    :   DO UNTIL THE AMTIE DATA=ZERO
                                        ;*    ENDDO
                                        ;*ENDSUB
                                        S
                                        ; ************************************************************************
4827  47                          4.0   CONTR2: MOV     B,A             ;
4828  DB  E0                      10.0          IN      INTSTA          ;GET THE MASSBUS SELECT DATA
482A  E6  80                      7.0          ANI     BIT7            ;SAVE JUST THE SELECT BIT
482C  B0                          4.0          ORA     B               ;ADD IN THE SELECTED PORT #
482D  D3  E0                      10.0         OUT     MBSEL           ;SELECT PORT TO TEST
482F  3E  20                      7.0          MVI     A,P.LWR         ;CLEAR PORT CONTROL BYTE
4831  D3  48                      10.0         OUT     PDIAG           ;SET LOOP MODE ON THE PORT
4833  21  4A  49                  10.0         LXI     H,DATA          ;LOAD ADDRESS OF DATA
                                               :
4836  AF                          4.0          XRA     A               ;CLEAR THE ACCUMULATOR
4837  77                          7.0          MOV     M,A             ;STORE IN DATA
4838  7E                          7.0   ALP01: MOV     A,M             ;GET THE DATA
4839  47                          4.0          MOV     B,A             ;COPY TO B
483A  D3  44                      10.0         OUT     TAMT            ;STORE IN THE AMTIE REGISTER
483C  E6  01                      7.0          ANI     $01             ;MASK OFF THE LSB
483E  F6  20                      7.0          ORI     P.LWR           ;OR IN THE LOOP WRITE TO READ BIT
```

```
2671  4840  D3  48        10.0        OUT     PDIAG           ;WRITE THE AMTIE PTY.
2672  4842  00             4.0        NOP                     ;
2673  4843  DB  44        10.0        IN      TAMT            ;READ THE AMTIE REGISTER
2674                                                          ;
2675  4845  B8             4.0        CMP     B               ;COMPARE ACTUAL AND EXP.
2676  4846  CA  55  48     10.0        JZ      ALP02           ;EQUAL - CONTINUE
2677  4849                             ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  4849  D3  94        10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  484B  7F             4.0                MOV     A,A             ;RETRY LINK
2678  484C  7E             7.0        MOV     A,M             ;GET EXP. DATA
2679  484D                             ROUT    EDATA           ;STORE EXP. DATA
 (1)  484D  D3  95        10.0                OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  484F  7F             4.0                MOV     A,A             ;RETRY LINK
2680  4850                             ERRB    ALP01,ALP02,0
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4850  CD  12  28     18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      000F                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4853  0F                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4854  00                            .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  4855  CD  15  28     18.0        ALP02:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4858  DA  38  48     10.0                JC      ALP01           ;LOOP ADDRESS IF LOOP SPECIFIED
2681                        ;>DATA WRITTEN TO AMTIE NOT = DATA READ
2682  485B  78             4.0        MOV     A,B             ;GET EXP. PTY.
2683  485C  E6  01          7.0        ANI     $01             ;REMOVE UNWANTED BITS
2684  485E  47             4.0        MOV     B,A             ;RESTORE EXP. PTY.
2685  485F  DB  48        10.0        IN      PSTAT           ;GET ACTUAL PTY.
2686  4861  E6  01          7.0        ANI     $01             ;REMOVE UNWANTED BITS
2687  4863  B8             4.0        CMP     B               ;EQUAL?
2688  4864  CA  73  48     10.0        JZ      A1LP02          ;YES
2689  4867                             ROUT    ADATA           ;STORE ACTUAL PTY.
 (1)  4867  D3  94        10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4869  7F             4.0                MOV     A,A             ;RETRY LINK
2690  486A  78             4.0        MOV     A,B             ;GET EXP. PTY.
2691  486B                             ROUT    EDATA           ;STORE EXP. PTY.
 (1)  486B  D3  95        10.0                OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  486D  7F             4.0                MOV     A,A             ;RETRY LINK
2692  486E                             ERRB    ALP01,A1LP02
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  486E  CD  12  28     18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0010                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4871  10                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4872  00                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4873  CD  15  28     18.0        A1LP02::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4876  DA  38  48     10.0                JC      ALP01           ;LOOP ADDRESS IF LOOP SPECIFIED
2693                        ;>VALUE WRITTEN TO AMTIE PTY. NOT = VALUE READ
2694  4879  3E  00          7.0        MVI     A,0             ;CLEAR THE LOOP WRITE TO READ BIT
2695  487B  D3  48        10.0        OUT     PDIAG           ;CLEAR LOOP WRT TO READ BIT
2696
2697  487D  7E             7.0        MOV     A,M             ;GET DATA
2698  487E  3C             4.0        INR     A               ;UPDATE THE DATA BYTE
2699  487F  77             7.0        MOV     M,A             ;SAVE UPDATED DATA
2700  4880  C2  38  48     10.0        JNZ     ALP01           ;CONTINUE UNTIL DONE
2701  4883  C9             10.0       RET
2702
```

```
                                        .SBTTL  SUBROUTINE CONTR3 - WMC PARITY GENERATOR
                                        SSUB
                                        ; ***************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*SUBROUTINE CONTROL 3 - WMC PARITY ERROR GENERATOR
                                        SD
                                        ; ***************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE CHECKS THE WCS PARITY ERROR INTERRUPT LOGIC FOR THE
                                        ;*SPECIFIC TU PORT.
                                        SP
                                        ; ***************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*    BGNDO
                                        ;*    :   CLEAR THE WCS PARITY ERROR INTERRUPT COUNTER
                                        ;*    :   ISSUE THE CONTENTS OF COMMAND DATA TO TU PORT COMMAND BYTE
                                        ;*    :   INPUT TU PORT STATUS BYTE
                                        ;*    :   CALCULATE PARITY ON CONTENTS OF COMMAND DATA
                                        ;*    :      IF CALCULATED PARITY IS ODD
                                        ;*    :      :   THEN-CONTINUE
                                        ;*    :      :   ELSE-IF A PARITY ERROR INTERRUPT
                                        ;*    :      :         :   THEN-CONTINUE
                                        ;*    :      :         :   ELSE-ERROR
                                        ;*    :      :        ENDIF
                                        ;*    :      ENDIF
                                        ;*    :      IF CALCULATED PARITY WAS ODD
                                        ;*    :      :   THEN-IF A PARITY ERROR INTERRUPT
                                        ;*    :      :         :   THEN-ERROR
                                        ;*    :      :         :   ELSE-CONTINUE
                                        ;*    :      :        ENDIF
                                        ;*    :      :   ELSE-CONTINUE
                                        ;*    :      ENDIF
                                        ;*    :   INCREMENT THE TU PORT COMMAND DATA
                                        ;*    :   DO UNTIL THE TU PORT COMMAND DATA-ZERO
                                        ;*    ENDDO
                                        ;*ENDSUB
                                        S
                                        ; ***************************************************************************
4884  47                          4.0   CONTR3: MOV     B,A             ;SAVE THE SELECTED PORT #
4885  DB  E0                     10.0           IN      INTSTA          ;GET THE MASSBUS SELECT DATA
4887  E6  80                      7.0           ANI     BIT7            ;SAVE ONLY SELECT BIT
4889  B0                          4.0           ORA     B               ;ADD IN THE SELECTED PORT #
488A  D3  E0                     10.0           OUT     MBSEL           ;SELECT THE PORT FOR TESTING
488C  AF                         4.0           XRA     A               ;CLEAR INTERRUPT FOR PORT SELECTED
488D  D3  4C                    10.0           OUT     PENAB           ;ENABLE PORT
488F  3E  60                     7.0           MVI     A,P.LCS!P.LWR   ;LOAD LOOP CMD. TO STATUS
                                                                       ;       LOOP WRITE TO READ
4891  D3  48                    10.0           OUT     PDIAG           ;SET THE ABOVE MODE TO PORT
4893  21  4A  49                10.0           LXI     H,DATA          ;LOAD THE ADDRESS OF DATA
4896  AF                         4.0           XRA     A               ;CLEAR THE DATA BYTE
```

```
2748  4897  77                    7.0          MOV     M,A             ;STORE THE DATA
2749  4898  AF                    4.0  TUPTY0: XRA     A               ;CLEAR THE ACC.
2750  4899  32  20  4F           13.0          STA     STPCT           ;CLEAR THE WCS PTY. ERROR COUNT
2751  489C  7E                    7.0          MOV     A,M             ;GET THE DATA
2752  489D                               ROUT    ADATA                   ;STORE IN CAS
 (1)  489D  D3  94                10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  489F  7F                    4.0                  MOV     A,A             ;RETRY LINK
2753  48A0  D3  40                10.0          OUT     TCMD            ;STORE IN THE CMD. REGISTER
2754  48A2  00                    4.0          NOP
2755  48A3  00                    4.0          NOP
2756  48A4  DB  40                10.0          IN      TSTS            ;ACCESS THE TAPE UNIT STATUS BYTE
2757  48A6  7E                    7.0          MOV     A,M             ;GET THE DATA BYTE ISSUED
2758  48A7  A7                    4.0          ANA     A               ;SET THE CONDITION CODE
2759  48A8  E2  DA  48            10.0          JPO     TUPAR0          ;CONTINUE-ODD NUMBER OF ONES
2760  48AB  DB  48                10.0          IN      PSTAT           ;GET THE PORT STATUS WORD
2761  48AD  E6  80                 7.0          ANI     BIT7            ;SAVE STATUS PARITY ERROR BIT
2762  48AF  C2  B7  48            10.0          JNZ     TUCN10          ;JUMP IF OK
2763  48B2                               ERRA    TUPTY0,TUCN10
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  48B2  CD  0F  28            18.0                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0011                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48B5  11                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48B6  00                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  48B7  CD  15  28            18.0          TUCN10::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48BA  DA  98  48            10.0                  JC      TUPTY0          ;LOOP ADDRESS IF LOOP SPECIFIED
2764                            ;>STATUS PARITY ERROR BIT NOT SET ON DATA BYTE (ACTUAL)
2765  48BD  3E  80                 7.0          MVI     A,P.INTEN       ;ENABLE FOR INTERRUPT (ALSO CLEAR ERROR BIT)
2766  48BF  D3  4C                10.0          OUT     PENAB
2767  48C1  DB  40                10.0          IN      TSTS            ;CAUSE BIT TO SE' AGAIN
2768  48C3  00                    4.0          NOP                     ;DELAY FOR INTERRUPT TO HAPPEN
2769  48C4  00                    4.0          NOP
2770  48C5  3A  20  4F            13.0          LDA     STPCT           ;ELSE-EVEN NUMBER OF ONES
2771  48C8  A7                    4.0          ANA     A               ;THERE SHOULD HAVE BEEN
2772  48C9  C2  D1  48            10.0          JNZ     TUCN11          ;AN INTER.
2773  48CC                               ERRA    TUPTY0,TUCN11
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  48CC  CD  0F  28            18.0                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0012                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48CF  12                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48D0  00                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  48D1  CD  15  28            18.0          TUCN11::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48D4  DA  98  48            10.0                  JC      TUPTY0          ;LOOP ADDRESS IF LOOP SPECIFIED
2774                            ;>NO INTERRUPT WHEN ONE EXPECTED ON DATA BYTE (ACTUAL)
2775  48D7  C3  06  49            10.0          JMP     TUPR10
2776  48DA  DB  48                10.0  TUPAR0: IN      PSTAT           ;GET PORT STATUS WORD
2777  48DC  E6  80                 7.0          ANI     BIT7
2778  48DE  CA  E6  48            10.0          JZ      TUCN20          ;JUMP IF OK
```

```
                                        ERRA    TUPTY0,TUCN20
                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
48E1  CD 0F 28          18.0                    CALL    ERLPA           ;PROCESS ERROR - DO 2.3
      0013                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
48F4  13                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
48E5  00                                        .BYTE                   ;PRINT ROUTINE NUMBER
48E6  CD 15 28          18.0            TUCN20::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
48E9  DA 98 48          10.0                    JC      TUPTY0          ;LOOP ADDRESS IF LOOP SPECIFIED
                        ;>STATUS PARITY ERROR BIT NOT EXPECTED FOR DATA BYTE (ACTUAL)
48EC  3E 80              7.0            MVI     A,P.INTEN       ;RE-ENABLE
48EE  D3 4C             10.0            OUT     PENAB
48F0  DB 40             10.0            IN      TSTS            ;CAUSE BIT TO SET (IF IT DOES, ITS AN ERROR)
48F2  00                 4.0            NOP                     ;SMALL DELAY
48F3  00                 4.0            NOP
48F4  3A 20 4F          13.0            LDA     STPCT           ;GET THE INTER. COUNT
48F7  A7                 4.0            ANA     A               ;SET CONDITION BITS
48F8  CA 00 49          10.0            JZ      TUCN21          ;CONTINUE IF NO INTER.
48FB                                    ERRA    TUPTY0,TUCN21
                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
48FB  CD 0F 28          18.0                    CALL    ERLPA           ;PROCESS ERROR - DO 2.3
      0014                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
48FE  14                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
48FF  00                                        .BYTE                   ;PRINT ROUTINE NUMBER
4900  CD 15 28          18.0            TUCN21::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
4903  DA 98 48          10.0                    JC      TUPTY0          ;LOOP ADDRESS IF LOOP SPECIFIED
                        ;>INTER. WHEN NONE EXP.
4906  7E                 7.0    TUPRI0: MOV     A,M             ;GET THE DATA BYTE
4907  3C                 4.0            INR     A               ;UPDATE THE DATA BYTE
4908  77                 7.0            MOV     M,A             ;STORE THE UPDATED DATA
4909  AF                 4.0            XRA     A               ;CLEAR INTERRUPT ENABLE
490A  D3 4C             10.0            OUT     PENAB
490C  7E                 7.0            MOV     A,M             ;GET DATA BYTE AGAIN
490D  A7                 4.0            ANA     A               ;SET CONDITIONS
490E  C2 98 48          10.0            JNZ     TUPTY0          ;CONTINUE UNTIL DONE
4911  C9                10.0            RET
```

```
                                .SBTTL  SUBROUTINE CONTR4 - LOOP TACH TO TACH
4912                            SSUB
 (1)                            ; ***************************************************************************
 (1)                            ;*SUBROUTINE TITLE
 (1)                            ;*---------- -----
                                ;*SUBROUTINE CONTROL 4 - LOOP TACH TO TACH
4912                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS SUBROUTINE CHECKS THE ABILITY OF THE TACH BIT FOR THE SPECIFIED
                                ;*TU PORT TO SET AND RESET.
4912                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNSUB
                                ;*     SET THE TACH AND LOOP W/R BITS FOR SPECIFIED TU PORT
                                ;*     INPUT THE TACH BIT FOR THE TU PORT
                                ;*          IF TACH BIT SET
                                ;*          :   THEN-CONTINUE
                                ;*          :   ELSE-ERROR
                                ;*          ENDIF
                                ;*     CLEAR THE TACH BIT AND SET THE LOOP W/R BITS FOR SPECIFIED TU PORT
                                ;*     INPUT THE TACH BIT FOR THE TU PORT
                                ;*          IF TACH BIT RESET
                                ;*          :   THEN-CONTINUE
                                ;*          :   ELSE-ERROR
                                ;*          ENDIF
                                ;*ENDSUB
4912                            S
 (1)                            ; ***************************************************************************
                                ;
4912  47              4.0   CONTR4: MOV     B,A             ;SAVE THE SELECTED PORT #
4913  DB  E0           10.0         IN      INTSTA          ;GET MB SELECT DATA
4915  E6  80            7.0         ANI     BIT7
4917  B0                4.0         ORA     B               ;ADD IN THE PORT #
4918  D3  E0           10.0         OUT     MBSEL           ;SELECT THE PORT
                                
491A  3E  28            7.0   TACH00: MVI     A,P.TACH!P.LWR  ;LOAD THE TACH BIT AND SET LOOP WRITE TO READ
491C  D3  48           10.0         OUT     PDIAG           ;SET THE TACH BIT
491E  00                4.0         NOP                     ;
491F  DB  48           10.0         IN      PSTAT           ;GET THE PORT STATUS
4921  E6  08            7.0         ANI     P.TACH          ;CHECK THE TACH BIT
4923  C2  28  49       10.0         JNZ     TACH01          ;CONTINUE IF SET
4926                               ERR     TACH00,TACH01
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4926  CD  09  28  18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0015                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4929  15                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  492A  00                                     .BYTE
 (1)  492B  CD  15  28  18.0                TACH01::             CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  492E  DA  1A  49  10.0                                     JC      TACH00          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>"TACH" DID NOT SET
4931  3E  20            7.0   TACH02: MVI     A,P.LWR         ;CLEAR THE TACH BIT
```

```
2839  4933  D3  48          10.0        OUT     PDIAG           ;CLEAR THE TACH BIT
2840  4935  00               4.0        NOP
2841  4936  DB  48          10.0        IN      PSTAT           ;GET THE PORT STATUS
2842  4938  E6  08           7.0        ANI     P.TACH          ;CHECK FOR TACH
2843  493A  CA  42  49       10.0        JZ      TACH03          ;CONTINUE IF ZERO
2844  493D                              ERR     TACH02,TACH03
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  493D  CD  09  28       18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0016                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4940  16                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4941  00                          .BYTE
 (1)  4942  CD  15  28       18.0    TACH03::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4945  DA  31  49       10.0                    JC      TACH02          ;LOOP ADDRESS IF LOOP SPECIFIED
2845                        ;>"TACH" DID NOT RESET
2846  4948  C9               10.0        RET
2847
2848
2849
2850                        .SBTTL  PROGRAM VARIABLES
2851
2852  4949  00              UNITMP: .BYTE   0               ;UNIT MAP
2853  494A  00              DATA:   .BYTE   0               ;DATA PATTERN
2854       0000             .END
```

```
A      =%0007          ADATA  = 0094          ALP01   4838          ALP02   4855  G
AMTIEP= 0001           AMTIE7= 0002           ARAIDF= 0098          ASAVE   4F9B
ATTCD   4F97           AXNUM   4F91           A1LP02  4873  G       B      =%0000
BADST  = 0090          BIT0   = 0001          BIT1   = 0002          BIT15  = 8000
BIT2   = 0004          BIT3   = 0008          BIT4   = 0010          BIT5   = 0020
BIT6   = 0040          BIT7   = 0080          BIT8   = 0100          BIT9   = 0200
BRKPBC= 4F0A           BRKRAM= 4F10           BRKSTR= 4E60          BRKXCT= 4F00
BSAVE   4F9C           BYTCNT= 00D4           BYTEH   4F24          BYTEL   4F23
C      =%0001          CASCT   4F21           CASCTL= 00A0          CASSTA= 00A0
CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1          CBYTH  = 008B
CBYTL  = 008A          CDG1H  = 0087          CDG1L  = 0086          CDG2H  = 0093
CDG2L  = 0092          CDG3H  = 0095          CDG3L  = 0094          CDVTH  = 008D
CDVTL  = 008C          CHPTIE= 0028           CHOTIE= 0020          CH1TIE= 0021
CH2TIE= 0022           CH3TIE= 0023           CH4TIE= 0024          CH5TIE= 0025
CH6TIE= 0026           CH7TIE= 0027           CKLOP  = 2815          CLEAR   4798
CLKCTL= 00F0           CLOCK   4F26           CLRLP   479C          CMC0H  = 0099
CMC0L  = 0098          CMC1H  = 009B          CMC1L  = 009A          CMC2H  = 009D
CMC2L  = 009C          CMC3H  = 009F          CMC3L  = 009E          CMINH  = 0097
CMINL  = 0096          CNTCTL= 00D7           CONTR1  47C3          CONTR2  4827
CONTR3  4884           CONTR4  4912           CRCWRD= 0018          CSAVE   4F9D
CSRLH  = 0091          CSRLL  = 0090          CTCH   = 0085          CTCL   = 0084
CTSTH  = 008F          CTSTL  = 008E          CXCTH  = 0081          CXCTL  = 0080
CXINH  = 0083          CXINL  = 0082          C.     = 0001          C.AVAI= 0080
C.DP   = 0008          C.DSE  = 0010          C.DTU  = 0003          C.DVA  = 0008
C.FAIL= 00FC           C.FMT  = 0070          C.FNCT= 003E          C.GO   = 0001
C.INTC= 00FE           C.MAIN= 0020           C.NSA  = 0080          C.RCT  = 00FC
C.SER  = 0080          C.SHR  = 0040          C.SKPC= 000F          C.TAPE= 0040
C.WCS  = 0002          D      =%0002          DATA    494A          DATACT= 00D0
DBUS    4F28           DBUSCT= 00C0           DBUSST= 00C0          DDRA   = 00D8
DDRAIN= 0010           DDRB   = 00D9          DDRBIN= 0002          DDRC   = 00DA
DDRCIN= 0001           DDRCO  = 0088          DDRCTL= 00DB          DIAFLG  4F22
DIAGPG= 4300           DIAGRM= 4F90           DIARD  = 000B          DONE1  = 0045
DONINT= 0010           DPEN00  4524           DPEN01  453C  G       DPEN02  4545
DPEN03  4560  G        DPEN10  4592           DPEN11  45AD  G       DPEN12  45B3
DPEN13  45CE  G        DPEN20  4615           DPEN21  4630  G       DPEN22  4636
DPEN23  4651  G        DPEN30  4698           DPEN31  46B3  G       DPEN32  46B9
DPEN33  46D4  G        DSAVE   4F9E           DSE    = 0006          DUMMY   431E  G
D.ATH0= 0001           D.ATH1= 0002           D.EOTD= 0010          D.LAGC= 0020
D.NOTW= 0040           D.NTHR= 0004           D.TACH= 0008          D.WR4  = 0080
E      =%0003          ECCBAD= 0042           ECCCOR= 0019          ECCOK  = 0041
ECCSTA= 001A           ECCTST= 000E           EDATA  = 0095          EOTCLR= 0003
ERFLG   4F93           ERLP   = 2809          ERLPA  = 280F          ERLPB  = 2812
ERLPE  = 280C          ERNUM   4F90           ERRCNT= 00D6          ESAVE   4F9F
EXIT    4792           E.ACRC= 0010           E.AMT  = 0020          E.CDP  = 0080
E.CRC  = 0080          E.PNTR= 0008           E.RPE  = 0040          E.STEC= 0001
E.TTEC= 0002           E.UNC  = 0004          FIFORD= 006A          FORMAT  4F25
FWDTST= 0061           GCRID  = 0089          GCRSET= 0002          GOODTM= 0092
H      =%0004          HLSAVE  4FA0           IE     = 0008          INTSTA= 00E0
ITERA   4F9A           I.PWR  = 0020          I.RMPE= 0040          I5.5   = 0010
I6.5   = 0020          I7.5   = 0040          KCALL  = 005F          KCLR   = 007B
KDEP   = 003F          KENAB  = 0078          KEXAM  = 003E          KEYBRD= 00C8
KEY1   = 0078          KEY10  = 006D          KEY11  = 006E          KEY12  = 006F
KEY13  = 005C          KEY14  = 005D          KEY15  = 005F          KEY16  = 005F
KEY17  = 003C          KEY18  = 003D          KEY19  = 003E          KEY2   = 0079
```

```
KEY20 = 003F            KEY3  = 007A            KEY4  = 007B            KEY5  = 0074
KEY6  = 0075            KEY7  = 0076            KEY8  = 0077            KEY9  = 006C
KINTA = 006F            KLDAD = 003D            KN0   = 003C            KN1   = 005C
KN2   = 005D            KN3   = 005E            KN4   = 006C            KN5   = 006D
KN6   = 006E            KN7   = 0074            KN8   = 0075            KN9   = 0076
KU2   = 0079            KU3   = 007A            KU8   = 0077            L     =%0005
LBLANK= 000F            LCE   = 0008            LCH   = 000C            LCL   = 000D
LCP   = 000E            LC0   = 0000            LC1   = 0001            LC2   = 0002
LC3   = 0003            LC4   = 0004            LC5   = 0005            LC6   = 0006
LC7   = 0007            LC8   = 0008            LC9   = 0009            LDLEDA= 00CA
LDLEDB= 00CB            LDLEDC= 00CC            LDLEDD= 00CD            LDLEDE= 00CE
LDLEDF= 00CF            LKDIAG= 2800            LKKBD = 004C            LKKFY = 0049
LKLWMG= 0058            LKLWMP= 0055            LKLWPG= 0052            LKLWPP= 004F
LKMOD7= 0046            LKOPR = 0046            LPFLG   4F94            LPNUM   4F92
LP01    47D4            LP02    47EB  G         LP03    4805  G         LP04    481A  G
M     =%0006            MBSEL = 00E0            MB.A  = 0008            MB.B  = 0004
MEMTOP= 4FFF            MINUS = 000A            MM    = 8000            MSE   = 0008
MSGN  = 0016            MTACLR= 0000            MT.ARA= 0020            MT.CPE= 0080
MT.DSE= 0001            MT.FWD= 0040            MT.INH= 0008            MT.LWR= 0004
MT.MOT= 0002            MT.NWT= 0080            MT.PEC= 0040            MT.PSB= 0004
MT.PSO= 0001            MT.PS1= 0002            MT.REV= 0020            MT.WRT= 0010
MT.Z  = 0008            M.ATA = 0080            M.CAPE= 0020            M.CONT= 0080
M.DEM = 0020            M.EBL = 0010            M.EXC = 0008            M.FAIL= 0008
M.ILR = 0010            M.INIT= 0010            M.OCC = 0020            M.ONLI= 0001
M.PE  = 0040            M.PORT= 0080            M.RDEN= 0002            M.RDPE= 0008
M.RUN = 0004            M.SCLK= 0001            M.TRA = 0040            M.UNIT= 0007
M.WCLK= 0040            M.WCLN= 0080            M.WREN= 0080            M5.5  = 0001
M6.5  = 0002            M7.5  = 0004            NOTCAP= 0088            OKAY  = 00FF
OPRRAM= 4300            OPSTRT= 0058            OPVER = 0040            PADCNT= 00D5
PADCRC= 0080            PDIAG = 0048            PEID  = 008A            PENAB = 004C
PESET = 0001            PL    = 00B1            PRDD  = 004C            PRENF = 009C
PS    = 00B2            PSTAT = 0048            PSW   =%0009            P.AMTP= 0001
P.BCIC= 0040            P.CMDP= 0020            P.INTE= 0080            P.LCS = 0040
P.LWR = 0020            P.RDP = 0002            P.RPEN= 0004            P.RPST= 0002
P.RPOE= 0020            P.RP1E= 0010            P.RP2E= 0020            P.RP3E= 0010
P.SING= 0080            P.STAT= 0002            P.STPE= 0080            P.TACH= 0C08
P.TUPR= 0010            P.WCSP= 0004            P.WDS = 0040            P.WFLP= 0001
P.WPEN= 0010            P.WPOE= 0008            P.WP1E= 0004            P.WP2E= 0008
P.WP3E= 0004            P.5VOK= 0002            QUE   = 281B            QUEM  = 281E
RAMT  = 0010            RARA  = 0006            RARAI = 0004            RCHBD0= 0048
RCHBD1= 0047            RCHOK = 0046            RCHTST= 000C            RCLRT = 000D
RCMD  = 000B            RCMLP = 0003            RCONT = 0080            RDATA = 0017
RDCLK = 0010            RDON  = 0011            READG = 0007            REND  = 0014
REQST = 2806            RESCHR= 00D1            REVTST= 0064            REWIND= 0004
RFIFOL= 0008            RGCLK = 0002            RGCRI = 0003            RIBG  = 0001
RILL  = 0012            RINST = 000C            RMCTST= 0008            RMK2  = 0013
RNOP  = 0000            RPATH = 0001            RPBAD = 0044            RPCHI = 0001
RPCLK = 0003            RPCTL = 0009            RPFI  = 0002            RPFAIL= 0000
RPF1  = 009D            RPF2  = 009E            RPOK  = 0043            RPOSTN= 0016
RPSTA = 0015            RRCMT = 000A            RSTAT = 0002            RTIEB = 000A
RTIER = 0030            RTM   = 0005            RUNKI = 0009            RUPTST= 005E
RWDUNL= 0005            R.AMT = 0001            R.BOP = 0008            R.DATA= 0040
R.DON = 0002            R.DRDY= 0010            R.END = 0010            R.ILL = 0004
R.JVOK= 0004            R.MK2 = 0008            R.PLOD= 0008            R.PLOO= 0010
```

```
R.PLO1= 0020          R.POST= 0020          R.STNM= 0002          R.STOP= 0004
R.STPC= 0001          R.TBJN= 0080          R.TSTD= 0040          R.VOK = 0080
R00H  = 0081          R00L  = 0080          R01H  = 0083          R01L  = 0082
R02H  = 0085          R02L  = 0084          R03H  = 0087          R03L  = 0086
R04H  = 0089          R04L  = 0088          R05H  = 008B          R05L  = 008A
R06H  = 008D          R06L  = 008C          R07H  = 008F          R07L  = 008E
R10H  = 0091          R10L  = 0090          R11H  = 0093          R11L  = 0092
R12H  = 0095          R12L  = 0094          R13H  = 0097          R13L  = 0096
R14H  = 0099          R14L  = 0098          R15H  = 009B          R15L  = 009A
R16H  = 009D          R16L  = 009C          R17H  = 009F          R17L  = 009E
R7.5  = 0010          SELCLR= 0000          SETATA= 00A1          SID   = 0080
SOD   = 0080          SOE   = 0040          SP    =%0008          SSCLK = 0040
SSTEP = 0005          STACK = 4FFF          STATRM= 4F20          STPCT   4F20
STRSP = 5000          TACH00  491A          TACH01  492B  G       TACH02  4931
TACH03  4942  G       TADR00= 0080          TADR01= 0081          TADR02= 0082
TADR03= 0083          TADR04= 0084          TADR05= 0085          TADR06= 0086
TADR07= 0087          TADR10= 0088          TADR11= 0089          TADR12= 008A
TADR13= 008B          TAMT  = 0044          TASEL = 0080          TCMD  = 0040
TC.FWD= 0040          TC.INH= 0008          TC.LWR= 0004          TC.REV= 0020
TC.WRT= 0010          TEMP    4F99          TEST1   4300          TEST10  443F
TEST11  4468          TEST12  4491          TEST13  44BA          TEST14  44E3
TEST15  450C          TEST16  457A          TEST17  45E8          TEST2   434A
TEST20  466B          TEST21  46EE          TEST22  4717          TEST23  4740
TEST24  4769          TEST3   4373          TEST4   439C          TEST5   43C5
TEST6   43ED          TEST7   4416          TMF   = 0099          TMRDY = 0040
TRKENA= 0CD2          TSET  = 2803          TSTEND= 2818          TSTS  = 0040
TST01X  4327          TST02X  434F          TST03X  4378          TST04X  43A1
TST05X  43CA          TST06X  43F2          TST07X  441B          TST10X  4444
TST11X  446D          TST12X  4496          TST13X  44BF          TST14X  44E8
TST15X  4511          TST16X  457F          TST17P  460F  G       TST17X  45ED
TST20P  4692  G       TST20X  4670          TST21X  46F3          TST22X  471C
TST23X  4745          TST24X  476E          TUCN10  48B7  G       TUCN11  48D1  G
TUCN20  48E6  G       TUCN21  4900  G       TUPAR0  48DA          TUPR10  4906
TUPTY0  4898          TUSEL0= 00D1          TUSEL1= 00D2          TU78  = 0010
T.ATH0= 0001          T.ATH1= 0002          T.BOT = 0004          T.EOT = 0002
T.FPT = 0001          T.NTHR= 0004          T.ONL = 0020          T.PES = 0008
T.PSBJ= 0020          T.PS0J= 0008          T.PS1J= 0010          T.RDY = 0080
T.RDY0= 0040          T.RWD = 0010          T.SCLK= 0002          UIBG  = 00A1
UNITMP  4949          UNIT0C  480E          UNIT0X  4820          VALFC   4F98
VALTB   4F95          VELTST= 005B          WDR.P = 0010          WMCCTL= 00D3
WMCERR= 00DA          WMCSTA= 00D0          WRTCLK= 0000          WRTDAT= 00D3
W.ACRC= 0004          W.CRC = 0008          W.DIAG= 0002          W.DONN= 0040
W.ECC = 0010          W.ENAB= 0080          W.ERR = 0020          W.FMT = 0070
W.GCR = 0010          W.LEFT= 0004          W.ONES= 0020          W.RESI= 0002
W.REV = 0004          W.ROME= 0010          W.RST = 0001          W.SKIP= 000F
W.WRIT= 0008          W.XFER= 0020          X     =%000A          X.DONN= 0080
X.ENAB= 0040          X.PEPE= 0002          X.ROME= 0001          X.WCLK= 0001
Y     =%000B          .     = 494B
```

ERRORS DETECTED: 0

*TUP1.A78/PTP,TUP1-NLIST,PARAM,MACRO,LIST,TUP1

RUN-TIME: 5 8 0 SECONDS

3 "MEMORY MAP" EXTENDED ADDRESS BIT
19 READ MICRO CONTROLLER CONTROL & STATUS WORDS
139 TRANSPORT/PORT COMMAND + STATUS WORDS
330 COMMON ADDRESS SPACE REGISTERS
444 MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485 MASSBUS CONTROL WORD DEFINITIONS
557 WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709 KEYBOARD/DISPLAY CONTROL + STATUS WORD
797 MISCELLANEOUS VARIABLE DEFINITIONS
801 RAM ADDRESS DEFINITIONS
819 LINKAGE TABLE ADDRESSES
842 BIT DEFINITIONS
856 RIM AND SIM INSTRUCTION BIT DEFINITIONS
877 INTERRUPTABLE INSTRUCTION MACROS
924 DIAGNOSTIC VARIABLES
945 COMMON VARIABLE STORAGE
959 MACRO CALL DEFINITIONS
1047 TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101 8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1335 TEST 01 - LOOP COMMAND TO STATUS - MTA #0
1458 TEST 02 - LOOP COMMAND TO STATUS - MTA #1
1567 TEST 03 - LOOP COMMAND TO STATUS - MTA #2
1677 TEST 04 - LOOP COMMAND TO STATUS - MTA #3
1786 TEST 05 - LOOP AMTIE TO AMTIE - MTA #0
1900 TEST 06 - LOOP AMTIE TO AMTIE - MTA #1
2014 TEST 07 - LOOP AMTIE TO AMTIE - MTA #2
2128 TEST 10 - LOOP AMTIE TO AMTIE - MTA #3
2242 TEST 11 - LOOP DATA WRITE/READ - MTA #0
2349 TEST 12 - LOOP DATA WRITE/READ - MTA #1
2456 TEST 13 - LOOP DATA WRITE/READ - MTA #2
2563 TEST 14 - LOOP DATA WRITE/READ - MTA #3
2674 SUBROUTINE CLEAR ALL TU PORTS
2719 PROGRAM VARIABLES

```
                ;*************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;               "BYTE/SCLK COUNT NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               MM = DATA FORMAT FROM CAS REGISTER 2
                ;               NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;               BYTE-SCLK COUNT = LLL
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               LLL = AS ABOVE
                ;               MM  = AS ABOVE
                ;               NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;               TRANSITION COUNT = LLL
                ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;               EXPECTED 18 BITS =      E EEEEFE
                ;               ACTUAL 18 BITS =        A AAAAAA
                ;
                ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;               OF CAS REG 15 LOW BYTE.
                ;
                ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;               17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;               AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;               "SUBGROUP NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;*************************************************************************
```

```
1016                    ; ************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUES  A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ; ************************************************************************
1035                    ; ************************************************************************
1036                    ;DATA PATTERN CODES
1037
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ; ************************************************************************
1329    4300                    .-      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
1332                                .TITLE  MTA1 - MAG TAPE ADAPTER TEST PART #1
1333                                ;ID     MTA1-MAG TAPE ADAPTER CONTROLLER PART #1
1334
1335                                .SBTTL  TEST 01 - LOOP COMMAND TO STATUS - MTA #0
1336  4300                          ST
 (1)                                ; **********************************************************************
 (1)                                ;*TEST TITLE
 (1)                                ;*---- -----
1337                                ;*LOOP COMMAND TO STATUS TEST-MTA #0
1338  4300                          SD
 (1)                                ; **********************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
1339                                ;*THIS TEST CHECKS THE ABILITY OF THE COMMAND BITS TO SET AND RESET, FOR
1340                                ;*MTA #0.
1341  4300                          SP
 (1)                                ; **********************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
1342                                ;*BGNTST
1343                                ;*     IF TU PORT #0 NOT SELECTED BY USER
1344                                ;*     :   THEN-NEXT TEST
1345                                ;*     :   ELSE-CONTINUE
1346                                ;*     ENDIF
1347                                ;*     CALL SUBROUTINE CLEAR
1348                                ;*     SELECT PORT 0
1349                                ;*     IF TU PRESENT FOR TU PORT 0 = 0
1350                                ;*     :   THEN-ERROR 02
1351                                ;*     :   ELSE-CONTINUE
1352                                ;*     ENDIF
1353                                ;*     SET THE FWD, REV, WRT, WRT INH AND LWR COMMAND BITS FOR MTA #0
1354                                ;*     INPUT THE STATUS BYTE A FOR MTA #0
1355                                ;*     IF FWD, REV, WRT, WRT INH AND LWR BITS SET
1356                                ;*     :   THEN-CONTINUE
1357                                ;*     :   ELSE-ERROR 03
1358                                ;*     ENDIF
1359                                ;*     RESET THE COMMAND BITS FOR MTA #0
1360                                ;*     INPUT THE STATUS BYTE A FOR MTA #0
1361                                ;*     MASK OUT FWD, REV, WRT, WRT INH AND LWR BITS
1362                                ;*     IF ALL BITS OF RESULT RESET
1363                                ;*     :   THEN-CONTINUE
1364                                ;*     :   ELSE-ERROR 04
1365                                ;*     ENDIF
1366                                ;*ENDTST
1367  4300                          SE
 (1)                                ; **********************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1368                                ;*MTA1 MICRO TEST 01
1369                                ;*MTA1 MICRO ERROR 01
1370                                ;*MTA1-LOOP COMMAND TO STATUS-MTA #0
1371                                ;*M8954, M8955'S
1372                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1373                                ;*FATAL ERROR - MICRO MODULE ABORTED
```

```
                                        ;*
                                        ;*MTA1 MICRO TEST 01
                                        ;*MTA1 MICRO ERROR 02
                                        ;*MTA1-LOOP COMMAND TO STATUS-MTA #0
                                        ;*M8954, M8955'S
                                        ;*PORT 0 'TU PRES' BIT NOT SET
                                        ;*
                                        ;*MTA1 MICRO TEST 01
                                        ;*MTA1 MICRO ERROR 03
                                        ;*MTA1-LOOP COMMAND TO STATUS-MTA #0
                                        ;*M8954, M8955'S
                                        ;*COMMAND BYTE WRITTEN NOT = STATUS BYTE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA1 MICRO TEST 01
                                        ;*MTA1 MICRO ERROR 04
                                        ;*MTA1-LOOP COMMAND TO STATUS-MTA #0
                                        ;*M8954, M8955'S
                                        ;*STATUS BYTE NOT = ZERO WHEN COMMAND - 0
4300                                    S
 (1)                                    ; ************************************************************************
                                        TEST01: TESTX   @01             ;INITIALIZE THE TEST
 (1) 4300  3E  01          7.0                 MVI     A,@01           ;DEFINE THE TEST NUMBER
 (1) 4302  CD  03  28     18.0                 CALL    TSET            ;SETUP THE TEST
                                        ;%MTA1-LOOP COMMAND TO STATUS-MTA #0
                                        ;&M8954, M8955'S
4305                                            REQ     @7,0,0,0,0
 (1) 4305  CD  06  28     18.0                 CALL    REQST
 (1) 4308  00                                  .BYTE   0               ;DATA PATTERN NUMBER
 (1) 4309  00  00                              .WORD   0               ;SYSTEM "0" COUNT
 (1) 430B  00  00                              .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1) 430D  00                                  .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1) 430E  07                                  .BYTE   @7              ;REQUEST CODE
430F                                    RIN     R12L                    ;GET THE UNITS DESIRED
 (1) 430F  DB  94         10.0                 IN      R12L                    ;READ R12L INTO AC
 (1) 4311  7F              4.0                 MOV     A,A                     ;RETRY LINK
4312  32  CD  49          13.0          STA     UNITMP                  ;STORE IN MEMORY
4315  A7                   4.0          ANA     A                       ;SET THE CONDITION CODE
4316  C2  27  43          10.0          JNZ     TST01X                  ;GO TEST THE SPECIFIED UNITS
                                                                       ;NO UNITS SPECIFIED
4319                                    ERR     EXIT,DUMMY
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1) 4319  CD  09  28     18.0                 CALL    ERLP                    ;PROCESS ERROR - DO 2.3
 (1)       0001                               MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 431C  01                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 431D  00                                 .BYTE
 (1) 431E  CD  15  28     18.0          DUMMY:: CALL    CKLOP                   ;CHECK LOOP FUNCTION - DO 2.3
 (1) 4321  DA  9C  49     10.0                 JC      EXIT                    ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                ;>FATAL ERROR - MICRO MODULE ABORTED
4324  C3  9C  49          10.0          JMP     EXIT                    ;EXIT THE TEST
```

```
1410  4327  3A CD 49      13.0  TST01X: LDA     UNITMP          ;GET THE UNIT MAP
1411  432A  E6 01           7.0          ANI     @01             ;TEST FOR UNIT 0
1412  432C  CA A5 43        10.0         JZ      TEST02          ;NO-GO CHECK FOR TEST 02
1413                                     ;
1414  432F  CD A2 49        18.0         CALL    CLEAR           ;CLEAN UP THE PORTS
1415  4332  DB E0          10.0         IN      INTSTA          ;GET MB SELECT STATUS
1416  4334  E6 80           7.0          ANI     BIT7            ;
1417  4336  D3 E0          10.0         OUT     MBSEL           ;SELECT PORT 0
1418  4338  DB 48          10.0         IN      PSTAT           ;GET TAPE STATUS TO SEE IF A UNIT IS PRESENT
1419  433A  E6 10           7.0          ANI     P.TUPRES        ;SAVE ONLY PRES BIT
1420  433C  C2 44 43        10.0         JNZ     TST01C          ;CONTINUE IF ITS SET
1421  433F                               ERR     TST01X,TST01C
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  433F  CD 09 28        18.0                         CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4342  02                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4343  00                          .BYTE
 (1)  4344  CD 15 28        18.0         TST01C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4347  DA 27 43        10.0                         JC      TST01X          ;LOOP ADDRESS IF LOOP SPECIFIED
1422                                ;>PORT 0 'TU PRES' BIT NOT SET
1423
1424  434A  3E 80           7.0  DPEN00: MVI     A,$80           ;LOAD MTA #0 REGISTER 0 SELECT CODE
1425  434C  D3 40          10.0         OUT     TCMD
1426  434E  3E 7C           7.0          MVI     A,$7C           ;SET THE MTA #0 COMMAND BITS
1427  4350                               ROUT    EDATA           ;
 (1)  4350  D3 95          10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4352  7F              4.0                 MOV     A,A             ;RETRY LINK
1428  4353  D3 40          10.0         OUT     TCMD
1429  4355  00              4.0          NOP                     ;
1430  4356  3E 81           7.0          MVI     A,$81           ;LOAD MTA #0 REGISTER 1 SELECT CODE
1431  4358  D3 40          10.0         OUT     TCMD
1432  435A  00              4.0          NOP                     ;WAIT
1433  435B  DB 40          10.0         IN      TSTS            ;INPUT THE MTA STATUS A BYTE
1434  435D  E6 7C           7.0          ANI     $7C             ;REMOVE UNWANTED BITS
1435  435F                               ROUT    ADATA           ;
 (1)  435F  D3 94          10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4361  7F              4.0                 MOV     A,A             ;RETRY LINK
1436  4362  FE 7C           7.0          CPI     $7C             ;
1437  4364  CA 6C 43        10.0         JZ      DPEN01          ;CONTINUE IF SET
1438  4367                               ERRB    DPEN00,DPEN01
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4367  CD 12 28        18.0                         CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0003                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  436A  03                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  436B  00                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  436C  CD 15 28        18.0         DPEN01::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  436F  DA 4A 43        10.0                         JC      DPEN00          ;LOOP ADDRESS IF LOOP SPECIFIED
1439                            ;>COMMAND BYTE WRITTEN NOT = STATUS BYTE READ
1440  4372  3E 80           7.0  DPEN02: MVI     A,$80           ;LOAD MTA #0 REGISTER 0 SELECT CODE
1441  4374  D3 40          10.0         OUT     TCMD            ;
1442  4376  AF              4.0          XRA     A               ;CLEAR THE MTA #0 LOOP WRITE/READ BIT
1443  4377  D3 40          10.0         OUT     TCMD
1444  4379  00              4.0          NOP
1445  437A  3E 81           7.0          MVI     A,$81           ;LOAD MTA #0 REGISTER 1 SELECT CODE
```

```
1446  437C  D3  40          10.0        OUT     TCMD
1447  437E  00               4.0        NOP                         ;WAIT
1448  437F  DB  40          10.0        IN      TSTS                ;INPUT THE MTA STATUS A BYTE
1449  4381  E6  04           7.0        ANI     $04                 ;CHECK FOR LWR RESET
1450  4383  CA  8B  43      10.0        JZ      DPEN03              ;CONTINUE IF ZERO
1451  4386                             ERR     DPEN02,DPEN03
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4386  CD  09  28      18.0                CALL    ERLP                ;PROCESS ERROR - DO 2.3
 (1)        0004                      MSGN    =       MSGN+1                      ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4389  04                        .BYTE   MSGN                ;MESSAGE NUMBER ID
 (1)  438A  00                        .BYTE
 (1)  438B  CD  15  28      18.0      DPEN03::        CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.3
 (1)  438E  DA  72  43      10.0                JC      DPEN02              ;LOOP ADDRESS IF LOOP SPECIFIED
1452                           ;>STATUS BYTE NOT = ZERO WHEN COMMAND = 0
1453                                           ;
1454  4391                             ENDTST  TST01X
 (1)                           ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4391                             REQ     7                   ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4391  CD  06  28      18.0                CALL    REQST
 (2)  4394  00                        .BYTE                       ;DATA PATTERN NUMBER
 (2)  4395  00  00                    .WORD                       ;SYSTEM "" COUNT
 (2)  4397  00  00                    .WORD                       ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4399  00                        .BYTE                       ;DATA COMPARE FLAG IF =1
 (2)  439A  07                        .BYTE   7                   ;REQUEST CODE
 (1)  439B  3A  9A  4F      13.0        LDA     ITERA               ;GET ITERATION COUNT
 (1)  439E  3D               4.0        DCR     A                   ;DOWNCOUNT
 (1)  439F  32  9A  4F      13.0        STA     ITERA               ;SAVE COUNT
 (1)  43A2  F2  27  43      10.0        JP      TST01X              ;DO TEST UNTIL TILL = 0
1455
1456
```

```
                                .SBTTL  TEST 02 - LOOP COMMAND TO STATUS - MTA #1

43A5                            ST
 (1)                            ;*****************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*LOOP COMMAND TO STATUS TEST-MTA #1
43A5                            SD
 (1)                            ;*****************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE COMMAND BITS TO SET AND RESET, FOR
                                ;*MTA #1.
43A5                            SP
 (1)                            ;*****************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #1 NOT SELECTED BY USER
                                ;*     :   THEN-NEXT TEST
                                ;*     :   ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT PORT 1
                                ;*     IF TU PRESENT FOR TU PORT 1 = 0
                                ;*     :   THEN-ERROR 05
                                ;*     :   ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     SET THE FWD, REV, WRT, WRT INH AND LWR COMMAND BITS FOR MTA #1
                                ;*     INPUT THE STATUS BYTE A FOR MTA #1
                                ;*     IF FWD, REV, WRT, WRT INH AND LWR BITS SET
                                ;*     :   THEN-CONTINUE
                                ;*     :   ELSE-ERROR 06
                                ;*     ENDIF
                                ;*     RESET THE COMMAND BITS FOR MTA #1
                                ;*     INPUT THE STATUS BYTE A FOR MTA #1
                                ;*     MASK OUT FWD, REV, WRT, WRT INH AND LWR BITS
                                ;*     IF ALL BITS OF RESULT RESET
                                ;*     :   THEN-CONTINUE
                                ;*     :   ELSE-ERROR 07
                                ;*     ENDIF
                                ;*ENDTST
43A5                            SE
 (1)                            ;*****************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA1 MICRO TEST 02
                                ;*MTA1 MICRO ERROR 05
                                ;*MTA1-LOOP COMMAND TO STATUS-MTA #1
                                ;*M8954, M8955'S
                                ;*PORT 1 'TU PRES' BIT NOT SET
                                ;*
                                ;*MTA1 MICRO TEST 02
                                ;*MTA1 MICRO ERROR 06
```

```
                                        ;*MTA1-LOOP COMMAND TO STATUS-MTA #1
                                        ;*M8954, M8955'S
                                        ;*COMMAND BYTE WRITTEN NOT = STATUS BYTE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA1 MICRO TEST 02
                                        ;*MTA1 MICRO ERROR 07
                                        ;*MTA1-LOOP COMMAND TO STATUS-MTA #1
                                        ;*M8954, M8955'S
                                        ;*STATUS BYTE NOT = ZERO WHEN COMMAND = 0
1511  43A5                              $
 (1)                                    ; ************************************************************************
1512
1513  43A5                              TEST02: TESTX   @02             ;INITIALIZE THE TEST
 (1)  43A5  3E  02          7.0                  MVI     A,@02           ;DEFINE THE TEST NUMBER
 (1)  43A7  CD  03  28      18.0                 CALL    TSET            ;SETUP THE TEST
1514                                    ;%MTA1-LOOP COMMAND TO STATUS-MTA #1
1515                                    ;&M8954, M8955'S
1516
1517  43AA  3A  CD  49      13.0  TST02X: LDA     UNITMP          ;GET THE UNIT MAP
1518  43AD  E6  02           7.0          ANI     @02             ;TEST FOR UNIT 1
1519  43AF  CA  2A  44      10.0          JZ      TEST03          ;NO-CHECK FOR TEST 03
1520                                                              ;
1521  43B2  CD  A2  49      18.0          CALL    CLEAR           ;CLEAN UP THE PORTS
1522  43B5  DB  E0          10.0          IN      INTSTA          ;GET SELECT STATUS
1523  43B7  E6  80           7.0          ANI     BIT7
1524  43B9  F6  01           7.0          ORI     1
1525  43BB  D3  E0          10.0          OUT     MBSEL           ;SELECT PORT 1
1526
1527  43BD  DB  48          10.0          IN      PSTAT           ;GET TAPE STATUS TO SEE IF A UNIT IS PRESENT
1528  43BF  E6  10           7.0          ANI     P.TUPRES        ;SAVE ONLY PRES BIT
1529  43C1  C2  C9  43      10.0          JNZ     TST02C          ;CONTINUE IF ITS SET
1530  43C4                              ERR     TST02X,TST02C
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  43C4  CD  09  28      18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0005                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43C7  05                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43C8  00                                 .BYTE
 (1)  43C9  CD  15  28      18.0          TST02C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43CC  DA  AA  43      10.0                 JC      TST02X          ;LOOP ADDRESS IF LOOP SPECIFIED
1531                                    ;>PORT 1 'TU PRES' BIT NOT SET
1532  43CF  3E  80           7.0  OPEN10: MVI     A,$80
1533  43D1  D3  40          10.0          OUT     TCMD
1534  43D3  3E  7C           7.0          MVI     A,$7C           ;SET THE MTA #0 COMMAND BITS
1535  43D5                              ROUT    EDATA           ;
 (1)  43D5  D3  95          10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  43D7  7F               4.0                 MOV     A,A             ;RETRY LINK
1536  43D8  D3  40          10.0          OUT     TCMD
1537  43DA  00               4.0          NOP
1538  43DB  3E  81           7.0          MVI     A,$81
1539  43DD  D3  40          10.0          OUT     TCMD
1540  43DF  00               4.0          NOP                     ;
1541                                                              ;
```

```
1542  43E0  DB  40          10.0        IN    TSTS            ;GET THE MTA STATUS A
1543  43E2  E6  7C           7.0        ANI   $7C             ;REMOVE UNWANTED BITS
1544  43E4                             ROUT  ADATA           ;
 (1)  43E4  D3  94          10.0              OUT   ADATA             ;WRITE AC INTO ADATA
 (1)  43E6  7F               4.0              MOV   A,A               ;RETRY LINK
1545  43E7  FE  7C           7.0        CPI   $7C             ;
1546  43E9  CA  F1  43      10.0        JZ    DPEN11          ;CONTINUE IF SET
1547  43EC                             ERRB  DPEN10,DPEN11
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  43EC  CD  12  28      18.0              CALL  ERLPB             ;PROCESS ERROR - DO 2.3
 (1)      0006                              MSGN  =     MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43EF  06                              .BYTE MSGN              ;MESSAGE NUMBER ID
 (1)  43F0  00                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  43F1  CD  15  28      18.0        DPEN11::  CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43F4  DA  CF  43      10.0              JC    DPEN10            ;LOOP ADDRESS IF LOOP SPECIFIED
1548                        ;>COMMAND BYTE WRITTEN NOT = STATUS BYTE READ
1549  43F7  3E  80           7.0  DPEN12: MVI   A,$80           ;LOAD MTA #1 REGISTER 0 SELECT CODE
1550  43F9  D3  40          10.0        OUT   TCMD            ;
1551  43FB  AF               4.0        XRA   A               ;CLEAR THE MTA #1 LOOP WRITE/READ BIT
1552  43FC  D3  40          10.0        OUT   TCMD
1553  43FE  00               4.0        NOP
1554  43FF  3E  81           7.0        MVI   A,$81           ;LOAD MTA #1 REGISTER 1 SELECT CODE
1555  4401  D3  40          10.0        OUT   TCMD
1556  4403  00               4.0        NOP                   ;WAIT
1557  4404  DB  40          10.0        IN    TSTS            ;INPUT THE MTA STATUS A BYTE
1558  4406  E6  04           7.0        ANI   $04             ;CHECK FOR LWR RESET
1559  4408  CA  10  44      10.0        JZ    DPEN13          ;CONTINUE IF ZERO
1560  440B                             ERR   DPEN12,DPEN13
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  440B  CD  09  28      18.0              CALL  ERLP              ;PROCESS ERROR - DO 2.3
 (1)      0007                              MSGN  =     MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  440E  07                              .BYTE MSGN              ;MESSAGE NUMBER ID
 (1)  440F  00                              .BYTE
 (1)  4410  CD  15  28      18.0        DPEN13::  CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4413  DA  F7  43      10.0              JC    DPEN12            ;LOOP ADDRESS IF LOOP SPECIFIED
1561                        ;>STATUS BYTE NOT = ZERO WHEN COMMAND = 0
1562
1563  4416                        ENDTST  TST02X
 (1)                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4416                        REQ     7                      ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4416  CD  06  28      18.0              CALL  REQST
 (2)  4419  00                              .BYTE                   ;DATA PATTERN NUMBER
 (2)  441A  00  00                          .WORD                   ;SYSTEM "" COUNT
 (2)  441C  00  00                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  441E  00                              .BYTE                   ;DATA COMPARE FLAG IF  1
 (2)  441F  07                              .BYTE 7                 ;REQUEST CODE
 (1)  4420  3A  9A  4F      13.0        LDA   ITERA           ;GET ITERATION COUNT
 (1)  4423  3D               4.0        DCR   A               ;DOWNCOUNT
 (1)  4424  32  9A  4F      13.0        STA   ITERA           ;SAVE COUNT
 (1)  4427  F2  AA  43      10.0        JP    TST02X          ;DO TEST UNTIL TILL = 0
1564
1565
```

```
                                .SBTTL  TEST 03 - LOOP COMMAND TO STATUS - MTA #2

                                ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP COMMAND TO STATUS TEST-MTA #2
                                SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE COMMAND BITS TO SET AND RESET, FOR
                                ;*MTA #2.
                                SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #2 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT PORT 2
                                ;*    IF TU PRESENT FOR TU PORT 2 = 0
                                ;*    :   THEN-ERROR 10
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    SET THE FWD, REV, WRT, WRT INH AND LWR COMMAND BITS FOR MTA #2
                                ;*    INPUT THE STATUS BYTE A FOR MTA #2
                                ;*    IF FWD, REV, WRT, WRT INH AND LWR BITS SET
                                ;*    :   THEN-CONTINUE
                                ;*    :   ELSE-ERROR 11
                                ;*    ENDIF
                                ;*    RESET THE COMMAND BITS FOR MTA #2
                                ;*    INPUT THE STATUS BYTE A FOR MTA #2
                                ;*    MASK OUT FWD, REV, WRT, WRT INH AND LWR BITS
                                ;*    IF ALL BITS OF RESULT RESET
                                ;*    :   THEN-CONTINUE
                                ;*    :   ELSE-ERROR 12
                                ;*    ENDIF
                                ;*ENDTST
                                SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA1 MICRO TEST 03
                                ;*MTA1 MICRO ERROR 10
                                ;*MTA1-LOOP COMMAND TO STATUS-MTA #2
                                ;*M8954, M8955'S
                                ;*PORT 2 'TU PRES' BIT NOT SET
                                ;*
                                ;*MTA1 MICRO TEST 03
                                ;*MTA1 MICRO ERROR 11
```

```
                                        ;*MTA1-LOOP COMMAND TO STATUS-MTA #2
                                        ;*M8954, M8955'S
                                        ;*COMMAND BYTE WRITTEN NOT = STATUS BYTE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA1 MICRO TEST 03
                                        ;*MTA1 MICRO ERROR 12
                                        ;*MTA1-LOOP COMMAND TO STATUS-MTA #2
                                        ;*M8954, M8955'S
                                        ;*STATUS BYTE NOT = ZERO WHEN COMMAND = 0
1620  442A                              $
 (1)                                    ; ************************************************************************************

                                        TEST03: TESTX   @03             ;INITIALIZE THE TEST
 (1)  442A  3E  03          7.0                  MVI     A,@03           ;DEFINE THE TEST NUMBER
 (1)  442C  CD  03  28     18.0                  CALL    TSET            ;SETUP THE TEST
                                        ;%MTA1-LOOP COMMAND TO STATUS-MTA #2
                                        ;&M8954, M8955'S

442F  3A  CD  49          13.0  TST03X: LDA     UNITMP          ;GET THE UNIT MAP
4432  E6  04                7.0         ANI     @04             ;TEST FOR UNIT 2
4434  CA  AF  44           10.0         JZ      TEST04          ;NO-GO CHECK FOR TEST 04
                                        ;
4437  CD  A2  49           18.0         CALL    CLEAR           ;CLEAN UP THE PORTS
443A  DB  E0               10.0         IN      INTSTA          ;GET SELECT STATUS
443C  E6  80                7.0         ANI     BIT7
443E  F6  02                7.0         ORI     2
4440  D3  E0               10.0         OUT     MBSEL           ;SELECT PORT 2

4442  DB  48               10.0         IN      PSTAT           ;GET TAPE STATUS TO SEE IF A UNIT IS PRESENT
4444  E6  10                7.0         ANI     P.TUPRES        ;SAVE ONLY PRES BIT
4446  C2  4E  44           10.0         JNZ     TST03C          ;CONTINUE IF ITS SET
4449                                    ERR     TST03X,TST03C
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4449  CD  09  28     18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0008                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  444C  08                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  444D  00                                 .BYTE
 (1)  444E  CD  15  28     18.0          TST03C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4451  DA  2F  44     10.0                  JC      TST03X          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>PORT 2 'TU PRES' BIT NOT SET

4454  3E  80                7.0  DPEN20: MVI     A,$80
4456  D3  40               10.0         OUT     TCMD
4458  3E  7C                7.0         MVI     A,$7C           ;SET THE MTA #2 COMMAND BITS
445A                                    ROUT    EDATA           ;
 (1)  445A  D3  95         10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  445C  7F              4.0                  MOV     A,A             ;RETRY LINK
445D  D3  40               10.0         OUT     TCMD
445F  00                    4.0         NOP
4460  3E  81                7.0         MVI     A,$81
4462  D3  40               10.0         OUT     TCMD
4464  00                    4.0         NOP                     ;
```

```
1651
1652  4465  DB 40          10.0          IN      TSTS            ;
1653  4467  E6 7C           7.0          ANI     $7C             ;GET THE MTA STATUS A
1654  4469                               ROUT    ADATA           ;REMOVE UNWANTED BITS
 (1)  4469  D3 94           10.0                  OUT     ADATA           ;
 (1)  446B  7F              4.0                   MOV     A,A             ;WRITE AC INTO ADATA
1655  446C  FE 7C           7.0          CPI     $7C             ;RETRY LINK
1656  446E  CA 76 44        10.0          JZ      DPEN21          ;
1657  4471                               ERRB    DPEN20,DPEN21   ;CONTINUE IF SET
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4471  CD 12 28        18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0009                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4474  09                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4475  00                                  .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4476  CD 15 28        18.0          DPEN21::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4479  DA 54 44        10.0                  JC      DPEN20          ;LOOP ADDRESS IF LOOP SPECIFIED
1658                           ;>COMMAND BYTE WRITTEN NOT = STATUS BYTE READ
1659  447C  3E 80           7.0  DPEN22: MVI     A,$80           ;LOAD MTA #2 REGISTER 0 SELECT CODE
1660  447E  D3 40           10.0         OUT     TCMD            ;
1661  4480  AF              4.0          XRA     A               ;CLEAR THE MTA #2 LOOP WRITE/READ BIT
1662  4481  D3 40           10.0         OUT     TCMD
1663  4483  00              4.0          NOP
1664  4484  3E 81           7.0          MVI     A,$81           ;LOAD MTA #2 REGISTER 1 SELECT CODE
1665  4486  D3 40           10.0         OUT     TCMD
1666  4488  00              4.0          NOP                     ;WAIT
1667  4489  DB 40           10.0         IN      TSTS            ;INPUT THE MTA STATUS A BYTE
1668  448B  E6 04           7.0          ANI     $04             ;CHECK FOR LWR RESET
1669  448D  CA 95 44        10.0         JZ      DPEN23          ;CONTINUE IF ZERO
1670  4490                               ERR     DPEN22,DPEN23
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4490  CD 09 28        18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000A                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4493  0A                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4494  00                                  .BYTE
 (1)  4495  CD 15 28        18.0          DPEN23::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4498  DA 7C 44        10.0                  JC      DPEN22          ;LOOP ADDRESS IF LOOP SPECIFIED
1671                           ;>STATUS BYTE NOT = ZERO WHEN COMMAND   0
1672
1673  449B                         ENDTST  TST03X
 (1)                             ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  449B                       REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  449B  CD 06 28        18.0                  CALL    REQST
 (2)  449E  00                                  .BYTE                   ;DATA PATTERN NUMBER
 (2)  449F  00 00                               .WORD                   ;SYSTEM '*' COUNT
 (2)  44A1  C0 00                               .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  44A3  00                                  .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  44A4  07                                  .BYTE   7               ;REQUEST CODE
 (1)  44A5  3A 9A 4F        13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  44A8  3D              4.0           DCR     A               ;DOWNCOUNT
 (1)  44A9  32 9A 4F        13.0          STA     ITERA           ;SAVE COUNT
 (1)  44AC  F2 2F 44        10.0          JP      TST03X          ;DO TEST UNTIL TILL = 0
1674
1675
```

```
                                .SBTTL  TEST 04 - LOOP COMMAND TO STATUS - MTA #3

44AF                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*LOOP COMMAND TO STATUS TEST-MTA #3
44AF                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE COMMAND BITS TO SET AND RESET, FOR
                                ;*MTA #3.
44AF                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #3 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT PORT 3
                                ;*    IF TU PRESENT FOR TU PORT 3 = 0
                                ;*    :   THEN-ERROR 13
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    SET THE FWD, REV, WRT, WRT INH AND LWR COMMAND BITS FOR MTA #3
                                ;*    INPUT THE STATUS BYTE A FOR MTA #3
                                ;*    IF FWD, REV, WRT, WRT INH AND LWR BITS SET
                                ;*    :   THEN-CONTINUE
                                ;*    :   ELSE-ERROR 14
                                ;*    ENDIF
                                ;*    RESET THE COMMAND BITS FOR MTA #3
                                ;*    INPUT THE STATUS BYTE A FOR MTA #3
                                ;*    MASK OUT FWD, REV, WRT, WRT INH AND LWR BITS
                                ;*    IF ALL BITS OF RESULT RESET
                                ;*    :   THEN-CONTINUE
                                ;*    :   ELSE-ERROR 15
                                ;*    ENDIF
                                ;*ENDTST
44AF                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA1 MICRO TEST 04
                                ;*MTA1 MICRO ERROR 13
                                ;*MTA1-LOOP COMMAND TO STATUS-MTA #3
                                ;*M8954, M8955'S
                                ;*PORT 3 'TU PRES' BIT NOT SET
                                ;*
                                ;*MTA1 MICRO TEST 04
                                ;*MTA1 MICRO ERROR 14
```

```
                                        ;*MTA1-LOOP COMMAND TO STATUS-MTA #3
                                        ;*M8954, M8955'S
                                        ;*COMMAND BYTE WRITTEN NOT = STATUS BYTE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA1 MICRO TEST 04
                                        ;*MTA1 MICRO ERROR 15
                                        ;*MTA1-LOOP COMMAND TO STATUS-MTA #3
                                        ;*M8954, M8955'S
                                        ;*STATUS BYTE NOT = ZERO WHEN COMMAND = 0
1730  44AF                              S
 (1)                                    ; ************************************************************************************
1731
1732  44AF                              TEST04: TESTx   @04             ;INITIALIZE THE TEST
 (1)  44AF  3E  04          7.0                        MVI     A,@04           ;DEFINE THE TEST NUMBER
 (1)  44B1  CD  03  28     18.0                        CALL    TSET            ;SETUP THE TEST
1733                                    ;%MTA1-LOOP COMMAND TO STATUS-MTA #3
1734                                    ;&M8954, M8955'S
1735
1736  44B4  3A  CD  49     13.0  TST04X: LDA     UNITMP          ;GET THE UNIT MAP
1737  44B7  E6  08          7.0        ANI     @010            ;TEST FOR UNIT 3
1738  44B9  CA  34  45     10.0        JZ      TEST05          ;NO-GO CHECK FOR TEST 05
1739                                                            ;
1740  44BC  CD  A2  49     18.0        CALL    CLEAR           ;CLEAN UP THE PORTS
1741  44BF  DB  E0         10.0        IN      INTSTA          ;GET SELECT STATUS
1742  44C1  E6  80          7.0        ANI     BIT7
1743  44C3  F6  03          7.0        ORI     3
1744  44C5  D3  E0         10.0        OUT     MBSEL           ;SELECT PORT 3
1745
1746  44C7  DB  48         10.0        IN      PSTAT           ;GET TAPE STATUS TO SEE IF A UNIT IS PRESENT
1747  44C9  E6  10          7.0        ANI     P.TUPRES        ;SAVE ONLY PRES BIT
1748  44CB  C2  D3  44     10.0        JNZ     TST04C          ;CONTINUE IF ITS SET
1749  44CE                              ERR     TST04X,TST04C
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44CE  CD  09  28     18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000B                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44D1  0B                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44D2  00                                       .BYTE
 (1)  44D3  CD  15  28     18.0                TST04C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44D6  DA  B4  44     10.0                        JC      TST04X          ;LOOP ADDRESS IF LOOP SPECIFIED
1750                                    ;>PORT 3 'TU PRES' BIT NOT SET
1751
1752  44D9  3E  80          7.0  DPEN30: MVI     A,$80
1753  44DB  D3  40         10.0        OUT     TCMD
1754  44DD  3E  7C          7.0        MVI     A,$7C           ;SET THE MTA #3 COMMAND BITS
1755  44DF                              ROUT    EDATA           ;
 (1)  44DF  D3  95         10.0                OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44E1  7F              4.0                MOV     A,A             ;RETRY LINK
1756  44E2  D3  40         10.0        OUT     TCMD
1757  44E4  00              4.0        NOP
1758  44E5  3E  81          7.0        MVI     A,$81
1759  44E7  D3  40         10.0        OUT     TCMD
1760  44E9  00              4.0        NOP                     ;
```

```
1761                                                    ;
1762  44EA  DB  40          10.0       IN    TSTS       ;GET THE MTA STATUS A
1763  44EC  E6  7C           7.0       ANI   $7C        ;REMOVE UNWANTED BITS
1764  44EE                            ROUT  ADATA      ;
 (1)  44EE  D3  94          10.0             OUT   ADATA          ;WRITE AC INTO ADATA
 (1)  44F0  7F               4.0             MOV   A,A            ;RETRY LINK
1765  44F1  FE  7C           7.0       CPI   $7C        ;
1766  44F3  CA  FB  44       10.0       JZ    DPEN31     ;CONTINUE IF SET
1767  44F6                            ERRB  DPEN30,DPEN31
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44F6  CD  12  28       18.0             CALL  ERLPB          ;PROCESS ERROR - DO 2.3
 (1)      000C                               MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44F9  0C                               .BYTE MSGN           ;MESSAGE NUMBER ID
 (1)  44FA  00                               .BYTE                ;PRINT ROUTINE NUMBER
 (1)  44FB  CD  15  28       18.0       DPEN31::  CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44FE  DA  D9  44       10.0             JC    DPEN30         ;LOOP ADDRESS IF LOOP SPECIFIED
1768                                ;>COMMAND BYTE WRITTEN NOT = STATUS BYTE READ
1769  4501  3E  80           7.0  DPEN32: MVI   A,$80      ;LOAD MTA #3 REGISTER 0 SELECT CODE
1770  4503  D3  40          10.0       OUT   TCMD       ;
1771  4505  AF               4.0       XRA   A          ;CLEAR THE MTA #3 LOOP WRITE/READ BIT
1772  4506  D3  40          10.0       OUT   TCMD
1773  4508  00               4.0       NOP
1774  4509  3E  81           7.0       MVI   A,$81      ;LOAD MTA #3 REGISTER 1 SELECT CODE
1775  450B  D3  40          10.0       OUT   TCMD
1776  450D  00               4.0       NOP              ;WAIT
1777  450E  DB  40          10.0       IN    TSTS       ;INPUT THE MTA STATUS A BYTE
1778  4510  E6  04           7.0       ANI   $04        ;CHECK FOR LWR RESET
1779  4512  CA  1A  45       10.0       JZ    DPEN33     ;CONTINUE IF ZERO
1780  4515                            ERR   DPEN32,DPEN33
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4515  CD  09  28       18.0             CALL  ERLP           ;PROCESS ERROR - DO 2.3
 (1)      000D                               MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4518  0D                               .BYTE MSGN           ;MESSAGE NUMBER ID
 (1)  4519  00                               .BYTE
 (1)  451A  CD  15  28       18.0       DPEN33::  CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  451D  DA  01  45       10.0             JC    DPEN32         ;LOOP ADDRESS IF LOOP SPECIFIED
1781                                ;>STA'S BYTE NOT = ZERO WHEN COMMAND = 0
1782
1783  4520                            ENDTST  TST04X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4520                          REQ   7                ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4520  CD  06  28       18.0             CALL  REQST
 (2)  4523  00                               .BYTE                ;DATA PATTERN NUMBER
 (2)  4524  00  00                           .WORD                ;SYSTEM '"' COUNT
 (2)  4526  00  00                           .WORD                ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4528  00                               .BYTE                ;DATA COMPARE FLAG IF =1
 (2)  4529  07                               .BYTE  7             ;REQUEST CODE
 (1)  452A  3A  9A  4F       13.0       LDA   ITERA      ;GET ITERATION COUNT
 (1)  452D  3D               4.0        DCR   A          ;DOWNCOUNT
 (1)  452E  32  9A  4F       13.0       STA   ITERA      ;SAVE COUNT
 (1)  4531  F2  B4  44       10.0       JP    TST04X     ;DO TEST UNTIL TILL = 0
1784
```

```
                                .SBTTL  TEST 05 - LOOP AMTIE TO AMTIE - MTA #0

                                ST
                                ;*****************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP AMTIE TO AMTIE TEST-MTA #0
                                SD
                                ;*****************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE DATA PATH FROM TU PORT #0 TO
                                ;*MTA #0 - #2 AND BACK VIA THE AMTIE LINES.  THROUGH THE MTA #0 AMTIE
                                ;*REGISTER TO THE TU PORT #0 AMTIE REGISTER.
                                SP
                                ;*****************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #0 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT PORT 0
                                ;*    CLEAR THE AMTIE DATA
                                ;*    BGNDO
                                ;*    :   ISSUE 7 BITS OF THE CONTENTS OF AMTIE DATA TO MTA #0 AMTIE DATA REGISTER
                                ;*    :   ISSUE THE MSB OF THE AMTIE DATA TO MTA #0 AMTIE 7 DATA BIT
                                ;*    :   ISSUE THE LSB OF THE AMTIE DATA TO MTA #0 AMTIE PARITY REGISTER
                                ;*    :   INPUT THE TU PORT #0 AMTIE DATA REGISTER
                                ;*    :   IF AMTIE DATA READ=AMTIE DATA
                                ;*    :   :   THEN-CONTINUE
                                ;*    :   :   ELSE-ERROR 16
                                ;*    :   ENDIF
                                ;*    :   INPUT THE AMTIE BIT 7 AND PARITY BIT
                                ;*    :   IF THE AMTIE BIT 7 AND PARITY READ=AMTIE BIT 7 AND PARITY WRITTEN
                                ;*    :   :   THEN-CONTINUE
                                ;*    :   :   ELSE-ERROR 17
                                ;*    :   ENDIF
                                ;*    :   INCREMENT THE AMTIE DATA
                                ;*    :   DO UNTIL THE AMTIE DATA=ZERO
                                ;*    ENDDO
                                ;*    CALL SUBROUTINE CLEAR
                                ;*ENDTST
                                SE
                                ;*****************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA1 MICRO TEST 05
                                ;*MTA1 MICRO ERROR 16
                                ;*MTA1-LOOP AMTIE TO AMTIE-MTA #0
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN TO MTA AMTIE NOT - DATA READ
```

```
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA1 MICRO TEST 05
                                        ;*MTA1 MICRO ERROR 17
                                        ;*MTA1-LOOP AMTIE TO AMTIE-MTA #0
                                        ;*M8954, M8955'S
                                        ;*VALUE WRITTEN TO MTA AMTIE PTY. NOT = VALUE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4534                                    $
 (1)                                    ; ************************************************************************
                                        
4534                                    TEST05: TESTX   @05             ;SET UP THE TEST NUMBER
 (1)  4534  3E  05          7.0                         MVI     A,@05          ;DEFINE THE TEST NUMBER
 (1)  4536  CD  03  28     18.0                         CALL    TSET           ;SETUP THE TEST
                                        ;%MTA1-LOOP AMTIE TO AMTIE-MTA #0
                                        ;&M8954, M8955'S
                                        
4539  3A  CD  49          13.0  TST05X: LDA     UNITMP          ;GET THE UNIT MAP
453C  E6  01                7.0        ANI     @001            ;TEST UNIT 0?
453E  CA  BB  45           10.0        JZ      TEST06          ;NO-CHECK FOR TEST06
4541  CD  A2  49           18.0        CALL    CLEAR           ;CLEAN UP THE PORTS
4544  DB  E0               10.0        IN      INTSTA          ;GET SELECT STATUS
4546  E6  80                7.0        ANI     BIT7
4548  D3  E0               10.0        OUT     MBSEL           ;SELECT PORT 0
454A  21  CE  49           10.0        LXI     H,DATA          ;LOAD ADDRESS OF DATA
454D  AF                    4.0        XRA     A               ;CLEAR THE ACCUMULATOR
454E  77                    7.0        MOV     M,A             ;STORE IN DATA
                                        ;
454F  3E  80                7.0  ALP01: MVI     A,@200          ;SELECT MTA #0 REGISTER 0
4551  D3  40               10.0        OUT     TCMD
4553  7E                    7.0        MOV     A,M             ;GET THE AMTIE DATA
4554  E6  01                7.0        ANI     $01             ;GET THE PSEUDO PARITY BIT
4556  47                    4.0        MOV     B,A             ;SAVE IN B
4557  7E                    7.0        MOV     A,M             ;GET THE AMTIE DATA-AGAIN
4558  E6  80                7.0        ANI     @200            ;GET THE MSB
455A  07                    4.0        RLC                     ;POSITION DATA FOR OUTPUT
455B  07                    4.0        RLC
455C  B0                    4.0        ORA     B               ;OR IN THE PSEUDO PARITY BIT
455D  F6  04                7.0        ORI     @004            ;OR IN THE LOOP W/R BIT
455F  D3  40               10.0        OUT     TCMD            ;LOAD MTA REGISTER 0
4561  3E  82                7.0        MVI     A,@202          ;SELECT MTA #0 REGISTER 2
4563  D3  40               10.0        OUT     TCMD
4565  7E                    7.0        MOV     A,M             ;GET THE AMTIE DATA
4566  47                    4.0        MOV     B,A             ;SAVE FOR DATA COMPARE
4567  E6  7F                7.0        ANI     $7F             ;REMOVE THE MSB
4569  D3  40               10.0        OUT     TCMD            ;LOAD MTA REGISTER 2
456B  DB  44               10.0        IN      TAMT            ;READ THE AMTIE REGISTER
456D  B8                    4.0        CMP     B               ;COMPARE ACTUAL AND EXP.
456E  CA  7D  45           10.0        JZ      ALP02           ;EQUAL - CONTINUE
4571                                    ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  4571  D3  94          10.0                OUT     ADATA          ;WRITE AC INTO ADATA
 (1)  4573  7F               4.0                MOV     A,A            ;RETRY LINK
```

```
1877  4574  7E                  7.0          MOV     A,M             ;GET EXP. DATA
1878  4575                                   ROUT    EDATA           ;STORE EXP. DATA
 (1)  4575  D3  95              10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4577  7F                   4.0                 MOV     A,A             ;RETRY LINK
1879  4578                                   ERRB    ALP01,ALP02,0
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4578  CD  12  28          18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000E                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  457B  0E                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  457C  00                                       .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  457D  CD  15  28          18.0         ALP02:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4580  DA  4F  45          10.0                 JC      ALP01           ;LOOP ADDRESS IF LOOP SPECIFIED
1880                                   ;>DATA WRITTEN TO MTA AMTIE NOT = DATA READ
1881  4583  78                  4.0          MOV     A,B             ;GET EXP. PTY.
1882  4584  E6  01              7.0          ANI     $01             ;REMOVE UNWANTED BITS
1883  4586  47                  4.0          MOV     B,A             ;RESTORE EXP. PTY.
1884  4587  DB  48              10.0          IN      PSTAT           ;GET ACTUAL PTY.
1885  4589  E6  01              7.0          ANI     $01             ;REMOVE UNWANTED BITS
1886  458B  B8                  4.0          CMP     B               ;EQUAL?
1887  458C  CA  9B  45          10.0          JZ      A1LP02          ;YES
1888  458F                                   ROUT    ADATA           ;STORE ACTUAL PTY.
 (1)  458F  D3  94              10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4591  7F                   4.0                 MOV     A,A             ;RETRY LINK
1889  4592  78                  4.0          MOV     A,B             ;GET EXP. PTY.
1890  4593                                   ROUT    EDATA           ;STORE EXP. PTY.
 (1)  4593  D3  95              10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4595  7F                   4.0                 MOV     A,A             ;RETRY LINK
1891  4596                                   ERRB    ALP01,A1LP02
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4596  CD  12  28          18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000F                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4599  0F                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  459A  00                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  459B  CD  15  28          18.0         A1LP02:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  459E  DA  4F  45          10.0                 JC      ALP01           ;LOOP ADDRESS IF LOOP SPECIFIED
1892                                   ;>VALUE WRITTEN TO MTA AMTIE PTY. NOT = VALUE READ
1893
1894  45A1  7E                  7.0          MOV     A,M             ;GET DATA
1895  45A2  3C                  4.0          INR     A               ;UPDATE THE DATA BYTE
1896  45A3  77                  7.0          MOV     M,A             ;SAVE UPDATED DATA
1897  45A4  C2  4F  45          10.0          JNZ     ALP01           ;CONTINUE UNTIL DONE
1898  45A7                                   ENDTST  TST05X
 (1)                                       ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  45A7                                 REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  45A7  CD  06  28          18.0                 CALL    REQST
 (2)  45AA  00                                       .BYTE                   ;DATA PATTERN NUMBER
 (2)  45AB  00  00                                   .WORD                   ;SYSTEM "" COUNT
 (2)  45AD  00  00                                   .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  45AF  00                                       .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  45B0  07                                       .BYTE   7               ;REQUEST CODE
 (1)  45B1  3A  9A  4F           7.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  45B4  3D                   4.0          DCR     A               ;DOWNCOUNT
 (1)  45B5  32  9A  4F          13.0          STA     ITERA           ;SAVE COUNT
 (1)  45B8  F2  39  45          10.0          JP      TST05X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 06 - LOOP AMTIE TO AMTIE - MTA #1
45BB                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*----- -----
                                ;*LOOP AMTIE TO AMTIE TEST-MTA #1
45BB                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE DATA PATh FROM TU PORT #1 TO
                                ;*MTA #0 + #2 AND BACK VIA THE AMTIE LINES THROUGH THE MTA #1 AMTIE
                                ;*REGISTER TO THE TU PORT #1 AMTIE REGISTER.
45BB                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #1 NOT SELECTED BY USER
                                ;*     :   THEN-NEXT TEST
                                ;*     :   ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT PORT 1
                                ;*     CLEAR THE AMTIE DATA
                                ;*     BGNDO
                                ;*     :   ISSUE 7 BITS OF THE CONTENTS OF AMTIE DATA TO MTA #1 AMTIE DATA REGISTER
                                ;*     :   ISSUE THE MSB OF THE AMTIE DATA TO MTA #1 AMTIE 7 DATA BIT
                                ;*     :   ISSUE THE LSB OF THE AMTIE DATA TO MTA #1 AMTIE PARITY REGISTER
                                ;*     :   INPUT THE TU PORT #1 AMTIE DATA REGISTER
                                ;*     :   IF AMTIE DATA READ=AMTIE DATA
                                ;*     :   :   THEN-CONTINUE
                                ;*     :   :   ELSE-ERROR 20
                                ;*     :   ENDIF
                                ;*     :   INPUT THE AMTIE BIT 7 AND PARITY BIT
                                ;*     :   IF THE AMTIE BIT 7 AND PARITY READ-AMTIE BIT 7 AND PARITY WRITTEN
                                ;*     :   :   THEN-CONTINUE
                                ;*     :   :   ELSE-ERROR 21
                                ;*     :   ENDIF
                                ;*     :   CLEAR TU PORT #1 LOOP W/R BIT
                                ;*     :   INCREMENT THE AMTIE DATA
                                ;*     :   DO UNTIL THE AMTIE DATA=ZERO
                                ;*     ENDDO
                                ;*     CALL SUBROUTINE CLEAR
                                ;*ENDTST
45BB                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA1 MICRO TEST 06
                                ;*MTA1 MICRO ERROR 20
                                ;*MTA1-LOOP AMTIE TO AMTIE-MTA #1
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN TO MTA AMTIE NOT = DATA READ
```

```
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA1 MICRO TEST 06
                                        ;*MTA1 MICRO ERROR 21
                                        ;*MTA1-LOOP AMTIE TO AMTIE-MTA #1
                                        ;*M8954, M8955'S
                                        ;*VALUE WRITTEN TO MTA AMTIE PTY. NOT = VALUE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
45BB                                    $
 (1)                                    ; ************************************************************************
                                        TEST06: TESTX   @06             ;SET UP THE TEST NUMBER
 (1) 45BB 3E 06          7.0                            MVI     A,@06           ;DEFINE THE TEST NUMBER
 (1) 45BD CD 03 28      18.0                            CALL    TSET            ;SETUP THE TEST
                                        ;%MTA1-LOOP AMTIE TO AMTIE-MTA #1
                                        ;&M8954, M8955'S
45C0 3A CD 49          13.0 TST06X: LDA        UNITMP          ;GET THE UNIT MAP
45C3 E6 02              7.0         ANI     @002            ;TEST UNIT 1?
45C5 CA 44 46          10.0         JZ      TEST07          ;NO-CHECK FOR TEST07
45C8 CD A2 49          18.0         CALL    CLEAR           ;CLEAN UP THE PORTS
45CB DB E0             10.0         IN      INTSTA          ;GET SELECT STATUS
45CD E6 80              7.0         ANI     BIT7
45CF F6 01              7.0         ORI     1
45D1 D3 E0             10.0         OUT     MBSEL           ;SELECT PORT 1
45D3 21 CE 49          10.0         LXI     H,DATA          ;LOAD ADDRESS OF DATA
45D6 AF                 4.0         XRA     A               ;CLEAR THE ACCUMULATOR
45D7 77                 7.0         MOV     M,A             ;STORE IN DATA
45D8 3E 80              7.0 ALP11:  MVI     A,$80           ;SELECT MTA #1 REGISTER 0
45DA D3 40             10.0         OUT     TCMD
45DC 7E                 7.0         MOV     A,M             ;GET THE AMTIE DATA
45DD E6 01              7.0         ANI     $01             ;GET THE PSEUDO PARITY BIT
45DF 47                 4.0         MOV     B,A             ;SAVE IN B
45E0 7E                 7.0         MOV     A,M             ;GET THE AMTIE DATA-AGAIN
45E1 E6 80              7.0         ANI     $80             ;GET THE MSB
45E3 07                 4.0         RLC                     ;POSITION DATA FOR OUTPUT
45E4 07                 4.0         RLC
45E5 B0                 4.0         ORA     B               ;OR IN THE PSEUDO PARITY BIT
45E6 F6 04              7.0         ORI     @004            ;OR IN THE LOOP W/R BIT
45E8 D3 40             10.0         OUT     TCMD            ;LOAD MTA REGISTER 0
45EA 3E 82              7.0         MVI     A,$82           ;SELECT MTA #1 REGISTER 2
45EC D3 40             10.0         OUT     TCMD
45EE 7E                 7.0         MOV     A,M             ;GET THE AMTIE DATA
45EF 47                 4.0         MOV     B,A             ;SAVE FOR DATA COMPARE
45F0 E6 7F              7.0         ANI     $7F             ;REMOVE THE MSB
45F2 D3 40             10.0         OUT     TCMD            ;LOAD MTA REGISTER 2
45F4 DB 44             10.0         IN      TAMT            ;READ THE AMTIE REGISTER
45F6 B8                 4.0         CMP     B               ;COMPARE ACTUAL AND EXP.
45F7 CA 06 46          10.0         JZ      ALP12           ;EQUAL - CONTINUE
45FA                                ROUT    ADATA           ;STORE ACTUAL DATA
 (1) 45FA D3 94         10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1) 45FC 7F             4.0                 MOV     A,A             ;RETRY LINK
```

```
 1991  45FD  7E                    7.0          MOV     A,M             ;GET EXP. DATA
 1992  45FE                                     ROUT    FDATA           ;STORE EXP. DATA
  (1)  45FE  D3  95                10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  4600  7F                     4.0                 MOV     A,A             ;RETRY LINK
 1993  4601                                     ERRB    ALP11,ALP12,0
  (1)                                         ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  4601  CD  12  28            18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0010                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4604  10                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4605  00                                        .BYTE   0               ;PRINT ROUTINE NUMBER
  (1)  4606  CD  15  28            18.0         ALP12:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  4609  DA  D8  45            10.0                 JC      ALP11           ;LOOP ADDRESS IF LOOP SPECIFIED
 1994                                    ;>DATA WRITTEN TO MTA AMTIE NOT = DATA READ
 1995  460C  78                     4.0          MOV     A,B             ;GET EXP. PTY.
 1996  460D  E6  01                 7.0          ANI     $01             ;REMOVE UNWANTED BITS
 1997  460F  47                     4.0          MOV     B,A             ;RESTORE EXP. PTY.
 1998  4610  DB  48                10.0          IN      PSTAT           ;GET ACTUAL PTY.
 1999  4612  E6  01                 7.0          ANI     $01             ;REMOVE UNWANTED BITS
 2000  4614  B8                     4.0          CMP     B               ;EQUAL?
 2001  4615  CA  24  46            10.0          JZ      A2LP02          ;YES
 2002  4618                                     ROUT    ADATA           ;STORE ACTUAL PTY.
  (1)  4618  D3  94                10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  461A  7F                     4.0                 MOV     A,A             ;RETRY LINK
 2003  461B  78                     4.0          MOV     A,B             ;GET EXP. PTY.
 2004  461C                                     ROUT    EDATA           ;STORE EXP. PTY.
  (1)  461C  D3  95                10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  461E  7F                     4.0                 MOV     A,A             ;RETRY LINK
 2005  461F                                     ERRB    ALP11,A2LP02
  (1)                                         ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  461F  CD  12  28            18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0011                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4622  11                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4623  00                                        .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  4624  CD  15  28            18.0         A2LP02::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  4627  DA  D8  45            10.0                 JC      ALP11           ;LOOP ADDRESS IF LOOP SPECIFIED
 2006                                    ;>VALUE WRITTEN TO MTA AMTIE PTY. NOT = VALUE READ
 2007
 2008  462A  7E                     7.0          MOV     A,M             ;GET DATA
 2009  462B  3C                     4.0          INR     A               ;UPDATE THE DATA BYTE
 2010  462C  77                     7.0          MOV     M,A             ;SAVE UPDATED DATE
 2011  462D  C2  D8  45            10.0          JNZ     ALP11           ;CONTINUE UNTIL DONE
 2012  4630                                     ENDTST  TST06X
  (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  4630                                     REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  4630  CD  06  28            18.0                 CALL    REQST
  (2)  4633  00                                        .BYTE                   ;DATA PATTERN NUMBER
  (2)  4634  00  00                                    .WORD                   ;SYSTEM "M" COUNT
  (2)  4636  00  00                                    .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  4638  00                                        .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  4639  07                                        .BYTE   7               ;REQUEST CODE
  (1)  463A  3A  9A  4F            13.0          LDA     ITERA           ;GET ITERATION COUNT
  (1)  463D  3D                     4.0          DCR     A               ;DOWNCOUNT
  (1)  463E  32  9A  4F            13.0          STA     ITERA           ;SAVE COUNT
  (1)  4641  F2  C0  45            10.0          JP      TST06X          ;DO TEST UNTIL TILL - 0
```

```
                                .SBTTL  TEST 07 - LOOP AMTIE TO AMTIE - MTA #2
4644                            ST
 (1)                            ; ***************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*LOOP AMTIE TO AMTIE TEST-MTA #2
4644                            SD
 (1)                            ; ***************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE DATA PATH FROM TU PORT #2 TO
                                ;*MTA #0 AND #2 AND BACK VIA THE AMTIE LINES THROUGH THE MTA #2 AMTIE
                                ;*REGISTER TO TU PORT 2 AMTIE REGISTER.
4644                            SP
 (1)                            ; ***************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #2 NOT SELECTED BY USER
                                ;*     :   THEN-NEXT TEST
                                ;*     :   ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT PORT 2
                                ;*     CLEAR THE AMTIE DATA
                                ;*     BGNDO
                                ;*     :   ISSUE 7 BITS OF THE CONTENTS OF AMTIE DATA TO MTA #2 AMTIE DATA REGISTER
                                ;*     :   ISSUE THE MSB OF THE AMTIE DATA TO MTA #2 AMTIE 7 DATA BIT
                                ;*     :   ISSUE THE LSB OF THE AMTIE DATA REGISTER TO MTA #2 AMTIE PARITY REGISTER
                                ;*     :   INPUT THE TU PORT #2 AMTIE DATA REGISTER
                                ;*     :   IF AMTIE DATA READ=AMTIE DATA
                                ;*     :   :   THEN-CONTINUE
                                ;*     :   :   ELSE-ERROR 22
                                ;*     :   ENDIF
                                ;*     :   INPUT THE AMTIE BIT 7 ANDPARITY BIT
                                ;*     :   IF THE AMTIE BIT 7 AND PARITY READ=AMTIE BIT 7 AND PARITY WRITTEN
                                ;*     :   :   THEN-CONTINUE
                                ;*     :   :   ELSE-ERROR 23
                                ;*     :   ENDIF
                                ;*     :   CLEAR TU PORT #2 LOOP W/R BIT
                                ;*     :   INCREMENT THE AMTIE DATA
                                ;*     :   DO UNTIL THE AMTIE DATA=ZERO
                                ;*     ENDDO
                                ;*     CALL SUBROUTINE CLEAR
                                ;*ENDTST
4644                            SE
 (1)                            ; ***************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA1 MICRO TEST 07
                                ;*MTA1 MICRO ERROR 22
                                ;*MTA1-LOOP AMTIE TO AMTIE-MTA #2
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN TO MTA AMTIE NOT = DATA READ
```

```
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA1 MICRO TEST 07
                                        ;*MTA1 MICRO ERROR 23
                                        ;*MTA1-LOOP AMTIE TO AMTIE-MTA #2
                                        ;*M8954, M8955'S
                                        ;*VALUE WRITTEN TO MTA AMTIE PTY. NOT = VALUE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4644                                    S
 (1)                                    ; ************************************************************************************

                                        TEST07: TESTX   @07             ;SET UP THE TEST NUMBER
 (1) 4644 3E 07           7.0                           MVI     A,@07           ;DEFINE THE TEST NUMBER
 (1) 4646 CD 03 28       18.0                           CALL    TSET            ;SETUP THE TEST
                                        ;%MTA1-LOOP AMTIE TO AMTIE-MTA #2
                                        ;&M8954, M8955'S

4649 3A CD 49            13.0  TST07X: LDA     UNITMP          ;GET THE UNIT MAP
464C E6 04                7.0         ANI     @004            ;TEST FOR UNIT 2
464E CA CD 46            10.0         JZ      TEST10          ;NO-CHECK FOR TEST10
4651 CD A2 49            18.0         CALL    CLEAR           ;CLEAN UP THE PORTS
4654 DB E0               10.0         IN      INTSTA          ;GET SELECT STATUS
4656 E6 80                7.0         ANI     BIT7
4658 F6 02                7.0         ORI     2
465A D3 E0               10.0         OUT     MBSEL           ;SELECT PORT 2
465C 21 CE 49            10.0         LXI     H,DATA          ;LOAD ADDRESS OF DATA
465F AF                   4.0         XRA     A               ;CLEAR THE ACCUMULATOR
4660 77                   7.0         MOV     M,A             ;STORE IN DATA
4661 3E 80                7.0  ALP21:  MVI     A,$80           ;SELECT MTA #2 REGISTER 0
4663 D3 40               10.0         OUT     TCMD
4665 7E                   7.0         MOV     A,M             ;GET THE AMTIE DATA
4666 E6 01                7.0         ANI     $01             ;GET THE PSEUDO PARITY BIT
4668 47                   4.0         MOV     B,A             ;SAVE IN B
4669 7E                   7.0         MOV     A,M             ;GET THE AMTIE DATA-AGAIN
466A E6 80                7.0         ANI     $80             ;GET THE MSB
466C 07                   4.0         RLC                     ;POSITION DATA FOR OUTPUT
466D 07                   4.0         RLC
466E B0                   4.0         ORA     B               ;OR IN THE PSEUDO PARITY BIT
466F F6 04                7.0         ORI     @004            ;OR IN THE LOOP W/R BIT
4671 D3 40               10.0         OUT     TCMD            ;LOAD MTA REGISTER 0
4673 3E 82                7.0         MVI     A,$82           ;SELECT MTA #2 REGISTER 2
4675 D3 40               10.0         OUT     TCMD
4677 7E                   7.0         MOV     A,M             ;GET THE AMTIE DATA
4678 47                   4.0         MOV     B,A             ;SAVE FOR DATA COMPARE
4679 E6 7F                7.0         ANI     $7F             ;REMOVE THE MSB
467B D3 40               10.0         OUT     TCMD            ;LOAD MTA REGISTER 2
467D DB 44               10.0         IN      TAMT            ;READ THE AMTIE REGISTER
467F B8                   4.0         CMP     B               ;COMPARE ACTUAL AND EXP.
4680 CA 8F 46            10.0         JZ      ALP22           ;EQUAL - CONTINUE
4683                                   ROUT    ADATA           ;STORE ACTUAL DATA
 (1) 4683 D3 94           10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1) 4685 7F               4.0                   MOV     A,A             ;RETRY LINK
```

```
2105  4686  7E                    7.0          MOV     A,M             ;GET EXP. DATA
2106  4687                                     ROUT    EDATA           ;STORE EXP. DATA
 (1)  4687  D3  95                10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4689  7F                    4.0                  MOV     A,A             ;RETRY LINK
2107  468A                                     ERRB    ALP21,ALP22,0
 (1)                                         ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  468A  CD  12  28            18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0012                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  468D  12                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  468E  00                                        .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  468F  CD  15  28            18.0                ALP22:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4692  DA  61  46            10.0                 JC      ALP21           ;LOOP ADDRESS IF LOOP SPECIFIED
2108                                     ;>DATA WRITTEN TO MTA AMTIE NOT = DATA READ
2109  4695  78                    4.0          MOV     A,B             ;GET EXP. PTY.
2110  4696  E6  01                7.0          ANI     $01             ;REMOVE UNWANTED BITS
2111  4698  47                    4.0          MOV     B,A             ;RESTORE EXP. PTY.
2112  4699  DB  48                10.0         IN      PSTAT           ;GET ACTUAL PTY.
2113  469B  E6  01                7.0          ANI     $01             ;REMOVE UNWANTED BITS
2114  469D  B8                    4.0          CMP     B               ;EQUAL?
2115  469E  CA  AD  46            10.0         JZ      A3LP02          ;YES
2116  46A1                                     ROUT    ADATA           ;STORE ACTUAL PTY.
 (1)  46A1  D3  94                10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46A3  7F                    4.0                  MOV     A,A             ;RETRY LINK
2117  46A4  78                    4.0          MOV     A,B             ;GET EXP. PTY.
2118  46A5                                     ROUT    EDATA           ;STORE EXP. PTY.
 (1)  46A5  D3  95                10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  46A7  7F                    4.0                  MOV     A,A             ;RETRY LINK
2119  46A8                                     ERRB    ALP21,A3LP02
 (1)                                         ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46A8  CD  12  28            18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0013                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46AB  13                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46AC  00                                        .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  46AD  CD  15  28            18.0                A3LP02::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46B0  DA  61  46            10.0                 JC      ALP21           ;LOOP ADDRESS IF LOOP SPECIFIED
2120                                     ;>VALUE WRITTEN TO MTA AMTIE PTY. NOT = VALUE READ
2121
2122  46B3  7E                    7.0          MOV     A,M             ;GET DATA
2123  46B4  3C                    4.0          INR     A               ;UPDATE THE DATA BYTE
2124  46B5  77                    7.0          MOV     M,A             ;SAVE UPDATED DATA
2125  46B6  C2  61  46            10.0         JNZ     ALP21           ;CONTINUE UNTIL DONE
2126  46B9                                     ENDTST  TST07X
 (1)                                         ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  46B9                                   PEQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  46B9  CD  06  28            18.0                 CALL    REQST
 (2)  46BC  00                                        .BYTE                   ;DATA PATTERN NUMBER
 (2)  46BD  00  00                                    .WORD                   ;SYSTEM "" COUNT
 (2)  46BF  00  00                                    .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  46C1  00                                        .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  46C2  07                                        .BYTE   7               ;REQUEST CODE
 (1)  46C3  3A  9A  4F            13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1)  46C6  3D                    4.0          DCR     A               ;DOWNCOUNT
 (1)  46C7  32  9A  4F            13.0         STA     ITERA           ;SAVE COUNT
 (1)  46CA  F2  49  46            10.0         JP      TST07X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 10 - LOOP AMTIE TO AMTIE - MTA #3
46CD                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*LOOP AMTIE TO AMTIE TEST-MTA #3
46CD                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE DATA PATH FROM TU PORT #3 TO
                                ;*MTA #0 + #2 AND BACK VIA THE AMTIE LINES THROUGH THE MTA #3 AMTIE
                                ;*REGISTER TO THE TU PORT #3 AMTIE REGISTER.
46CD                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #3 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT PORT 3
                                ;*    CLEAR THE AMTIE DATA
                                ;*    BGNDO
                                ;*    :   ISSUE 7 BITS OF THE CONTENTS OF AMTIE DATA TO MTA #3 AMTIE DATA REGISTER
                                ;*    :   ISSUE THE LSB OF THE AMTIE DATA REGISTER TO MTA #3 AMTIE PARITY REGISTER
                                ;*    :   INPUT THE TU PORT #3 AMTIE DATA REGISTER
                                ;*    :   IF AMTIE DATA READ=AMTIE DATA
                                ;*    :   :   THEN-CONTINUE
                                ;*    :   :   ELSE-ERROR 24
                                ;*    :   ENDIF
                                ;*    :   INPUT THE AMTIE BIT 7 AND PARITY BIT
                                ;*    :   IF THE AMTIE BIT 7 AND PARITY READ-AMTIE BIT 7 AND PARITY WRITTEN
                                ;*    :   :   THEN-CONTINUE
                                ;*    :   :   ELSE-ERROR 25
                                ;*    :   ENDIF
                                ;*    :   CLEAR TU PORT #3 LOOP W/R BIT
                                ;*    :   INCREMENT THE AMTIE DATA
                                ;*    :   DO UNTIL THE AMTIE DATA=ZERO
                                ;*    ENDDO
                                ;*    CALL SUBROUTINE CLEAR
                                ;*ENDTST
46CD                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA1 MICRO TEST 10
                                ;*MTA1 MICRO ERROR 24
                                ;*MTA1-LOOP AMTIE TO AMTIE-MTA #3
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN TO MTA AMTIE NOT - DATA READ
                                ;*ACTUAL = NNNN
```

```
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA1 MICRO TEST 10
                                        ;*MTA1 MICRO ERROR 25
                                        ;*MTA1-LOOP AMTIE TO AMTIE-MTA #3
                                        ;*M8954, M8955'S
                                        ;*VALUE WRITTEN TO MTA AMTIE PTY. NOT = VALUE READ
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
46CD                                    $
 (1)                                    ; ************************************************************************
                                        TEST10: TESTX   @010            ;SET UP THE TEST NUMBER
 (1) 46CD  3E  08          7.0                       MVI     A,@010      ;DEFINE THE TEST NUMBER
 (1) 46CF  CD  03  28     18.0                       CALL    TSET        ;SETUP THE TEST
                                        ;%MTA1-LOOP AMTIE TO AMTIE-MTA #3
                                        ;&M8954, M8955'S

46D2  3A  CD  49         13.0  TST10X: LDA     UNITMP          ;GET THE UNIT MAP
46D5  E6  08              7.0         ANI     @010            ;TEST FOR UNIT 3?
46D7  CA  56  47         10.0         JZ      TEST11          ;NO-CHECK FOR TEST11
46DA  CD  A2  49         18.0         CALL    CLEAR           ;CLEAN UP THE PORTS
46DD  DB  E0             10.0         IN      INTSTA          ;GET SELECT STATUS
46DF  E6  80              7.0         ANI     BIT7
46E1  F6  03              7.0         ORI     3
46E3  D3  E0             10.0         OUT     MBSEL           ;SELECT PORT 3
46E5  21  CE  49         10.0         LXI     H,DATA          ;LOAD ADDRESS OF DATA
46E8  AF                  4.0         XRA     A               ;CLEAR THE ACCUMULATOR
46E9  77                  7.0         MOV     M,A             ;STORE IN DATA
46EA  3E  80              7.0  ALP31:  MVI     A,$80           ;SELECT MTA #3 REGISTER 0
46EC  D3  40             10.0         OUT     TCMD
46EE  7E                  7.0         MOV     A,M             ;GET THE AMTIE DATA
46EF  E6  01              7.0         ANI     $01             ;GET THE PSEUDO PARITY BIT
46F1  47                  4.0         MOV     B,A             ;SAVE IN B
46F2  7E                  7.0         MOV     A,M             ;GET THE AMTIE DATA-AGAIN
46F3  E6  80              7.0         ANI     $80             ;GET THE MSB
46F5  07                  4.0         RLC                     ;POSITION DATA FOR OUTPUT
46F6  07                  4.0         RLC
46F7  B0                  4.0         ORA     B               ;OR IN THE PSEUDO PARITY BIT
46F8  F6  04              7.0         ORI     @004            ;OR IN THE LOOP W/R BIT
46FA  D3  40             10.0         OUT     TCMD            ;LOAD MTA REGISTER 0
46FC  3E  82              7.0         MVI     A,$82           ;SELECT MTA #3 REGISTER 2
46FE  D3  40             10.0         OUT     TCMD
4700  7E                  7.0         MOV     A,M             ;GET THE AMTIE DATA
4701  47                  4.0         MOV     B,A             ;SAVE FOR DATA COMPARE
4702  E6  7F              7.0         ANI     $7F             ;REMOVE THE MSB
4704  D3  40             10.0         OUT     TCMD            ;LOAD MTA REGISTER 2
4706  DB  44             10.0         IN      TAMT            ;READ THE AMTIE REGISTER
4708  B8                  4.0         CMP     B               ;COMPARE ACTUAL AND EXP.
4709  CA  18  47         10.0         JZ      ALP32           ;EQUAL - CONTINUE
470C                                   ROUT    ADATA           ;STORE ACTUAL DATA
 (1) 470C  D3  94         10.0                       OUT     ADATA       ;WRITE AC INTO ADATA
 (1) 470E  7F              4.0                       MOV     A,A         ;RETRY LINK
470F  7E                  7.0         MOV     A,M             ;GET EXP. DATA
```

```
2219  4710                                          ROUT    EDATA           ;STORE EXP. DATA
 (1)  4710  D3  95          10.0                             OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4712  7F               4.0                             MOV     A,A             ;RETRY LINK
2220  4713                                          ERRB    ALP31,ALP32,0
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4713  CD  12  28      18.0                             CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0014                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4716  14                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4717  00                                          .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  4718  CD  15  28      18.0                 ALP32:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  471B  DA  EA  46      10.0                         JC      ALP31           ;LOOP ADDRESS IF LOOP SPECIFIED
2221                                        ;>DATA WRITTEN TO MTA AMTIE NOT = DATA READ
2222  471E  78                4.0          MOV     A,B             ;GET EXP. PTY.
2223  471F  E6  01             7.0          ANI     $01             ;REMOVE UNWANTED BITS
2224  4721  47                 4.0          MOV     B,A             ;RESTORE EXP. PTY.
2225  4722  DB  48            10.0          IN      PSTAT           ;GET ACTUAL PTY.
2226  4724  E6  01             7.0          ANI     $01             ;REMOVE UNWANTED BITS
2227  4726  B8                 4.0          CMP     B               ;EQUAL?
2228  4727  CA  36  47         10.0          JZ      A4LP02          ;YES
2229  472A                                  ROUT    ADATA           ;STORE ACTUAL PTY.
 (1)  472A  D3  94             10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  472C  7F                  4.0                 MOV     A,A             ;RETRY LINK
2230  472D  78                 4.0          MOV     A,B             ;GET EXP. PTY.
2231  472E                                  ROUT    EDATA           ;STORE EXP. PTY.
 (1)  472E  D3  95             10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4730  7F                  4.0                 MOV     A,A             ;RETRY LINK
2232  4731                                  ERRB    ALP31,A4LP02
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4731  CD  12  28      18.0                             CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0015                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4734  15                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4735  00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4736  CD  15  28      18.0                 A4LP02::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4739  DA  EA  46      10.0                         JC      ALP31           ;LOOP ADDRESS IF LOOP SPECIFIED
2233                                        ;>VALUE WRITTEN TO MTA AMTIE PTY. NOT - VALUE READ
2234
2235  473C  7E                 7.0          MOV     A,M             ;GET DATA
2236  473D  3C                 4.0          INR     A               ;UPDATE THE DATA BYTE
2237  473E  77                 7.0          MOV     M,A             ;SAVE UPDATED DATA
2238  473F  C2  EA  46         10.0          JNZ     ALP31           ;CONTINUE UNTIL DONE
2239
2240  4742                                  ENDTST  TST10X
 (1)                                              ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4742                                  REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4742  CD  06  28      18.0                             CALL    REQST
 (2)  4745  00                                          .BYTE                   ;DATA PATTERN NUMBER
 (2)  4746  00  00                                      .WORD                   ;SYSTEM '*' COUNT
 (2)  4748  00  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  474A  00                                          .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  474B  07                                          .BYTE   7               ;REQUEST CODE
 (1)  474C  3A  9A  4F      13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  474F  3D               4.0          DCR     A               ;DOWNCOUNT
 (1)  4750  32  9A  4F      13.0          STA     ITERA           ;SAVE COUNT
 (1)  4753  F2  D2  46      10.0          JP      TST10X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 11 - LOOP DATA WRITE/READ - MTA #0
4756                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP DATA WRITE TO DATA READ-MTA #0
4756                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE READ DATA PATH FROM MTA #0 TO TU
                                ;*PORT #0 USING THE LOOP DATA WRITE TO DATA READ CAPABILITY OF THE MTA
                                ;*BOARD.
4756                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #0 NOT SELECTED BY USER
                                ;*    :   THEN - NEXT TEST
                                ;*    :   ELSE - CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT TU PORT 0
                                ;*    CLEAR THE DATA BYTE
                                ;*    SET LOOP DATA WRITE/READ MTA 0
                                ;*    BGNDO
                                ;*    :   SELECT A NON-EXISTANT MTA REGISTER
                                ;*    :   OUTPUT THE DATA TO TU PORT #0 COMMAND BYTE
                                ;*    :   INPUT THE DATA FROM TU PORT #0 DATA BYTE
                                ;*    :   IF INPUT DATA = OUTPUT DATA
                                ;*    :   :   THEN - CONTINUE
                                ;*    :   :   ELSE-ERROR 26
                                ;*    :   ENDIF
                                ;*    :   INPUT THE DATA PARITY FROM TU PORT #0
                                ;*    :   IF THE INPUT PARITY IS ODD (FOR THE CURRENT DATA)
                                ;*    :   :   THEN - CONTINUE
                                ;*    :   :   ELSE-ERROR 27
                                ;*    :   ENDIF
                                ;*    :   INCREMENT THE DATA BYTE
                                ;*    :   DO UNTIL THE DATA = 0
                                ;*    ENDDO
                                ;*ENDTST
4756                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA1 MICRO TEST 11
                                ;*MTA1 MICRO ERROR 26
                                ;*MTA1-LOOP DATA WRITE TO READ-MTA #0
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN NOT = DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
```

```
                                        ;*
                                        ;*MTA1 MICRO TEST 11
                                        ;*MTA1 MICRO ERROR 27
                                        ;*MTA1-LOOP DATA WRITE TO READ-MTA #0
                                        ;*M8954, M8955'S
                                        ;*DATA PARITY READ NOT = PARITY EXPECTED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
 2293  4756                             S
  (1)                                   ; ************************************************************************************
 2294  4756                             TEST11: TESTX   @11             ;SET UP THE TEST NUMBER
  (1)  4756  3E  09            7.0                 MVI     A,@11      ;DEFINE THE TEST NUMBER
  (1)  4758  CD  03  28       18.0                 CALL    TSET       ;SETUP THE TEST
 2295                                   ;%MTA1-LOOP DATA WRITE TO READ-MTA #0
 2296                                   ;&M8954, M8955'S
 2297
 2298  475B  3A  CD  49       13.0  TST11X: LDA     UNITMP             ;GET THE UNIT MAP
 2299  475E  E6  01            7.0         ANI     @01                ;TEST FOR UNIT 0
 2300  4760  CA  E6  47       10.0         JZ      TEST12             ;NO-CHECK FOR TEST 12
 2301  4763  CD  A2  49       18.0         CALL    CLEAR              ;CLEAR THE DATA BYTE
 2302  4766  DB  E0           10.0         IN      INTSTA             ;GET SELECT STATUS
 2303  4768  E6  80            7.0         ANI     BIT7
 2304  476A  D3  E0           10.0         OUT     MBSEL              ;SELECT PORT 0
 2305  476C  AF                4.0         XRA     A                  ;
 2306  476D  32  CE  49       13.0         STA     DATA               ;SAVE THE PROGRAM DATA
 2307  4770                             ROUT    R05H                ;CLEAR UPPER BYTE FOR REPORTING
  (1)  4770  D3  8B           10.0                 OUT     R05H               ;WRITE AC INTO R05H
  (1)  4772  7F                4.0                 MOV     A,A                ;RETRY LINK
 2308  4773  3E  80            7.0         MVI     A,$80              ;SELECT MTA #0 REGISTER #0
 2309  4775  D3  40           10.0         OUT     TCMD               ;
 2310  4777  3E  04            7.0         MVI     A,$04              ;SET MTA #0 LOOP WRITE TO READ
 2311  4779  D3  40           10.0         OUT     TCMD               ;
 2312  477B  3E  87            7.0  DOLP01: MVI     A,$87              ;SELECT NON-X-REGISTER
 2313  477D  D3  40           10.0         OUT     TCMD
 2314  477F  3A  CE  49       13.0         LDA     DATA               ;GET THE DATA BYTE
 2315  4782  D3  40           10.0         OUT     TCMD               ;WRITE TO THE PORT
 2316  4784                             ROUT    R05L                ;AND AGAIN
  (1)  4784  D3  8A           10.0                 OUT     R05L               ;WRITE AC INTO R05L
  (1)  4786  7F                4.0                 MOV     A,A                ;RETRY LINK
 2317  4787                             ROUT    EDATA               ;SAVE IN THE CAS
  (1)  4787  D3  95           10.0                 OUT     EDATA              ;WRITE AC INTO EDATA
  (1)  4789  7F                4.0                 MOV     A,A                ;RETRY LINK
 2318  478A  47                4.0         MOV     B,A                ;SAVE IN REGISTER B
 2319  478B  DB  4C           10.0         IN      PRDD               ;GET THE ACTUAL DATA
 2320  478D                             ROUT    ADATA               ;SAVE THE ACTUAL DATA IN CAS
  (1)  478D  D3  94           10.0                 OUT     ADATA              ;WRITE AC INTO ADATA
  (1)  478F  7F                4.0                 MOV     A,A                ;RETRY LINK
 2321  4790  B8                4.0         CMP     B                  ;ACTUAL DATA = EXPECTED DATA?
 2322  4791  CA  99  47       10.0         JZ      DOLP02             ;YES - CONTINUE
```

```
2324  4794                                        ERRB    DOLP01,DOLP02,0
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4794  CD  12  28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0016                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4797  16                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4798  00                                            .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  4799  CD  15  28        18.0                DOLP02::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  479C  DA  7B  47        10.0                        JC      DOLP01          ;LOOP ADDRESS IF LOOP SPECIFIED
2325                                    ;>DATA WRITTEN NOT = DATA READ
2326  479F  3A  CE  49        13.0            LDA     DATA            ;GET THE EXPECTED DATA
2327  47A2  A7                  4.0            ANA     A
2328  47A3  EA  AA  47        10.0            JPE     DOLPP1          ;SET EXPECTED PARITY
2329  47A6  AF                  4.0    DOLPP0: XRA     A               ;EXPECTED PARITY =0
2330  47A7  C3  AC  47        10.0            JMP     DOLPPC
2331  47AA  3E  01              7.0    DOLPP1: MVI     A,1             ;EXPECTED PARITY =1
2332  47AC                             DOLPPC: ROUT    EDATA           ;SAVE IT
 (1)  47AC  D3  95            10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  47AE  7F                 4.0                    MOV     A,A             ;RETRY LINK
2333  47AF  47                 4.0            MOV     B,A             ;SAVE IN REGISTER B
2334  47B0  DB  48            10.0            IN      PSTAT           ;GET THE ACTUAL PARITY
2335  47B2  E6  02             7.0            ANI     P.RPST          ;REMOVE ONLY THE PARITY BIT
2336  47B4  0F                 4.0            RRC                     ;POSITION THE BIT
2337  47B5                                    ROUT    ADATA           ;STORE IN CAS
 (1)  47B5  D3  94            10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  47B7  7F                 4.0                    MOV     A,A             ;RETRY LINK
2338  47B8  B8                 4.0            CMP     B               ;ACTUAL PARITY = EXPECTED PARITY?
2339  47B9  CA  C1  47        10.0            JZ      DOLP03          ;YES - CONTINUE
2340  47BC                                    ERRB    DOLP01,DOLP03,1
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  47BC  CD  12  28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0017                                           MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47BF  17                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47C0  01                                            .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  47C1  CD  15  28        18.0                DOLP03::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47C4  DA  7B  47        10.0                        JC      DOLP01          ;LOOP ADDRESS IF LOOP SPECIFIED
2341                                    ;>DATA PARITY READ NOT = PARITY EXPECTED
2342  47C7  3A  CE  49        13.0            LDA     DATA            ;GET THE DATA BYTE
2343  47CA  3C                  4.0            INR     A               ;INCREMENT
2344  47CB  32  CE  49        13.0            STA     DATA            ;SAVE THE NEW DATA
2345  47CE  A7                  4.0            ANA     A               ;SET THE CONDITION BITS
2346  47CF  C2  7B  47        10.0            JNZ     DOLP01
2347  47D2                                    ENDTST  TST11X
 (1)                                      ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  47D2                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  47D2  CD  06  28        18.0                        CALL    REQST
 (2)  47D5  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2)  47D6  00  00                                        .WORD                   ;SYSTEM "" COUNT
 (2)  47D8  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  47DA  00                                            .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  47DB  07                                            .BYTE   7               ;REQUEST CODE
 (1)  47DC  3A  9A  4F        13.0            LDA     ITERA           ;GET ITERATION COUNT
 (1)  47DF  3D                 4.0            DCR     A               ;DOWNCOUNT
 (1)  47E0  32  9A  4F        13.0            STA     ITERA           ;SAVE COUNT
 (1)  47E3  F2  5B  47        10.0            JP      TST11X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 12 - LOOP DATA WRITE/READ - MTA #1
47E6                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*LOOP DATA WRITE TO DATA READ-MTA #1
47E6                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE READ DATA PATH FROM MTA #1 TO TU
                                ;*PORT #1 USING THE LOOP DATA WRITE TO DATA READ CAPABILITY OF THE MTA
                                ;*BOARD.
47E6                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #1 NOT SELECTED BY USER
                                ;*     :   THEN - NEXT TEST
                                ;*     :   ELSE - CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT TU PORT 1
                                ;*     CLEAR THE DATA BYTE
                                ;*     SET LOOP DATA WRITE/READ MTA 1
                                ;*     BGNDO
                                ;*     :   SELECT A NON-EXISTANT MTA REGISTER
                                ;*     :   OUTPUT THE DATA TO TU PORT #1 COMMAND BYTE
                                ;*     :   INPUT THE DATA FROM TU PORT #1 DATA BYTE
                                ;*     :   IF INPUT DATA = OUTPUT DATA
                                ;*     :   :   THEN - CONTINUE
                                ;*     :   :   ELSE-ERROR 30
                                ;*     :   ENDIF
                                ;*     :   INPUT THE DATA PARITY FROM TU PORT #1
                                ;*     :   IF THE INPUT PARITY IS ODD (FOR THE CURRENT DATA)
                                ;*     :   :   THEN - CONTINUE
                                ;*     :   :   ELSE-ERROR 31
                                ;*     :   ENDIF
                                ;*     :   INCREMENT THE DATA BYTE
                                ;*     :   DO UNTIL THE DATA = 0
                                ;*     ENDDO
                                ;*ENDTST
47E6                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA1 MICRO TEST 12
                                ;*MTA' MICRO ERROR 30
                                ;*MTA1-LOOP DATA WRITE TO READ-MTA #1
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN NOT = DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED - NNNN
```

```
                                        ;*
                                        ;*MTA1 MICRO TEST 12
                                        ;*MTA1 MICRO ERROR 31
                                        ;*MTA1-LOOP DATA WRITE TO READ-MTA #1
                                        ;*M8954, M8955'S
                                        ;*DATA PARITY READ NOT = PARITY EXPECTED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
2399  47E6                              $
 (1)                                    ;*****************************************************************************
2400  47E6                              TEST12: TESTX   @12             ;SET UP THE TEST NUMBER
 (1)  47E6  3E  0A           7.0                    MVI     A,@12       ;DEFINE THE TEST NUMBER
 (1)  47E8  CD  03  28      18.0                    CALL    TSET        ;SETUP THE TEST
2401                                    ;%MTA1-LOOP DATA WRITE TO READ-MTA #1
2402                                    ;&M8954, M8955'S
2403
2404  47EB  3A  CD  49      13.0  TST12X: LDA     UNITMP              ;GET THE UNIT MAP
2405  47EE  E6  02           7.0          ANI     @02                 ;TEST FOR UNIT 1
2406  47F0  CA  78  48      10.0          JZ      TEST13              ;NO-CHECK FOR TEST 13
2407  47F3  CD  A2  49      18.0          CALL    CLEAR               ;CLEAR THE DATA BYTE
2408  47F6  DB  E0          10.0          IN      INTSTA              ;GET SELECT STATUS
2409  47F8  E6  80           7.0          ANI     BIT7
2410  47FA  F6  01           7.0          ORI     1
2411  47FC  D3  EC          10.0          OUT     MBSEL               ;SELECT PORT 1
2412  47FE  AF               4.0          XRA     A                   ;
2413  47FF  32  CE  49      13.0          STA     DATA                ;SAVE THE PROGRAM DATA
2414  4802                                ROUT    R05H
 (1)  4802  D3  8B          10.0                  OUT     R05H                ;WRITE AC INTO R05H
 (1)  4804  7F               4.0                  MOV     A,A                 ;RETRY LINK
2415  4805  3E  80           7.0          MVI     A,$80               ;SELECT MTA #1 REGISTER #0
2416  4807  D3  40          10.0          OUT     TCMD                ;
2417  4809  3E  04           7.0          MVI     A,$04               ;SET MTA #1 LOOP WRITE TO READ
2418  480B  D3  40          10.0          OUT     TCMD                ;
2419  480D  3E  87           7.0  D1LP01: MVI     A,$87               ;SELECT NON-X-REG
2420  480F  D3  40          10.0          OUT     TCMD
2421  4811  3A  CE  49      13.0          LDA     DATA                ;GET THE DATA BYTE
2422  4814  D3  40          10.0          OUT     TCMD                ;WRITE TO THE PORT
2423  4816                                ROUT    R05L                ;SAVE FOR REPORTING
 (1)  4816  D3  8A          10.0                  OUT     R05L                ;WRITE AC INTO R05L
 (1)  4818  7F               4.0                  MOV     A,A                 ;RETRY LINK
2424  4819                                ROUT    EDATA               ;SAVE IN THE CAS
 (1)  4819  D3  95          10.0                  OUT     EDATA               ;WRITE AC INTO EDATA
 (1)  481B  7F               4.0                  MOV     A,A                 ;RETRY LINK
2425  481C  47               4.0          MOV     B,A                 ;SAVE IN REGISTER B
2426  481D  DB  4C          10.0          IN      PRDD                ;GET THE ACTUAL DATA
2427  481F                                ROUT    ADATA               ;SAVE THE ACTUAL DATA IN CAS
 (1)  481F  D3  94          10.0                  OUT     ADATA               ;WRITE AC INTO ADATA
 (1)  4821  7F               4.0                  MOV     A,A                 ;RETRY LINK
2428  4822  B8               4.0          CMP     B                   ;ACTUAL DATA = EXPECTED DATA?
2429  4823  CA  2B  48      10.0          JZ      D1LP02              ;YES - CONTINUE
```

```
2431  4826                                        ERRB    D1LP01,D1LP02,0
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4826  CD  12  28      18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0018                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4829  18                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  482A  00                                            .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  482B  CD  15  28      18.0                  D1LP02::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  482E  DA  0D  48      10.0                          JC      D1LP01          ;LOOP ADDRESS IF LOOP SPECIFIED
2432                                        ;>DATA WRITTEN NOT = DATA READ
2433  4831  3A  CE  49      13.0                  LDA     DATA                    ;GET THE EXPECTED DATA
2434  4834  A7                 4.0                  ANA     A
2435  4835  EA  3C  48      10.0                  JPE     D1LPP1                  ;SET EXPECTED PARITY
2436  4838  AF                 4.0          D1LPP0: XRA     A                       ;EXPECTED PARITY =0
2437  4839  C3  3E  48      10.0                  JMP     D1LPPC
2438  483C  3E  01               7.0          D1LPP1: MVI     A,1                     ;EXPECTED PARITY =1
2439  483E                                    D1LPPC: ROUT    EDATA                   ;SAVE IT
 (1)  483E  D3  95          10.0                          OUT     EDATA                   ;WRITE AC INTO EDATA
 (1)  4840  7F                 4.0                        MOV     A,A                     ;RETRY LINK
2440  4841  47                 4.0                  MOV     B,A                     ;SAVE IN REGISTER B
2441  4842  DB  48          10.0                  IN      PSTAT                   ;GET THE ACTUAL PARITY
2442  4844  E6  02               7.0              ANI     P.RPST                  ;REMOVE ONLY THE PARITY BIT
2443  4846  0F                 4.0                  RRC                             ;POSITION THE BIT
2444  4847                                        ROUT    ADATA                   ;STORE IN CAS
 (1)  4847  D3  94          10.0                          OUT     ADATA                   ;WRITE AC INTO ADATA
 (1)  4849  7F                 4.0                        MOV     A,A                     ;RETRY LINK
2445  484A  B8                 4.0                  CMP     B                       ;ACTUAL PARITY = EXPECTED PARITY?
2446  484B  CA  53  48      10.0                  JZ      D1LP03                  ;YES - CONTINUE
2447  484E                                        ERRB    D1LP01,D1LP03,1
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  484E  CD  12  28      18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0019                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4851  19                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4852  01                                            .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4853  CD  15  28      18.0                  D1LP03::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4856  DA  0D  48      10.0                          JC      D1LP01          ;LOOP ADDRESS IF LOOP SPECIFIED
2448                                        ;>DATA PARITY READ NOT = PARITY EXPECTED
2449  4859  3A  CE  49      13.0                  LDA     DATA                    ;GET THE DATA BYTE
2450  485C  3C                  4.0                  INR     A                       ;INCREMENT
2451  485D  32  CE  49      13.0                  STA     DATA                    ;SAVE THE NEW DATA
2452  4860  A7                 4.0                  ANA     A                       ;SET THE CONDITION BITS
2453  4861  C2  0D  48      10.0                  JNZ     D1LP01
2454  4864                                        ENDTST  TST12X
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4864                                          REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4864  CD  06  28      18.0                          CALL    REQST
 (2)  4867  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2)  4868  00  00                                        .WORD                   ;SYSTEM "" COUNT
 (2)  486A  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  486C  00                                            .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  486D  07                                            .BYTE   7               ;REQUEST CODE
 (1)  486E  3A  9A  4F      13.0                  LDA     ITERA                   ;GET ITERATION COUNT
 (1)  4871  3D                 4.0                  DCR     A                       ;DOWNCOUNT
 (1)  4872  32  9A  4F      13.0                  STA     ITERA                   ;SAVE COUNT
 (1)  4875  F2  EB  47      10.0                  JP      TST12X                  ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 13 - LOOP DATA WRITE/READ - MTA #2
4878                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*LOOP DATA WRITE TO DATA READ-MTA #2
4878                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE READ DATA PATH FROM MTA #2 TO TU
                                ;*PORT #2 USING THE LOOP DATA WRITE TO DATA READ CAPABILITY OF THE MTA
                                ;*BOARD.
4878                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #2 NOT SELECTED BY USER
                                ;*    :   THEN - NEXT TEST
                                ;*    :   ELSE - CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT TU PORT 2
                                ;*    CLEAR THE DATA BYTE
                                ;*    SET LOOP DATA WRITE/READ MTA 2
                                ;*    BGNDO
                                ;*    :   SELECT A NON-EXISTANT MTA REGISTER
                                ;*    :   OUTPUT THE DATA TO TU PORT #2 COMMAND BYTE
                                ;*    :   INPUT THE DATA FROM TU PORT #2 DATA BYTE
                                ;*    :   IF INPUT DATA = OUTPUT DATA
                                ;*    :   :   THEN - CONTINUE
                                ;*    :   :   ELSE-ERROR 32
                                ;*    :   ENDIF
                                ;*    :   INPUT THE DATA PARITY FROM TU PORT #2
                                ;*    :   IF THE INPUT PARITY IS ODD (FOR THE CURRENT DATA)
                                ;*    :   :   THEN - CONTINUE
                                ;*    :   :   ELSE-ERROR 33
                                ;*    :   ENDIF
                                ;*    :   INCREMENT THE DATA BYTE
                                ;*    :   DO UNTIL THE DATA = 0
                                ;*    ENDDO
                                ;*ENDTST
4878                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA1 MICRO TEST 13
                                ;*MTA1 MICRO ERROR 32
                                ;*MTA1-LOOP DATA WRITE TO READ-MTA #2
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN NOT = DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
```

```
                                        ;*
                                        ;*MTA1 MICRO TEST 13
                                        ;*MTA1 MICRO ERROR 33
                                        ;*MTA1-LOOP DATA WRITE TO READ-MTA #2
                                        ;*M8954, M8955'S
                                        ;*DATA PARITY READ NOT = PARITY EXPECTED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4878                                    S
 (1)                                    ; ************************************************************************
4878                                    TEST13: TESTX   @13             ;SET UP THE TEST NUMBER
 (1) 4878  3E  0B           7.0                         MVI     A,@13       ;DEFINE THE TEST NUMBER
 (1) 487A  CD  03  28      18.0                         CALL    TSET        ;SETUP THE TEST
                                        ;%MTA1-LOOP DATA WRITE TO READ-MTA #2
                                        ;&M8954, M8955'S

487D  3A  CD  49          13.0  TST13X: LDA     UNITMP          ;GET THE UNIT MAP
4880  E6  04                7.0        ANI     @04             ;TEST FOR UNIT 2
4882  CA  0A  49           10.0        JZ      TEST14          ;NO-CHECK FOR TEST 14
4885  CD  A2  49           18.0        CALL    CLEAR           ;CLEAR THE DATA BYTE
4888  DB  E0               10.0        IN      INTSTA          ;GET SELECT STATUS
488A  E6  80                7.0        ANI     BIT7
488C  F6  02                7.0        ORI     2
488E  D3  E0               10.0        OUT     MBSEL           ;SELECT PORT 2
4890  AF                    4.0        XRA     A               ;
4891  32  CE  49           13.0        STA     DATA            ;SAVE THE PROGRAM DATA
4894                                    ROUT    RO5H
 (1) 4894  D3  8B           10.0                OUT     RO5H            ;WRITE AC INTO RO5H
 (1) 4896  7F                4.0                MOV     A,A             ;RETRY LINK
4897  3E  80                7.0        MVI     A,$80           ;SELECT MTA #2 REGISTER #0
4899  D3  40               10.0        OUT     TCMD            ;
489B  3E  04                7.0        MVI     A,$04           ;SET MTA #2 LOOP WRITE TO READ
489D  D3  40               10.0        OUT     TCMD            ;
489F  3E  87                7.0  D2LP01: MVI     A,$87           ;GET A NON-X-REGISTER
48A1  D3  40               10.0        OUT     TCMD            ;SELECT IT
48A3  3A  CE  49           13.0        LDA     DATA            ;GET THE DATA BYTE
48A6  D3  40               10.0        OUT     TCMD            ;WRITE TO THE PORT
48A8                                    ROUT    RO5L            ;SAVE FOR REPORTING
 (1) 48A8  D3  8A           10.0                OUT     RO5L            ;WRITE AC INTO RO5L
 (1) 48AA  7F                4.0                MOV     A,A             ;RETRY LINK
48AB                                    ROUT    EDATA           ;SAVE IN THE CAS
 (1) 48AB  D3  95           10.0                OUT     EDATA           ;WRITE AC INTO EDATA
 (1) 48AD  7F                4.0                MOV     A,A             ;RETRY LINK
48AE  47                    4.0        MOV     B,A             ;SAVE IN REGISTER B
48AF  DB  4C               10.0        IN      PRDD            ;GET THE ACTUAL DATA
48B1                                    ROUT    ADATA           ;SAVE THE ACTUAL DATA IN CAS
 (1) 48B1  D3  94           10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1) 48B3  7F                4.0                MOV     A,A             ;RETRY LINK
48B4  B8                    4.0        CMP     B               ;ACTUAL DATA = EXPECTED DATA?
48B5  CA  BD  48           10.0        JZ      D2LP02          ;YES - CONTINUE
```

```
2538  48B8                                        ERRB    D2LP01,D2LP02,0
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  48B8  CD  12  28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       001A                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48BB  1A                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48BC  00                                           .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  48BD  CD  15  28        18.0               D2LP02::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48C0  DA  9F  48        10.0                        JC      D2LP01          ;LOOP ADDRESS IF LOOP SPECIFIED
2539                                    ;>DATA WRITTEN NOT = DATA READ
2540  48C3  3A  CE  49        13.0              LDA     DATA                    ;GET THE EXPECTED DATA
2541  48C6  A7                   4.0              ANA     A
2542  48C7  EA  CE  48        10.0              JPE     D2LPP1                  ;SET EXPECTED PARITY
2543  48CA  AF                   4.0  D2LPP0.    XRA     A                       ;FXPECTED PARITY =0
2544  48CB  C3  D0  48        10.0              JMP     D2LPPC
2545  48CE  3E  01               7.0  D2LPP1:    MVI     A,1                     ;EXPECTED PARITY =1
2546  48D0                            D2LPPC:    ROUT    EDATA                   ;SAVE IT
 (1)  48D0  D3  95             10.0                        OUT     EDATA                   ;WRITE AC INTO EDATA
 (1)  48D2  7F                  4.0                        MOV     A,A                     ;RETRY LINK
2547  48D3  47                   4.0              MOV     B,A                     ;SAVE IN REGISTER B
2548  48D4  DB  48             10.0              IN      PSTAT                   ;GET THE ACTUAL PARITY
2549  48D6  E6  02               7.0              ANI     P.RPST                  ;REMOVE ONLY THE PARITY BIT
2550  48D8  0F                   4.0              RRC                             ;POSITION THE BIT
2551  48D9                                        ROUT    ADATA                   ;STORE IN CAS
 (1)  48D9  D3  94             10.0                        OUT     ADATA                   ;WRITE AC INTO ADATA
 (1)  48DB  7F                  4.0                        MOV     A,A                     ;RETRY LINK
2552  48DC  B8                   4.0              CMP     B                       ;ACTUAL PARITY = EXPECTED PARITY?
2553  48DD  CA  E5  48         10.0              JZ      D2LP03                  ;YES - CONTINUE
2554  48E0                                        ERRB    D2LP01,D2LP03,1
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  48E0  CD  12  28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       001B                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48E3  1B                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48E4  01                                           .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  48E5  CD  15  28        18.0               D2LP03::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48E8  DA  9F  48        10.0                        JC      D2LP01          ;LOOP ADDRESS IF LOOP SPECIFIED
2555                                    ;>DATA PARITY READ NOT = PARITY EXPECTED
2556  48EB  3A  CE  49        13.0              LDA     DATA                    ;GET THE DATA BYTE
2557  48EE  3C                   4.0              INR     A                       ;INCREMENT
2558  48EF  32  CE  49        13.0              STA     DATA                    ;SAVE THE NEW DATA
2559  48F2  A7                   4.0              ANA     A                       ;SET THE CONDITION BITS
2560  48F3  C2  9F  48         10.0              JNZ     D2LP01
2561  48F6                                      ENDTST  TST13X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  48F6                                      REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  48F6  CD  06  28        18.0                        CALL    REQST
 (2)  48F9  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  48FA  00  00                                       .WORD                   ;SYSTEM "" COUNT
 (2)  48FC  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  48FE  00                                           .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  48FF  07                                           .BYTE   7               ;REQUEST CODE
 (1)  4900  3A  9A  4F        13.0              LDA     ITERA                   ;GET ITERATION COUNT
 (1)  4903  3D                   4.0              DCR     A                       ;DOWNCOUNT
 (1)  4904  32  9A  4F        13.0              STA     ITERA                   ;SAVE COUNT
 (1)  4907  F2  7D  48         10.0              JP      TST13X                  ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 14 - LOOP DATA WRITE/READ - MTA #3
490A                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*LOOP DATA WRITE TO DATA READ-MTA #3
490A                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE READ DATA PATH FROM MTA #3 TO TU
                                ;*PORT #3 USING THE LOOP DATA WRITE TO DATA READ CAPABILITY OF THE MTA
                                ;*BOARD.
490A                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDU
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #3 NOT SELECTED BY USER
                                ;*    :   THEN - NEXT TEST
                                ;*    :   ELSE - CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT TU PORT 3
                                ;*    CLEAR THE DATA BYTE
                                ;*    SET LOOP DATA WRITE/READ MTA 3
                                ;*    BGNDO
                                ;*    :   SELECT A NON-EXISTANT MTA REGISTER
                                ;*    :   OUTPUT THE DATA TO TU PORT #3 COMMAND BYTE
                                ;*    :   INPUT THE DATA FROM TU PORT #3 DATA BYTE
                                ;*    :   IF INPUT DATA = OUTPUT DATA
                                ;*    :   :   THEN - CONTINUE
                                ;*    :   :   ELSE-ERROR 34
                                ;*    :   ENDIF
                                ;*    :   INPUT THE DATA PARITY FROM TU PORT #3
                                ;*    :   IF THE INPUT PARITY IS ODD (FOR THE CURRENT DATA)
                                ;*    :   :   THEN - CONTINUE
                                ;*    :   :   ELSE-ERROR 35
                                ;*    :   ENDIF
                                ;*    :   INCREMENT THE DATA BYTE
                                ;*    :   DO UNTIL THE DATA = 0
                                ;*    ENDDO
                                ;*ENDTST
490A                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA1 MICRO TEST 14
                                ;*MTA1 MICRO ERROR 34
                                ;*MTA1-LOOP DATA WRITE TO READ-MTA #3
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN NOT = DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
```

```
2605                                    ;*
2606                                    ;*MTA1 MICRO TEST 14
2607                                    ;*MTA1 MICRO ERROR 35
2608                                    ;*MTA1-LOOP DATA WRITE TO READ-MTA #3
2609                                    ;*M8954, M8955'S
2610                                    ;*DATA PARITY READ NOT = PARITY EXPECTED
2611                                    ;*ACTUAL = NNNN
2612                                    ;*EXPECTED = NNNN
2613  490A                              S
 (1)                                    ; ***************************************************************************
2614  490A                              TEST14: TESTX   @14             ;SET UP THE TEST NUMBER
 (1)  490A  3E  0C            7.0                 MVI     A,@14           ;DEFINE THE TEST NUMBER
 (1)  490C  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
2615                                    ;%MTA1-LOOP DATA WRITE TO READ-MTA #3
2616                                    ;&M8954, M8955'S
2617
2618  490F  3A  CD  49         13.0     TST14X: LDA     UNITMP          ;GET THE UNIT MAP
2619  4912  E6  08               7.0             ANI     @10             ;TEST FOR UNIT 3
2620  4914  CA  9C  49          10.0             JZ      EXIT            ;NO-EXIT THE PROGRAM
2621  4917  CD  A2  49          18.0             CALL    CLEAR           ;CLEAR THE DATA BYTE
2622  491A  DB  E0               10.0             IN      INTSTA          ;GET SELECT STATUS
2623  491C  E6  80               7.0             ANI     BIT7
2624  491E  F6  03               7.0             ORI     3
2625  4920  D3  E0              10.0             OUT     MBSEL           ;SELECT PORT 3
2626  4922  AF                   4.0             XRA     A               ;
2627  4923  32  CE  49          13.0             STA     DATA            ;SAVE THE PROGRAM DATA
2628  4926                              ROUT    RO5H
 (1)  4926  D3  8B              10.0                     OUT     RO5H            ;WRITE AC INTO RO5H
 (1)  4928  7F                   4.0                     MOV     A,A             ;RETRY LINK
2629  4929  3E  80               7.0             MVI     A,$80           ;SELECT MTA #3 REGISTER #0
2630  492B  D3  40              10.0             OUT     TCMD            ;
2631  492D  3E  04               7.0             MVI     A,$04           ;SET MTA #3 LOOP WRITE TO READ
2632  492F  D3  40              10.0             OUT     TCMD            ;
2633  4931  3E  87               7.0     D3LP01: MVI     A,$87           ;SELECT NON-X-REG
2634  4933  D3  40              10.0             OUT     TCMD
2635  4935  3A  CE  49          13.0             LDA     DATA            ;GET THE DATA BYTE
2636  4938  D3  40              10.0             OUT     TCMD            ;WRITE TO THE PORT
2637  493A                              ROUT    RO5L            ;SAVE FOR REPORTING
 (1)  493A  D3  8A              10.0                     OUT     RO5L            ;WRITE AC INTO RO5L
 (1)  493C  7F                   4.0                     MOV     A,A             ;RETRY LINK
2638  493D                              ROUT    EDATA           ;SAVE IN THE CAS
 (1)  493D  D3  95              10.0                     OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  493F  7F                   4.0                     MOV     A,A             ;RETRY LINK
2639  4940  47                   4.0             MOV     B,A             ;SAVE IN REGISTER B
2640  4941  DB  4C              10.0             IN      PRDD            ;GET THE ACTUAL DATA
2641  4943                              ROUT    ADATA           ;SAVE THE ACTUAL DATA IN CAS
 (1)  4943  D3  94              10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4945  7F                   4.0                     MOV     A,A             ;RETRY LINK
2642  4946  B8                   4.0             CMP     B               ;ACTUAL DATA = EXPECTED DATA?
2643  4947  CA  4F  49          10.0             JZ      D3LP02          ;YES - CONTINUE
```

```
2645  494A                                    ERRB    D3LP01,D3LP02,0
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  494A  CD  12  28        18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        001C                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  494D  1C                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  494E  00                                        .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  494F  CD  15  28        18.0            D3LP02::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4952  DA  31  49        10.0                    JC      D3LP01          ;LOOP ADDRESS IF LOOP SPECIFIED
2646                            ;>DATA WRITTEN NOT = DATA READ
2647
2648  4955  3A  CE  49        13.0            LDA     DATA            ;GET THE EXPECTED DATA
2649  4958  A7                   4.0            ANA     A
2650  4959  EA  60  49        10.0            JPE     D3LPP1          ;SET EXPECTED PARITY
2651  495C  AF                   4.0    D3LPP0: XRA     A               ;EXPECTED PARITY =0
2652  495D  C3  62  49        10.0            JMP     D3LPPC
2653  4960  3E  01             7.0    D3LPP1: MVI     A,1             ;EXPECTED PARITY =1
2654  4962                           D3LPPC: ROUT    EDATA           ;SAVE IT
 (1)  4962  D3  95             10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4964  7F                   4.0                    MOV     A,A             ;RETRY LINK
2655  4965  47                   4.0            MOV     B,A             ;SAVE IN REGISTER B
2656  4966  DB  48             10.0            IN      PSTAT           ;GET THE ACTUAL PARITY
2657  4968  E6  02             7.0            ANI     P.RPST          ;REMOVE ONLY THE PARITY BIT
2658  496A  0F                   4.0            RRC                     ;POSITION THE BIT
2659  496B                                    ROUT    ADATA           ;STORE IN CAS
 (1)  496B  D3  94             10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  496D  7F                   4.0                    MOV     A,A             ;RETRY LINK
2660  496E  B8                   4.0            CMP     B               ;ACTUAL PARITY = EXPECTED PARITY?
2661  496F  CA  77  49        10.0            JZ      D3LP03          ;YES - CONTINUE
2662  4972                                    ERRB    D3LP01,D3LP03,1
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4972  CD  12  28        18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        001D                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4975  1D                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4976  01                                        .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4977  CD  15  28        18.0            D3LP03::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  497A  DA  31  49        10.0                    JC      D3LP01          ;LOOP ADDRESS IF LOOP SPECIFIED
2663                            ;>DATA PARITY READ NOT = PARITY EXPECTED
2664  497D  3A  CE  49        13.0            LDA     DATA            ;GET THE DATA BYTE
2665  4980  3C                   4.0            INR     A               ;INCREMENT
2666  4981  32  CE  49        13.0            STA     DATA            ;SAVE THE NEW DATA
2667  4984  A7                   4.0            ANA     A               ;SET THE CONDITION BITS
2668  4985  C2  31  49        10.0            JNZ     D3LP01
```

```
2670  4988                                        ENDTST  TST14X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4988                                       REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4988  CD  06  28        18.0                                 CALL    REQST
 (2)  498B  00                                                     .BYTE                   ;DATA PATTERN NUMBER
 (2)  498C  00  00                                                 .WORD                   ;SYSTEM '"' COUNT
 (2)  498E  00  00                                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4990  00                                                     .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4991  07                                                     .BYTE   7               ;REQUEST CODE
 (1)  4992  3A  9A  4F        13.0               LDA     ITERA           ;GET ITERATION COUNT
 (1)  4995  3D                 4.0               DCR     A               ;DOWNCOUNT
 (1)  4996  32  9A  4F        13.0               STA     ITERA           ;SAVE COUNT
 (1)  4999  F2  0F  49        10.0               JP      TST14X          ;DO TEST UNTIL TILL = 0
2671  499C  CD  A2  49        18.0  EXIT:        CALL    CLEAR           ;LEAVE THE PORTS CLEAN
2672  499F  C3  18  28        10.0               JMP     TSTEND          ;END OF TEST
```

```
                                .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
 49A2                           SSUB
  (1)                           ; ************************************************************************
  (1)                           ;*SUBROUTINE TITLE
  (1)                           ;*---------- -----
                                ;*CLEAR ALL TU PORTS
 49A2                           SD
  (1)                           ; ************************************************************************
  (1)                           ;*DESCRIPTION
  (1)                           ;*-----------
                                ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                ;*AND LOOP MODES.
 49A2                           SP
  (1)                           ; ************************************************************************
  (1)                           ;*PROCEDURE
  (1)                           ;*---------
                                ;*BGNSUB
                                ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                ;*     CLEAR PORT SELECT FOR TRANSPORT
                                ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                ;*     CLEAR PORT AMTIE WORD
                                ;*ENDSUB
 49A2                           S
  (1)                           ; ************************************************************************
 49A2  F5            12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
 49A3  C5            12.0          PUSH    B
 49A4  06  00         7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
 49A6  DB  E0        10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
 49A8  E6  80         7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
 49AA  B0             4.0          ORA     B               ;ADD IN THE SELECTED PORT #
 49AB  D3  E0        10.0          OUT     MBSEL           ;RESET TO THIS PORT
 49AD  3E  80         7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
 49AF  D3  40        10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
 49B1  AF             4.0          XRA     A               ;CLEAR TU COMMAND A
 49B2  D3  40        10.0          OUT     TCMD            ;
 49B4  3E  81         7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
 49B6  D3  40        10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
 49B8  3E  00         7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
 49BA  D3  40        10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
 49BC  AF             4.0          XRA     A
 49BD  D3  44        10.0          OUT     TAMT            ;CLEAR AMTIE WORD
 49BF  D3  48        10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
 49C1  D3  4C        10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
 49C3  04             4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
 49C4  78             4.0          MOV     A,B
 49C5  FE  04         7.0          CPI     4               ;DONE?
 49C7  C2  A6  49    10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
 49CA  C1            10.0          POP     B               ;RESET B & C
 49CB  F1            10.0          POP     PSW             ;ALL DONE
 49CC  C9            10.0          RET                     ;EXIT
```

```
                                .SBTTL  PROGRAM VARIABLES

49CD  00                        UNITMP: .BYTE   0               ;UNIT MAP
49CE  00                        DATA:   .BYTE   0               ;DATA PATTERN
      0000                      .END
```

```
A      =%0007          ADATA  = 0094          ALP01  454F          ALP02  457D  G
ALP11  45D8            ALP12  4606  G        ALP21  4661          ALP22  468F  G
ALP31  46EA            ASAVE  4F9B           AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ATTCD  4F97           AXNUM  4F91
A1LP02 459B  G         A2LP02 4624  G        A3LP02 46AD  G        A4LP02 4736  G
B      =%0000          BADST  = 0090          BIT0   = 0001          BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008          BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080          BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00           BSAVE  4F9C           BYTCNT= 00D4          BYTEH  4F24
BYTEL  4F23            C      =%0001          CASCT  4F21          CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1
CBYTH  = 008B          CBYTL  = 008A          CDG1H  = 0087          CDG1L  = 0086
CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095          CDG3L  = 0094
CDVTH  = 008D          CDVTL  = 008C          CHPTIE= 0028          CHOTIE= 0020
CH1TIE= 0021           CH2TIE= 0022           CH3TIE= 0023          CH4TIE= 0024
CH5TIE= 0025           CH6TIE= 0026           CH7TIE= 0027          CKLOP  = 2815
CLEAR  49A2            CLKCTL= 00F0           CLOCK  4F26          CLRLP  49A6
CMC0H  = 0099          CMC0L  = 0098          CMC1H  = 009B          CMC1L  = 009A
CMC2H  = 009D          CMC2L  = 009C          CMC3H  = 009F          CMC3L  = 009E
CMINH  = 0097          CMINL  = 0096          CNTCTL= 00D7          CRCWRD= 0018
CSAVE  4F9D            CSRLH  = 0091          CSRLL  = 0090          CTCH   = 0085
CTCL   = 0084          CTSTH  = 008F          CTSTL  = 008E          CXCTH  = 0081
CXCTL  = 0080          CXINH  = 0083          CXINL  = 0082          C.     = 0001
C.AVAI= 0080           C.DP   = 0008          C.DSE  = 0010          C.DTU  = 0003
C.DVA  = 0008          C.FAIL= 00FC           C.FMT  = 0070          C.FNCT= 003E
C.GO   = 0001          C.INTC= 00FE           C.MAIN= 0020          C.NSA  = 0080
C.RCT  = 00FC          C.SER  = 0080          C.SHR  = 0040          C.SKPC= 000F
C.TAPE= 0040           C.WCS  = 0002          D      =%0002          DATA   49CE
DATACT= 00D0           DBUS   4F28           DBUSCT= 00C0          DBUSST= 00C0
DDRA   = 00D8          DDRAIN= 0010           DDRB   = 00D9          DDRBIN= 0002
DDRC   = 00DA          DDRCIN= 0001           DDRCO  = 0088          DDRCTL= 00DB
DIAFLG 4F22            DIAGPG= 4300           DIAGRM= 4F90          DIARD  = 000B
DONE1  = 0045          DONINT= 0010           DPEN00 434A          DPEN01 436C  G
DPEN02 4372            DPEN03 438B  G         DPEN10 43CF          DPEN11 43F1  G
DPEN12 43F7            DPEN13 4410  G         DPEN20 4454          DPEN21 4476  G
DPEN22 447C            DPEN23 4495  G         DPEN30 44D9          DPEN31 44FB  G
DPEN32 4501            DPEN33 451A  G         DSAVE  4F9E          DSE    = 0006
DUMMY  431E  G         D.ATH0= 0001           D.ATH1= 0002          D.EOTD= 0010
D.LAGC= 0020           D.NOTW= 0040           D.NTHR= 0004          D.TACH= 0008
D.WR4  = 0080          D0LPPC 47AC           D0LPP0 47A6          D0LPP1 47AA
D0LP01 477B            D0LP02 4799  G         D0LP03 47C1  G        D1LPPC 483E
D1LPP0 4838            D1LPP1 483C           D1LP01 480D          D1LP02 482B  G
D1LP03 4853  G         D2LPPC 48D0           D2LPP0 48CA          D2LPP1 48CE
D2LP01 489F            D2LP02 48BD  G         D2LP03 48E5  G        D3LPPC 4962
D3LPP0 495C            D3LPP1 4960           D3LP01 4931          D3LP02 494F  G
D3LP03 4977  G         E      =%0003          ECCBAD= 0042          ECCCOR= 0019
ECCOK  = 0041          ECCSTA= 001A           ECCTST= 000E          EDATA  = 0095
EOTCLR= 0003           ERFLG  4F93           ERLP   = 2809          ERLPA  = 280F
ERLPB  = 2812          ERLPE  = 280C          ERNUM  4F90          ERRCNT= 00D6
ESAVE  4F9F            EXIT   499C           E.ACRC= 0010          E.AMT  = 0020
E.CDP  = 0080          E.CRC  = 0080          E.PNTR= 0008          E.RPE  = 0040
E.STEC= 0001           E.TTEC= 0002           E.UNC  = 0004          FIFORD= 006A
FORMAT 4F25            FWDTST= 0061           GCRID  = 0089          GCRSET= 0002
```

```
GOODTM= 0092    H     =%0004    HLSAVE  4FA0    IE    = 0008
INTSTA= 00E0    ITERA   4F9A    I.PWR = 0020    I.RMPE= 0040
I5.5  = 0010    I6.5  = 0020    I7.5  = 0040    KCALL = 005F
KCLR  = 007B    KDEP  = 003F    KENAB = 0078    KEXAM = 003E
KEYBRD= 00C8    KEY1  = 0078    KEY10 = 006D    KEY11 = 006E
KEY12 = 006F    KEY13 = 005C    KEY14 = 005D    KEY15 = 005E
KEY16 = 005F    KEY17 = 003C    KEY18 = 003D    KEY19 = 003E
KEY2  = 0079    KEY20 = 003F    KEY3  = 007A    KEY4  = 007B
KEY5  = 0074    KEY6  = 0075    KEY7  = 0076    KEY8  = 0077
KEY9  = 006C    KINTA = 006F    KLDAD = 003D    KN0   = 003C
KN1   = 005C    KN2   = 005D    KN3   = 005E    KN4   = 006C
KN5   = 006D    KN6   = 006E    KN7   = 0074    KN8   = 0075
KN9   = 0076    KU2   = 0079    KU3   = 007A    KU8   = 0077
L     =%0005    LBLANK= 000F    LCE   = 000B    LCH   = 000C
LCL   = 000D    LCP   = 000E    LC0   = 0000    LC1   = 0001
LC2   = 0002    LC3   = 0003    LC4   = 0004    LC5   = 0005
LC6   = 0006    LC7   = 0007    LC8   = 0008    LC9   = 0009
LDLEDA= 00CA    LDLEDB= 00CB    LDLEDC= 00CC    LDLEDD= 00CD
LDLEDE= 00CE    LDLEDF= 00CF    LKDIAG= 2800    LKKBD = 004C
LKKEY = 0049    LKLWMG= 0058    LKLWMP= 0055    LKLWPG= 0052
LKLWPP= 004F    LKMOD7= 0046    LKOPR = 0046    LPFLG   4F94
LPNUM   4F92    M     =%0006    MBSEL = 00E0    MB.A  = 0008
MB.B  = 0004    MEMTOP= 4FFF    MINUS = 000A    MM    = 8000
MSE   = 0008    MSGN  = 001D    MTACLR= 0000    MT.ARA= 0020
MT.CPE= 0080    MT.DSE= 0001    MT.FWD= 0040    MT.INH= 0008
MT.LWR= 0004    MT.MOT= 0002    MT.NWT= 0080    MT.PEC= 0040
MT.PSB= 0004    MT.PS0= 0001    MT.PS1= 0002    MT.REV= 0020
MT.WRT= 0010    MT.Z  = 0008    M.ATA = 0080    M.CAPE= 0020
M.CONT= 0080    M.DEM = 0020    M.EBL = 0010    M.EXC = 0008
M.FAIL= 0008    M.ILR = 0010    M.INIT= 0010    M.OCC = 0020
M.ONLI= 0001    M.PE  = 0040    M.PORT= 0080    M.RDEN= 0002
M.RDPE= 0008    M.RUN = 0004    M.SCLK= 0001    M.TRA = 0040
M.UNIT= 0007    M.WCLK= 0040    M.WCLN= 0080    M.WREN= 0080
M5.5  = 0001    M6.5  = 0002    M7.5  = 0004    NOTCAP= 0088
OKAY  = 00FF    OPRRAM= 4300    OPSTRT= 0058    OPVER = 0040
PADCNT= 00D5    PADCRC= 0080    PDIAG = 0048    PEID  = 008A
PENAB = 004C    PESET = 0001    PL    = 00B1    PRDD  = 004C
PRENF = 009C    PS    = 00B2    PSTAT = 0048    PSW   =%0009
P.AMTP= 0001    P.BCTC= 0040    P.CMDP= 0020    P.INTE= 0080
P.LCS = 0040    P.LWR = 0020    P.RDP = 0002    P.RPEN= 0004
P.RPST= 0002    P.RP0E= 0020    P.RP1E= 0010    P.RP2E= 0020
P.RP3E= 0010    P.SING= 0080    P.STAT= 0002    P.STPE= 0080
P.TACH= 0008    P.TUPR= 0010    P.WCSP= 0004    P.WDS = 0040
P.WFLP= 0001    P.WPEN= 0010    P.WPOE= 0008    P.WP1E= 0004
P.WP2E= 0008    P.WP3E= 0004    P.5VOK= 0002    QUE   = 281B
QUEM  = 2B1E    RAMT  = 0010    RARA  = 0006    RARAI = 0004
RCHBD0= 0048    RCHBD1= 0047    RCHOK = 0046    RCHTST= 000C
RCLRT = 000D    RCMD  = 000B    RCMLP = 0003    RCONT = 0080
RDATA = 0017    RDCLK = 0010    RDON  = 0011    READG = 0007
REND  = 0014    REQST = 2806    RESCHR= 00D1    REVTST= 0064
REWIND= 0004    RFIFOL= 0008    RGCLK = 0002    RGCRI = 0003
RIBG  = 0001    RILL  = 0012    RINST = 000C    RMLTST= 0008
RMK2  = 0013    RNOP  = 0000    RPATH = 0001    RPBAD = 0044
RPCHI = 0001    RPCLK = 0003    RPCTL = 0009    RPEI  = 0002
```

```
RPFAIL= 0000          RPF1  = 009D          RPF2  = 009E          RPOK  = 0043
RPOSTN= 0016          RPSTA = 0015          RRCMT = 000A          RSTAT = 0002
RTIEB = 000A          RTIER = 0030          RTM   = 0005          RUNKI = 0009
RUPTST= 005E          RWDUNL= 0005          R.AMT = 0001          R.BOP = 0008
R.DATA= 0040          R.DON = 0002          R.DRDY= 0010          R.END = 0010
R.ILL = 0004          R.JVOK= 0004          R.MK2 = 0008          R.PLOD= 0008
R.PLO0= 0010          R.PLO1= 0020          R.POST= 0020          R.STNM= 0002
R.STOP= 0004          R.STPC= 0001          R.TBJN= 0080          R.TSTD= 0040
R.VOK = 0080          R00H  = 0081          R00L  = 0080          R01H  = 0083
R01L  = 0082          R02H  = 0085          R02L  = 0084          R03H  = 0087
R03L  = 0086          R04H  = 0089          R04L  = 0088          R05H  = 008B
R05L  = 008A          R06H  = 008D          R06L  = 008C          R07H  = 008F
R07L  = 008E          R10H  = 0091          R10L  = 0090          R11H  = 0093
R11L  = 0092          R12H  = 0095          R12L  = 0094          R13H  = 0097
R13L  = 0096          R14H  = 0099          R14L  = 0098          R15H  = 009B
R15L  = 009A          R16H  = 009D          R16L  = 009C          R17H  = 009F
R17L  = 009E          R7.5  = 0010          SELCLR= 0000          SETATA= 00A1
SID   = 0080          SOD   = 0080          SOE   = 0040          SP    =%0008
SSCLK = 0040          SSTEP = 0005          STACK = 4FFF          STATRM= 4F20
STPCT   4F20          STRSP = 5000          TADR00= 0080          TADR01= 0081
TADR02= 0082          TADR03= 0083          TADR04= 0084          TADR05= 0085
TADR06= 0086          TADR07= 0087          TADR10= 0088          TADR11= 0089
TADR12= 008A          TADR13= 008B          TAMT  = 0044          TASEL = 0080
TCMD  = 0040          TC.FWD= 0040          TC.INH= 0008          TC.LWR= 0004
TC.REV= 0020          TC.WRT= 0010          TEMP    4F99          TEST01  4300
TEST02  43A5          TEST03  442A          TEST04  44AF          TEST05  4534
TEST06  45BB          TEST07  4644          TEST10  46CD          TEST11  4756
TEST12  47E6          TEST13  4878          TEST14  490A          TMF   = 0099
TMRDY = 0040          TRKENA= 00D2          TSET  = 2803          TSTEND= 2818
TSTS  = 0040          TST01C  4344  G       TST01X  4327          TST02C  43C9  G
TST02X  43AA          TST03C  444E  G       TST03X  442F          TST04C  44D3  G
TST04X  44B4          TST05X  4539          TST06X  45C0          TST07X  4649
TST10X  46D2          TST11X  475B          TST12X  47EB          TST13X  487D
TST14X  490F          TUSEL0= 00D1          TUSEL1= 00D2          TU78  = 0010
T.ATH0= 0001          T.ATH1= 0002          T.BOT = 0004          T.EOT = 0002
T.FPT = 0001          T.NTHR= 0004          T.ONL = 0020          T.PES = 0008
T.PSBJ= 0020          T.PSOJ= 0008          T.PS1J= 0010          T.RDY = 0080
T.RDYO= 0040          T.RWD = 0010          T.SCLK= 0002          UIBG  = 00A1
UNITMP  49CD          VALFC   4F98          VALTB   4F95          VELTST= 005B
WDR.P = 0010          WMCCTL= 00D3          WMCERR= 00DA          WMCSTA= 00D0
WRTCLK= 0000          WRTDAT= 00D3          W.ACRC= 0004          W.CRC = 0008
W.DIAG= 0002          W.DONN= 0040          W.ECC = 0010          W.ENAB= 0080
W.ERR = 0020          W.FMT = 0070          W.GCR = 0010          W.LEFT= 0004
W.ONES= 0020          W.RESI= 0002          W.REV = 0004          W.ROME= 0010
W.RST = 0001          W.SKIP= 000F          W.WRIT= 0008          W.XFER= 0020
X     =%000A          X.DONN= 0080          X.ENAB= 0040          X.PEPE= 0002
X.ROME= 0001          X.WCLK= 0001          Y     =%000B          .     - 49CF


 ERRORS DETECTED:  0

 *MTA1.A78/PTP,MTA1=NLIST,PARAM,MACRO,LIST,MTA1
 RUN-TIME: 5 8 0 SECONDS
```

CORE USED: 10K

3 "MEMORY MAP" EXTENDED ADDRESS BIT
19 READ MICRO CONTROLLER CONTROL & STATUS WORDS
139 TRANSPORT/PORT COMMAND + STATUS WORDS
380 COMMON ADDRESS SPACE REGISTERS
444 MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485 MASSBUS CONTROL WORD DEFINITIONS
557 WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709 KEYBOARD/DISPLAY CONTROL + STATUS WORD
797 MISCELLANEOUS VARIABLE DEFINITIONS
801 RAM ADDRESS DEFINITIONS
819 LINKAGE TABLE ADDRESSES
842 BIT DEFINITIONS
856 RIM AND SIM INSTRUCTION BIT DEFINITIONS
877 INTERRUPTABLE INSTRUCTION MACROS
924 DIAGNOSTIC VARIABLES
945 COMMON VARIABLE STORAGE
959 MACRO CALL DEFINITIONS
1047 TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101 8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1334 TEST 01 - MASSBUS PORT SELECT SWITCH TEST - TU #0
1495 TEST 02 - MASSBUS PORT SELECT SWITCH TEST - TU #1
1637 TEST 03 - MASSBUS PORT SELECT SWITCH TEST - TU #2
1777 TEST 04 - MASSBUS PORT SELECT SWITCH TEST - TU #3
1917 TEST 05 - LOOP THRESHOLD CONTROL BITS - MTA #0
1996 TEST 06 - LOOP THRESHOLD CONTROL BITS - MTA #1
2076 TEST 07 - LOOP THRESHOLD CONTROL BITS - MTA #2
2156 TEST 10 - LOOP THRESHOLD CONTROL BITS - MTA #3
2236 TEST 11 - COMMAND FUNCTION TEST - MTA #0
2340 TEST 12 - COMMAND FUNCTION TEST - MTA #1
2445 TEST 13 - COMMAND FUNCTION TEST - MTA #2
2550 TEST 14 - COMMAND FUNCTION TEST - MTA #3
2658 SUBROUTINE CLEAR ALL TU PORTS
2705 SUBROUTINE COMMAND COMMON
2887 PROGRAM VARIABLES

```
961                     ;*************************************************************************
962                     ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
963                     ;ERROR MACRO CALLS
964                     ;
965                     ;       1  -    REQUEST HOST CPU TO PRINT:
966                     ;               "BYTE/SCLK COUNT NUMBER = LLL"
967                     ;               WHERE:
968                     ;               LLL = THE VALUE STORED IN CAS REGISTER 5
969                     ;                     (THE BYTE COUNT REGISTER 16 BITS).
970                     ;
971                     ;       2  -    REQUEST HOST CPU TO PRINT:
972                     ;               DATA FORMAT = MM
973                     ;               SKIP COUNT  = NN
974                     ;               WHERE:
975                     ;               MM = DATA FORMAT FROM CAS REGISTER 2
976                     ;               NN = SKIP COUNT FROM CAS REGISTER 2
977                     ;
978                     ;       5  -    REQUEST HOST CPU TO PRINT:
979                     ;               BYTE-SCLK COUNT = LLL
980                     ;               DATA FORMAT = MM
981                     ;               SKIP COUNT  = NN
982                     ;               WHERE:
983                     ;               LLL = AS ABOVE
984                     ;               MM  = AS ABOVE
985                     ;               NN  = AS ABOVE
986                     ;
987                     ;       4  -    REQUEST HOST TO PRINT:
988                     ;               TRANSITION COUNT = LLL
989                     ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
990                     ;
991                     ;       5  -    REQUEST HOST CPU TO PRINT:
992                     ;               EXPECTED 18 BITS =      E EEEEEE
993                     ;               ACTUAL 18 BITS =        A AAAAAA
994                     ;
995                     ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
996                     ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
997                     ;               OF CAS REG 15 LOW BYTE.
998                     ;
999                     ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
1000                    ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
1001                    ;               17 LOW BYTE SIGN BIT.
1002                    ;
1003                    ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
1004                    ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
1005                    ;               AND/OR ACTUAL DATA.
1006                    ;
1007                    ;       7  -    REQUEST HOST CPU TO PRINT:
1008                    ;               "SUBGROUP NUMBER = LLL"
1009                    ;               WHERE:
1010                    ;               LLL = THE VALUE STORED IN CAS REGISTER 5
1011                    ;                     (THE BYTE COUNT REGISTER 16 BITS).
1012                    ;
1013                    ;       10 -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
1014                    ;*************************************************************************
```

```
1016                            ; ************************************************************************
1017                            ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                            ;TO CAUSE SOME HOST CPU ACTION.
1019                            ;
1020                            ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                            ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                            ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                            ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                            ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                            ;               - HOST RESPONSE CODE 31 OR 33
1026                            ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                            ;               - HOST RESPONSE CODE 31 OR 33
1028                            ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                            ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                            ;                       BIT0 = 1 TEST PORT 0
1031                            ;                       BIT1 = 1 TEST PORT 1
1032                            ;                       BIT2 = 1 TEST PORT 2
1033                            ;                       BIT3 = 1 TEST PORT 3
1034                            ; ************************************************************************
1035                            ; ************************************************************************
1036                            ;DATA PATTERN CODES
1037                            ;
1038                            ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                            ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                            ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                            ;               18 BITS OF ALL 1'S
1042                            ;               18 BITS OF ALL 0'S
1043                            ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                            ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                            ; ************************************************************************
1329    4300                            .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
1332                                .TITLE  MTA2 - MAG TAPE ADAPTER TEST PART #2
1333                                ;ID MTA2-MAG TAPE ADAPTER CONTROLLER PART #2
1334                                .SBTTL  TEST 01 - MASSBUS PORT SELECT SWITCH TEST - TU #0
1335  4300                          ST
 (1)                                ; **********************************************************************
 (1)                                ;*TEST TITLE
 (1)                                ;*---- -----
1336                                ;*MASSBUS PORT SELECT SWITCH TEST - TU #0
1337  4300                          SD
 (1)                                ; **********************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
1338                                ;*THIS TEST CHECKS THE FUNCTIONALITY OF THE TU78 PORT SELECT SWITCH AND
1339                                ;*ASSOCIATED HARDWARE, FOR TU #0.
1340  4300                          SP
 (1)                                ; **********************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
1341                                ;*BGNTST
1342                                ;*   IF TU PORT #0 NOT SELECTED BY USER
1343                                ;*   :  THEN-NEXT TEST
1344                                ;*   :  ELSE-CONTINUE
1345                                ;*   ENDIF
1346                                ;*   IF MANUAL INTERVENTION NOT ALLOWED
1347                                ;*   :  THEN-NEXT TEST
1348                                ;*   :  ELSE-CONTINUE
1349                                ;*   ENDIF
1350                                ;*   CALL SUBROUTINE CLEAR
1351                                ;*   SELECT TU PORT #0
1352                                ;*   SET PSB NOT, PS1 NOT, PS0 NOT FOR MTA #0
1353                                ;*   REQUEST THE USER TO SET TU78 #0 TO MASSBUS PORT 0
1354                                ;*   INPUT MTA #0 STATUS REGISTER B
1355                                ;*   IF PSB NOT AND PS1 NOT=1 AND BS0 NOT=0
1356                                ;*   :  THEN-CONTINUE
1357                                ;*   :  ELSE-ERROR
1358                                ;*   ENDIF
1359                                ;*   REQUEST THE USER TO SET TU78 #0 TO MASSBUS PORT 1
1360                                ;*   INPUT MTA #0 STATUS REGISTER B
1361                                ;*   IF PSB NOT AND PS0 NOT=1 AND PS1 NOT=0
1362                                ;*   :  THEN-CONTINUE
1363                                ;*   :  ELSE-ERROR
1364                                ;*   ENDIF
1365                                ;*   REQUEST THE USER TO SET TU78 #0 TO BOTH MASSBUS PORTS
1366                                ;*   INPUT MTA #0 STATUS REGISTER B
1367                                ;*   IF PSB NOT=0 AND PS0 NOT=1 AND PS1 NOT-1
1368                                ;*   :  THEN-CONTINUE
1369                                ;*   :  ELSE-ERROR
1370                                ;*   ENDIF
1371                                ;*   REQUEST THE USER TO SET TU78 #0 TO MAINTENANCE MODE
1372                                ;*   INPUT MTA #0 STATUS REGISTER B
1373                                ;*   IF PSB NOT PS0 NOT AND PS1 NOT 1
1374                                ;*   :  THEN-CONTINUE
1375                                ;*   :  ELSE-ERROR
1376                                ;*   ENDIF
```

```
                                        ;*ENDTST
4300                                    SE
 (1)                                    ; ************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
                                        ;*MTA2 MICRO TEST 01
                                        ;*MTA2 MICRO ERROR 01
                                        ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #0
                                        ;*M8954, TU CABLE, M8955'S
                                        ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                        ;*FATAL ERROR - TEST ABORTED
                                        ;*
                                        ;*MTA2 MICRO TEST 01
                                        ;*MTA2 MICRO ERROR 03
                                        ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #0
                                        ;*M8954, TU CABLE, M8955'S
                                        ;*PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 0
                                        ;*ACUTAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA2 MICRO TEST 01
                                        ;*MTA2 MICRO ERROR 05
                                        ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #0
                                        ;*M8954, TU CABLE, M8955'S
                                        ;*PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 1
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA2 MICRO TEST 01
                                        ;*MTA2 MICRO ERROR 07
                                        ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #0
                                        ;*M8954, TU CABLE, M8955'S
                                        ;*PORT SELECT SWITCH DOES NOT INDICATE BOTH MASSBUS PORTS 0 AND 1
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*MTA2 MICRO TEST 01
                                        ;*MTA2 MICRO ERROR 11
                                        ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #0
                                        ;*M8954, TU CABLE, M8955'S
                                        ;*PORT SELECT SWITCH DOES NOT INDICATE MAINTENANCE
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4300                                    S
 (1)                                    ; ************************************************************************

4300                                    TEST01: TESTX   @01             ;INITIALIZE THE TEST
 (1) 4300  3E  01          7.0                         MVI     A,@01       ;DEFINE THE TEST NUMBER
 (1) 4302  CD  03  28     18.0                         CALL    TSET        ;SETUP THE TEST
                                        ;%MTA MASSBUS PORT SELECT SWITCH TEST - TU #0
                                        ;&M8954, TU CABLE, M8955'S
4305                            REQ     @7,0,0,0,0
 (1) 4305  CD  06  28     18.0                         CALL    REQST
 (1) 4308  00                                          .BYTE   0           ;DATA PATTERN NUMBER
```

```
 (1)  4309  00  00                                  .WORD   0               ;SYSTEM "0" COUNT
 (1)  430B  00  00                                  .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                      .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1)  430E  07                                      .BYTE   @7              ;REQUEST CODE
1423  430F                              RIN     R12L            ;GET THE UNITS DESIRED
 (1)  430F  DB  94          10.0                   IN      R12L            ;READ R12L INTO AC
 (1)  4311  7F               4.0                   MOV     A,A             ;RETRY LINK
1424  4312  32  BA  49       13.0          STA     UNITMP          ;STORE IN MEMORY
1425  4315  A7               4.0          ANA     A               ;SET THE CONDITION CODE
1426  4316  C2  27  43       10.0          JNZ     TST01X          ;GO TEST THE SPECIFIED UNITS
1427                                                              ;NO UNITS SPECIFIED
1428  4319                              ERR     EXIT,DUMMY
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28       18.0                   CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                      .BYTE
 (1)  431F  CD  15  28       18.0          DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  1D  48       10.0                  JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1429                                ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1430                                ;>FATAL ERROR - TEST ABORTED
1431  4324  C3  1D  48       10.0          JMP     EXIT            ;EXIT THE TEST
1432
1433  4327  3A  BA  49       13.0  TST01X: LDA     UNITMP          ;GET THE UNIT MAP
1434  432A  E6  01               7.0          ANI     @01             ;TEST UNIT 0
1435  432C  CA  D2  43       10.0          JZ      TEST02          ;NO-CHECK FOR TEST 02
1436  432F  CD  20  48       18.0          CALL    CLEAR           ;CLEAR THE TU PORTS
1437  4332  DB  E0           10.0          IN      INTSTA          ;SELECT TU PORT 0
1438  4334  E6  80               7.0          ANI     BIT7
1439  4336  F6  00               7.0          ORI     @00
1440  4338  D3  E0           10.0          OUT     MBSEL
1441  433A  3E  83               7.0          MVI     A,$83           ;SELECT MTA 0 REGISTER 3
1442  433C  D3  40           10.0          OUT     TCMD
1443  433E  3E  38               7.0          MVI     A,$38           ;SET THE PS0 NOT, PS1 NOT,
1444  4340  D3  40           10.0          OUT     TCMD            ;AND PSB NOT BITS.
1445  4342                              QMSGM   TEST02
 (1)  4342  CD  1E  28       18.0          CALL    QUEM
 (1)        0002                          MSGN=MSGN+1
 (1)  4345  02                            .BYTE   MSGN
 (1)  4346  DA  D2  43       10.0          JC      TEST02
1446                                ;>SET THE TU78 #0 PORT SELECT SWITCH TO MASSBUS PORT 0
1447  4349  3E  82               7.0          MVI     A,$82           ;SELECT MTA 0 REGISTER 2
1448  434B  D3  40           10.0          OUT     TCMD
1449  434D  DB  40           10.0          IN      TSTS            ;GET THE TU PORT STATUS
1450  434F  E6  07               7.0          ANI     $07             ;SAVE ONLY PORT SELECT BITS
1451  4351  FE  06               7.0          CPI     BIT1!BIT2       ;COMPARE WITH EXPECTED
1452  4353  CA  63  43       10.0          JZ      T1CN1           ;BRANCH IF EQUAL
1453  4356                              ROUT    ADATA           ;SAVE THE ACTUAL DATA
 (1)  4356  D3  94           10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4358  7F               4.0                   MOV     A,A             ;RETRY LINK
1454  4359  3E  06               7.0          MVI     A,BIT1!BIT2     ;SAVE THE EXPECTED DATA
1455  435B                              ROUT    EDATA
 (1)  435B  D3  95           10.0                   OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  435D  7F               4.0                   MOV     A,A             ;RETRY LINK
```

```
1456  435E                                    ERRB    TST01X,T1CN1
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  435E  CD  12  28        18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0003                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4361  03                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4362  00                                        .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4363  CD  15  28        18.0              T1CN1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4366  DA  27  43        10.0                    JC      TST01X          ;LOOP ADDRESS IF LOOP SPECIFIED
1457                                    ;>PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 0
1458
1459  4369                                    QMSGM   TEST02
 (1)  4369  CD  1E  28        18.0              CALL    QUEM
 (1)        0004                                MSGN=MSGN+1
 (1)  436C  04                                  .BYTE   MSGN
 (1)  436D  DA  D2  43        10.0              JC      TEST02
1460                                    ;>SET THE TU78 #0 PORT SELECT SWITCH TO MASSBUS PORT 1
1461  4370  DB  40              10.0              IN      TSTS
1462  4372  E6  07               7.0              ANI     $07
1463  4374  FE  05               7.0              CPI     BIT0!BIT2
1464  4376  CA  86  43          10.0              JZ      T1CN2
1465  4379                                    ROUT    ADATA
 (1)  4379  D3  94              10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  437B  7F                   4.0                    MOV     A,A             ;RETRY LINK
1466  437C  3E  05               7.0              MVI     A,BIT0!BIT2
1467  437E                                    ROUT    EDATA
 (1)  437E  D3  95              10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4380  7F                   4.0                    MOV     A,A             ;RETRY LINK
1468  4381                                    ERRB    TST01X,T1CN2
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4381  CD  12  28        18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0005                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4384  05                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4385  00                                        .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4386  CD  15  28        18.0              T1CN2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4389  DA  27  43        10.0                    JC      TST01X          ;LOOP ADDRESS IF LOOP SPECIFIED
1469                                    ;>PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 1
1470
1471  438C                                    QMSGM   TEST02
 (1)  438C  CD  1E  28        18.0              CALL    QUEM
 (1)        0006                                MSGN=MSGN+1
 (1)  438F  06                                  .BYTE   MSGN
 (1)  4390  DA  D2  43        10.0              JC      TEST02
1472                                    ;>SET THE TU78 #0 PORT SELECT SWITCH TO BOTH PORTS 0 AND 1
1473  4393  DB  40              10.0              IN      TSTS
1474  4395  E6  07               7.0              ANI     $07
1475  4397  FE  03               7.0              CPI     BIT0!BIT1
1476  4399  CA  A9  43          10.0              JZ      T1CN3
1477  439C                                    ROUT    ADATA
 (1)  439C  D3  94              10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  439E  7F                   4.0                    MOV     A,A             ;RETRY LINK
1478  439F  3E  03               7.0              MVI     A,BIT0!BIT1
1479  43A1                                    ROUT    EDATA
 (1)  43A1  D3  95              10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  43A3  7F                   4.0                    MOV     A,A             ;RETRY LINK
```

```
1480  43A4                                    ERRB    TST01X,T1CN3
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  43A4  CD  12  28        18.0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0007                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43A7  07                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43A8  00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  43A9  CD  15  28        18.0               T1CN3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43AC  DA  27  43        10.0                       JC      TST01X          ;LOOP ADDRESS IF LOOP SPECIFIED
1481                                    ;>PORT SELECT SWITCH DOES NOT INDICATE BOTH MASSBUS PORTS 0 AND 1
1482
1483  43AF                                    QMSGM   TEST02
 (1)  43AF  CD  1E  28        18.0               CALL    QUEM
 (1)        0008                                 MSGN=MSGN+1
 (1)  43B2  08                                   .BYTE   MSGN
 (1)  43B3  DA  D2  43        10.0               JC      TEST02
1484                                    ;>SET THE TU78 #0 PORT SELECT SWITCH TO MAINTENANCE
1485  43B6  DB  40              10.0       IN      TSTS
1486  43B8  E6  07               7.0       ANI     $07
1487  43BA  FE  07               7.0       CPI     BIT0!BIT1!BIT2
1488  43BC  CA  CC  43          10.0       JZ      T1CN4
1489  43BF                                 ROUT    ADATA
 (1)  43BF  D3  94              10.0               OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43C1  7F                   4.0               MOV     A,A             ;RETRY LINK
1490  43C2  3E  07               7.0       MVI     A,BIT0!BIT1.BIT2
1491  43C4                                 ROUT    EDATA
 (1)  43C4  D3  95              10.0               OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  43C6  7F                   4.0               MOV     A,A             ;RETRY LINK
1492  43C7                                    ERRB    TST01X,T1CN4
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  43C7  CD  12  28        18.0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0009                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43CA  09                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43CB  00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  43CC  CD  15  28        18.0               T1CN4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43CF  DA  27  43        10.0                       JC      TST01X          ;LOOP ADDRESS IF LOOP SPECIFIED
1493                                    ;>PORT SELECT SWITCH DOES NOT INDICATE MAINTENANCE
```

```
                                .SBTTL  TEST 02 - MASSBUS PORT SELECT SWITCH TEST - TU #1
43D2                            ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*MASSBUS PORT SELECT SWITCH TEST - TU #1
43D2                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE FUNCTIONALITY OF THE TU78 PORT SELECT SWITCH AND
                                ;*ASSOCIATED HARDWARE, FOR TU #1.
43D2                            SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  IF TU PORT #1 NOT SELECTED BY USER
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  IF MANUAL INTERVENTION NOT ALLOWED
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  CALL SUBROUTINE CLEAR
                                ;*  SELECT TU PORT #1
                                ;*  SET PSB NOT, PS1 NOT, PS0 NOT FOR MTA #1
                                ;*  REQUEST THE USER TO SET TU78 #1 TO MASSBUS PORT 0
                                ;*  INPUT MTA #1 STATUS REGISTER B
                                ;*  IF PSB NOT AND PS1 NOT=1 AND BS0 NOT=0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  REQUEST THE USER TO SET TU78 #1 TO MASSBUS PORT 1
                                ;*  INPUT MTA #1 STATUS REGISTER B
                                ;*  IF PSB NOT AND PS0 NOT=1 AND PS1 NOT 0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  REQUEST THE USER TO SET TU78 #1 TO BOTH MASSBUS PORTS
                                ;*  INPUT MTA #1 STATUS REGISTER B
                                ;*  IF PSB NOT=0 AND PS0 NOT=1 AND PS1 NOT=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  REQUEST THE USER TO SET TU78 #1 TO MAINTENANCE MODE
                                ;*  INPUT MTS #1 STATUS REGISTER B
                                ;*  IF PSB NOT PS0 NOT AND PS1 NOT-1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*ENDTST
43D2                            SE
```

```
 (1)                                ; ***************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1540                                ;*MTA2 MICRO TEST 02
1541                                ;*MTA2 MICRO ERROR 13
1542                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #1
1543                                ;*M8954, TU CABLE, M8955'S
1544                                ;*PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 0
1545                                ;*ACTUAL = NNNN
1546                                ;*EXPECTED = NNNN
1547                                ;*
1548                                ;*MTA2 MICRO TEST 02
1549                                ;*MTA2 MICRO ERROR 15
1550                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #1
1551                                ;*M8954, TU CABLE, M8955'S
1552                                ;*PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 1
1553                                ;*ACTUAL = NNNN
1554                                ;*EXPECTED = NNNN
1555                                ;*
1556                                ;*MTA2 MICRO TEST 02
1557                                ;*MTA2 MICRO ERROR 17
1558                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #1
1559                                ;*M8954, TU CABLE, M8955'S
1560                                ;*PORT SELECT SWITCH DOES NOT INDICATE BOTH MASSBUS PORTS 0 AND 1
1561                                ;*ACTUAL = NNNN
1562                                ;*EXPECTED = NNNN
1563                                ;*
1564                                ;*MTA2 MICRO TEST 02
1565                                ;*MTA2 MICRO ERROR 21
1566                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #1
1567                                ;*M8954, TU CABLE, M8955'S
1568                                ;*PORT SELECT SWITCH DOES NOT INDICATE MAINTENANCE
1569                                ;*ACTUAL = NNNN
1570                                ;*EXPECTED = NNNN
1571  43D2                          S
 (1)                                ; ***************************************************************************
1572
1573  43D2                          TEST02: TESTX   @02             ;INITIALIZE THE TEST
 (1)  43D2  3E  02          7.0                  MVI     A,@02       ;DEFINE THE TEST NUMBER
 (1)  43D4  CD  03  28     18.0                  CALL    TSET        ;SETUP THE TEST
1574                                ;%MTA MASSBUS PORT SELECT SWITCH TEST - TU #1
1575                                ;&M8954, TU CABLE, M8955'S
1576  43D7  3A  BA  49     13.0    TST02X: LDA     UNITMP          ;GET THE UNIT MAP
1577  43DA  E6  02          7.0            ANI     @02             ;TEST UNIT 1
1578  43DC  CA  82  44      10.0            JZ      TEST03          ;NO-CHECK FOR TEST 03
1579  43DF  CD  20  48      18.0            CALL    CLEAR           ;CLEAR THE TU PORTS
1580  43E2  DB  E0          10.0            IN      INTSTA          ;SELECT TU PORT 1
1581  43E4  E6  80           7.0            ANI     BIT7
1582  43E6  F6  01           7.0            ORI     @01
1583  43E8  D3  E0          10.0            OUT     MBSEL
1584  43EA  3E  83           7.0            MVI     A,$83           ,SELECT MTA 1 REGISTER 3
1585  43EC  D3  40          10.0            OUT     TCMD
1586  43EE  3E  38           7.0            MVI     A,$38           ;SET THE PS0 NOT, PS1 NOT,
1587  43F0  D3  40          10.0            OUT     TCMD            ;AND PSB NOT BITS.
```

```
1588  43F2                                    QMSGM   TEST03
 (1)  43F2  CD  1E  28        18.0            CALL    QUEM
 (1)        000A                              MSGN=MSGN+1
 (1)  43F5  0A                                .BYTE   MSGN
 (1)  43F6  DA  82  44        10.0            JC      TEST03
1589                                    ;>SET THE TU78 #1 PORT SELECT SWITCH TO MASSBUS PORT 0
1590  43F9  3E  82               7.0          MVI     A,$82           ;SELECT MTA 1 REGISTER 2
1591  43FB  D3  40              10.0          OUT     TCMD
1592  43FD  DB  40              10.0          IN      TSTS            ;GET THE TU PORT STATUS
1593  43FF  E6  07               7.0          ANI     $07             ;SAVE ONLY PORT SELECT BITS
1594  4401  FE  06               7.0          CPI     BIT1!BIT2       ;COMPARE WITH EXPECTED
1595  4403  CA  13  44          10.0          JZ      T2CN1           ;BRANCH IF EQUAL
1596  4406                                    ROUT    ADATA           ;SAVE THE ACTUAL DATA
 (1)  4406  D3  94              10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4408  7F                   4.0                  MOV     A,A             ;RETRY LINK
1597  4409  3E  06               7.0          MVI     A,BIT1!BIT2     ;SAVE THE EXPECTED DATA
1598  440B                                    ROUT    EDATA
 (1)  440B  D3  95              10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  440D  7F                   4.0                  MOV     A,A             ;RETRY LINK
1599  440E                                    ERRB    TST02X,T2CN1
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  440E  CD  12  28        18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000B                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4411  0B                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4412  00                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4413  CD  15  28        18.0            T2CN1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4416  DA  D7  43        10.0                    JC      TST02X          ;LOOP ADDRESS IF LOOP SPECIFIED
1600                                    ;>PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 0
1601  4419                                    QMSGM   TEST03
 (1)  4419  CD  1E  28        18.0            CALL    QUEM
 (1)        000C                              MSGN=MSGN+1
 (1)  441C  0C                                .BYTE   MSGN
 (1)  441D  DA  82  44        10.0            JC      TEST03
1602                                    ;>SET THE TU78 #1 PORT SELECT SWITCH TO MASSBUS PORT 1
1603  4420  DB  40              10.0          IN      TSTS
1604  4422  E6  07               7.0          ANI     $07
1605  4424  FE  05               7.0          CPI     BIT0!BIT2
1606  4426  CA  36  44          10.0          JZ      T2CN2
1607  4429                                    ROUT    ADATA
 (1)  4429  D3  94              10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  442B  7F                   4.0                  MOV     A,A             ;RETRY LINK
1608  442C  3E  05               7.0          MVI     A,BIT0!BIT2
1609  442E                                    ROUT    EDATA
 (1)  442E  D3  95              10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4430  7F                   4.0                  MOV     A,A             ;RETRY LINK
1610  4431                                    ERRB    TST02X,T2CN2
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4431  CD  12  28        18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000D                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4434  0D                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4435  00                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4436  CD  15  28        18.0            T2CN2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4439  DA  D7  43        10.0                    JC      TST02X          ;LOOP ADDRESS IF LOOP SPECIFIED
1611                                    ;>PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 1
```

```
1612
1613  443C                                    QMSGM   TEST03
 (1)  443C  CD  1E  28        18.0           CALL    QUEM
 (1)        000E                             MSGN=MSGN+1
 (1)  443F  0E                               .BYTE   MSGN
 (1)  4440  DA  82  44        10.0           JC      TEST03
1614                                ;>SET THE TU78 #1 PORT SELECT SWITCH TO BOTH PORTS 0 AND 1
1615  4443  DB  40              10.0          IN      TSTS
1616  4445  E6  07               7.0          ANI     $07
1617  4447  FE  03               7.0          CPI     BIT0!BIT1
1618  4449  CA  59  44          10.0          JZ      T2CN3
1619  444C                                    ROUT    ADATA
 (1)  444C  D3  94              10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  444E  7F                   4.0                  MOV     A,A             ;RETRY LINK
1620  444F  3E  03               7.0          MVI     A,BIT0!BIT1
1621  4451                                    ROUT    EDATA
 (1)  4451  D3  95              10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4453  7F                   4.0                  MOV     A,A             ;RETRY LINK
1622  4454                                    ERRB    TST02X,T2CN3
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4454  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000F                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4457  0F                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4458  00                                             .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4459  CD  15  28        18.0                T2CN3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  445C  DA  D7  43        10.0                        JC      TST02X          ;LOOP ADDRESS IF LOOP SPECIFIED
1623                                ;>PORT SELECT SWITCH DOES NOT INDICATE BOTH MASSBUS PORTS 0 AND 1
1624
1625  445F                                    QMSGM   TEST03
 (1)  445F  CD  1E  28        18.0           CALL    QUEM
 (1)        0010                             MSGN=MSGN+1
 (1)  4462  10                               .BYTE   MSGN
 (1)  4463  DA  82  44        10.0           JC      TEST03
1626                                ;>SET THE TU78 #1 PORT SELECT SWITCH TO MAINTENANCE
1627  4466  DB  40              10.0          IN      TSTS
1628  4468  E6  07               7.0          ANI     $07
1629  446A  FE  07               7.0          CPI     BIT0!BIT1!BIT2
1630  446C  CA  7C  44          10.0          JZ      T2CN4
1631  446F                                    ROUT    ADATA
 (1)  446F  D3  94              10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4471  7F                   4.0                  MOV     A,A             ;RETRY LINK
1632  4472  3E  07               7.0          MVI     A,BIT0!BIT1!BIT2
1633  4474                                    ROUT    EDATA
 (1)  4474  D3  95              10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4476  7F                   4.0                  MOV     A,A             ;RETRY LINK
1634  4477                                    ERRB    TST02X,T2CN4
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4477  CD  12  28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0011                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  447A  11                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  447B  00                                             .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  447C  CD  15  28        18.0                T2CN4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  447F  DA  D7  43        10.0                        JC      TST02X          ;LOOP ADDRESS IF LOOP SPECIFIED
1635                                ;>PORT SELECT SWITCH DOES NOT INDICATE MAINTENANCE
```

```
                                .SBTTL  TEST 03 - MASSBUS PORT SELECT SWITCH TEST - TU #2
4482                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*MASSBUS PORT SELECT SWITCH TEST - TU #2
4482                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE FUNCTIONALITY OF THE TU78 PORT SELECT SWITCH AND
                                ;*ASSOCIATED HARDWARE, FOR TU #2.
4482                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  IF TU PORT #2 NOT SELECTED BY USER
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  IF MANUAL INTERVENTION NOT ALLOWED
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  CALL SUBROUTINE CLEAR
                                ;*  SELECT TU PORT #2
                                ;*  SET PSB NOT, PS1 NOT, PS0 NOT FOR MTA #2
                                ;*  REQUEST THE USER TO SET TU78 #2 TO MASSBUS PORT 0
                                ;*  INPUT MTA #2 STATUS REGISTER B
                                ;*  IF PSB NOT AND PS1 NOT=1 AND BS0 NOT=0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  REQUEST THE USER TO SET TU78 #2 TO MASSBUS PORT 1
                                ;*  INPUT MTA #2 STATUS REGISTER B
                                ;*  IF PSB NOT AND PS0 NOT=1 AND PS1 NOT=0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  REQUEST THE USER TO SET TU78 #2 TO BOTH MASSBUS PORTS
                                ;*  INPUT MTA #2 STATUS REGISTER B
                                ;*  IF PSB NOT=0 AND PS0 NOT=1 AND PS1 NOT=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  REQUEST THE USER TO SET TU78 #2 TO MAINTENANCE MODE
                                ;*  INPUT MTS #2 STATUS REGISTER B
                                ;*  IF PSB NOT PS0 NOT AND PS1 NOT=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*ENDTST
4482                            SE
```

```
 (1)                    ; *****************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
1682                    ;*MTA2 MICRO TEST 03
1683                    ;*MTA2 MICRO ERROR 23
1684                    ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #2
1685                    ;*M8954, TU CABLE, M8955'S
1686                    ;*PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 0
1687                    ;*ACTUAL = NNNN
1688                    ;*EXPECTED = NNNN
1689                    ;*
1690                    ;*MTA2 MICRO TEST 03
1691                    ;*MTA2 MICRO ERROR 25
1692                    ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #2
1693                    ;*M8954, TU CABLE, M8955'S
1694                    ;*PORT SELECT DOES NOT INDICATE MASSBUS PORT 1
1695                    ;*ACTUAL = NNNN
1696                    ;*EXPECTED = NNNN
1697                    ;*
1698                    ;*MTA2 MICRO TEST 03
1699                    ;*MTA2 MICRO ERROR 27
1700                    ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #2
1701                    ;*M8954, TU CABLE, M8955'S
1702                    ;*PORT SELECT SWITCH DOES NOT INDICATE BOTH MASSBUS PORTS 0 AND 1
1703                    ;*ACTUAL = NNNN
1704                    ;*EXPECTED = NNNN
1705                    ;*
1706                    ;*MTA2 MICRO TEST 03
1707                    ;*MTA2 MICRO ERROR 31
1708                    ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #2
1709                    ;*M8954, TU CABLE, M8955'S
1710                    ;*PORT SELECT SWITCH DOES NOT INDICATE MAINTENANCE
1711                    ;*ACTUAL = NNNN
1712                    ;*EXPECTED = NNNN
1713  4482              S
 (1)                    ; *****************************************************************************
1714
1715  4482              TEST03: TESTX   @03                     ;INITIALIZE THE TEST
 (1)  4482  3E  03        7.0                    MVI     A,@03           ;DEFINE THE TEST NUMBER
 (1)  4484  CD  03  28   18.0                    CALL    TSET            ;SETUP THE TEST
1716                    ;%MTA MASSBUS PORT SELECT SWITCH TEST - TU #2
1717                    ;&M8954, TU CABLE, M8955'S
1718  4487  3A  BA  49   13.0  TST03X: LDA     UNITMP          ;GET THE UNIT MAP
1719  448A  E6  04        7.0           ANI     @04             ;TEST UNIT 2
1720  448C  CA  32  45   10.0           JZ      TEST04          ;NO-CHECK FOR TEST 04
1721  448F  CD  20  48   18.0           CALL    CLEAR           ;CLEAR THE TU PORTS
1722  4492  DB  E0       10.0           IN      INTSTA          ;SELECT TU PORT 2
1723  4494  E6  80        7.0           ANI     BIT7
1724  4496  F6  02        7.0           ORI     @02
1725  4498  D3  E0       10.0           OUT     MBSEL
1726  449A  3E  83        7.0           MVI     A,$83           ;SELECT MTA 2 REGISTER 3
1727  449C  D3  40       10.0           OUT     TCMD
1728  449E  3E  38        7.0           MVI     A,$38           ;SET THE PS0 NOT, PS1 NOT,
1729  44A0  D3  40       10.0           OUT     TCMD            ;AND PSB NOT BITS.
```

```
1730  44A2                                          QMSGM   TEST04
 (1)  44A2  CD  1E  28        18.0                 CALL    QUEM
 (1)        0012                                   MSGN=MSGN+1
 (1)  44A5  12                                     .BYTE   MSGN
 (1)  44A6  DA  32  45        10.0                 JC      TEST04
1731                                    ;>SET THE TU78 #2 PORT SELECT SWITCH TO MASSBUS PORT 0
1732  44A9  3E  82               7.0               MVI     A,$82           ;SELECT MTA 2 REGISTER 2
1733  44AB  D3  40              10.0               OUT     TCMD
1734  44AD  DB  40              10.0               IN      TSTS            ;GET THE TU PORT STATUS
1735  44AF  E6  07               7.0               ANI     $07             ;SAVE ONLY PORT SELECT BITS
1736  44B1  FE  06               7.0               CPI     BIT1!BIT2       ;COMPARE WITH EXPECTED
1737  44B3  CA  C3  44          10.0               JZ      T3CN1           ;BRANCH IF EQUAL
1738  44B6                                         ROUT    ADATA           ;SAVE THE ACTUAL DATA
 (1)  44B6  D3  94              10.0                       OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44B8  7F                   4.0                       MOV     A,A             ;RETRY LINK
1739  44B9  3E  06               7.0               MVI     A,BIT1!BIT2     ;SAVE THE EXPECTED DATA
1740  44BB                                         ROUT    EDATA
 (1)  44BB  D3  95              10.0                       OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44BD  7F                   4.0                       MOV     A,A             ;RETRY LINK
1741  44BE                                         ERRB    TST03X,T3CN1
 (1)                                             ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44BE  CD  12  28        18.0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0013                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44C1  13                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44C2  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  44C3  CD  15  28        18.0               T3CN1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44C6  DA  87  44        10.0                       JC      TST03X          ;LOOP ADDRESS IF LOOP SPECIFIED
1742                                    ;>PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 0
1743  44C9                                          QMSGM   TEST04
 (1)  44C9  CD  1E  28        18.0                 CALL    QUEM
 (1)        0014                                   MSGN=MSGN+1
 (1)  44CC  14                                     .BYTE   MSGN
 (1)  44CD  DA  32  45        10.0                 JC      TEST04
1744                                    ;>SET THE TU78 #2 PORT SELECT SWITCH TO MASSBUS PORT 1
1745  44D0  DB  40              10.0               IN      TSTS
1746  44D2  E6  07               7.0               ANI     $07
1747  44D4  FE  05               7.0               CPI     BIT0!BIT2
1748  44D6  CA  E6  44          10.0               JZ      T3CN2
1749  44D9                                         ROUT    ADATA
 (1)  44D9  D3  94              10.0                       OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44DB  7F                   4.0                       MOV     A,A             ;RETRY LINK
1750  44DC  3E  05               7.0               MVI     A,BIT0!BIT2
1751  44DE                                         ROUT    EDATA
 (1)  44DE  D3  95              10.0                       OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44E0  7F                   4.0                       MOV     A,A             ;RETRY LINK
1752  44E1                                         ERRB    TST03X,T3CN2
 (1)                                             ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44E1  CD  12  28        18.0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0015                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44E4  15                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44E5  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  44E6  CD  15  28        18.0               T3CN2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44E9  DA  87  44        10.0                       JC      TST03X          ;LOOP ADDRESS IF LOOP SPECIFIED
1753                                    ;>PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 1
```

```
                                        QMSGM   TEST04
(1) 44EC CD 1E 28       18.0            CALL    QUEM
(1)      0016                           MSGN=MSGN+1
(1) 44EF 16                             .BYTE   MSGN
(1) 44F0 DA 32 45       10.0            JC      TEST04
                                ;>SET THE TU78 #2 PORT SELECT SWITCH TO BOTH PORTS 0 AND 1
44F3 DB 40              10.0            IN      TSTS
44F5 E6 07               7.0            ANI     $07
44F7 FE 03               7.0            CPI     BIT0!BIT1
44F9 CA 09 45           10.0            JZ      T3CN3
44FC                                    ROUT    ADATA
(1) 44FC D3 94          10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
(1) 44FE 7F              4.0                    MOV     A,A             ;RETRY LINK
44FF 3E 03               7.0            MVI     A,BIT0!BIT1
4501                                    ROUT    EDATA
(1) 4501 D3 95          10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
(1) 4503 7F              4.0                    MOV     A,A             ;RETRY LINK
4504                                    ERRB    TST03X,T3CN3
(1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
(1) 4504 CD 12 28       18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
(1)      0017                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1) 4507 17                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
(1) 4508 00                                             .BYTE                   ;PRINT ROUTINE NUMBER
(1) 4509 CD 15 28       18.0                    T3CN3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
(1) 450C DA 87 44       10.0                            JC      TST03X          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>PORT SELECT SWITCH DOES NOT INDICATE BOTH MASSBUS PORTS 0 AND 1
450F                                    QMSGM   TEST04
(1) 450F CD 1E 28       18.0            CALL    QUEM
(1)      0018                           MSGN=MSGN+1
(1) 4512 18                             .BYTE   MSGN
(1) 4513 DA 32 45       10.0            JC      TEST04
                                ;>SET THE TU78 #2 PORT SELECT SWITCH TO MAINTENANCE
4516 DB 40              10.0            IN      TSTS
4518 E6 07               7.0            ANI     $07
451A FE 07               7.0            CPI     BIT0!BIT1!BIT2
451C CA 2C 45           10.0            JZ      T3CN4
451F                                    ROUT    ADATA
(1) 451F D3 94          10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
(1) 4521 7F              4.0                    MOV     A,A             ;RETRY LINK
4522 3E 07               7.0            MVI     A,BIT0!BIT1!BIT2
4524                                    ROUT    EDATA
(1) 4524 D3 95          10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
(1) 4526 7F              4.0                    MOV     A,A             ;RETRY LINK
4527                                    ERRB    TST03X,T3CN4
(1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
(1) 4527 CD 12 28       18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
(1)      0019                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
(1) 452A 19                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
(1) 452B 00                                             .BYTE                   ;PRINT ROUTINE NUMBER
(1) 452C CD 15 28       18.0                    T3CN4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
(1) 452F DA 87 44       10.0                            JC      TST03X          ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>PORT SELECT SWITCH DOES NOT INDICATE MAINTENANCE
```

```
                                .SBTTL  TEST 04 - MASSBUS PORT SELECT SWITCH TEST - TU #3
4532                            ST
(1)                             ;*********************************************************************
(1)                             ;*TEST TITLE
(1)                             ;*---- -----
                                ;*MASSBUS PORT SELECT SWITCH TEST - TU #3
4532                            SD
(1)                             ;*********************************************************************
(1)                             ;*DESCRIPTION
(1)                             ;*-----------
                                ;*THIS TEST CHECKS THE FUNCTIONALITY OF THE TU78 PORT SELECT SWITCH AND
                                ;*ASSOCIATED HARDWARE, FOR TU #3.
4532                            SP
(1)                             ;*********************************************************************
(1)                             ;*PROCEDURE
(1)                             ;*---------
                                ;*BGNTST
                                ;*  IF TU PORT #3 NOT SELECTED BY USER
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  IF MANUAL INTERVENTION NOT ALLOWED
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  CALL SUBROUTINE CLEAR
                                ;*  SELECT TU PORT #3
                                ;*  SET PSB NOT, PS1 NOT, PS0 NOT FOR MTA #3
                                ;*  REQUEST THE USER TO SET TU78 #3 TO MASSBUS PORT 0
                                ;*  INPUT MTA #3 STATUS REGISTER B
                                ;*  IF PSB NOT AND PS1 NOT=1 AND BS0 NOT=0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  REQUEST THE USER TO SET TU78 #3 TO MASSBUS PORT 1
                                ;*  INPUT MTA #3 STATUS REGISTER B
                                ;*  IF PSB NOT AND PS0 NOT=1 AND PS1 NOT=0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  REQUEST THE USER TO SET TU78 #3 TO BOTH MASSBUS PORTS
                                ;*  INPUT MTA #3 STATUS REGISTER B
                                ;*  IF PSB NOT=0 AND PS0 NOT=1 AND PS1 NOT=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  REQUEST THE USER TO SET TU78 #3 TO MAINTENANCE MODE
                                ;*  INPUT MTA #3 STATUS REGISTER B
                                ;*  IF PSB NOT PS0 NOT AND PS1 NOT=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*ENDTST
4532                            SE
```

```
 (1)                                ; ***************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1822                                ;*MTA2 MICRO TEST 04
1823                                ;*MTA2 MICRO ERROR 33
1824                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #3
1825                                ;*M8954, TU CABLE, M8955'S
1826                                ;*PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 0
1827                                ;*ACTUAL = NNNN
1828                                ;*EXPECTED = NNNN
1829                                ;*
1830                                ;*MTA2 MICRO TEST 04
1831                                ;*MTA2 MICRO ERROR 35
1832                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #3
1833                                ;*M8954, TU CABLE, M8955'S
1834                                ;*PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 1
1835                                ;*ACTUAL = NNNN
1836                                ;*EXPECTED = NNNN
1837                                ;*
1838                                ;*MTA2 MICRO TEST 04
1839                                ;*MTA2 MICRO ERROR 37
1840                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #3
1841                                ;*M8954, TU CABLE, M8955'S
1842                                ;*PORT SELECT SWITCH DOES NOT INDICATE BOTH MASSBUS PORTS 0 AND 1
1843                                ;*ACTUAL = NNNN
1844                                ;*EXPECTED = NNNN
1845                                ;*
1846                                ;*MTA2 MICRO TEST 04
1847                                ;*MTA2 MICRO ERROR 41
1848                                ;*MTA MASSBUS PORT SELECT SWITCH TEST - TU #3
1849                                ;*M8954, TU CABLE, M8955'S
1850                                ;*PORT SELECT SWITCH DOES NOT INDICATE MAINTENANCE
1851                                ;*ACTUAL = NNNN
1852                                ;*EXPECTED = NNNN
1853  4532                          S
 (1)                                ; ***************************************************************************
1854
1855  4532                          TEST04: TESTX   @04               ;INITIALIZE THE TEST
 (1)  4532  3E  04          7.0                 MVI     A,@04          ;DEFINE THE TEST NUMBER
 (1)  4534  CD  03  28     18.0                 CALL    TSET           ;SETUP THE TEST
1856                                ;%MTA MASSBUS PORT SELECT SWITCH TEST - TU #3
1857                                ;&M8954, TU CABLE, M8955'S
1858  4537  3A  BA  49     13.0    TST04X: LDA     UNITMP            ;GET THE UNIT MAP
1859  453A  E6  08           7.0            ANI     @010              ;TEST UNIT 3
1860  453C  CA  E2  45      10.0            JZ      TEST05            ;NO-EXIT TEST
1861  453F  CD  20  48      18.0            CALL    CLEAR             ;CLEAR THE TU PORTS
1862  4542  DB  E0           10.0            IN      INTS1A            ;SELECT TU PORT 3
1863  4544  E6  80           7.0            ANI     BIT7
1864  4546  F6  03           7.0            ORI     @03
1865  4548  D3  E0          10.0            OUT     MBSEL
1866  454A  3E  83           7.0            MVI     A,$83             ;SELECT MTA 3 REGISTER 3
1867  454C  D3  40          10.0            OUT     TCMD
1868  454E  3E  38           7.0            MVI     A,$38             ;SET THE PS0 NOT, PS1 NOT,
1869  4550  D3  40          10.0            OUT     TCMD              ;AND PSB NOT BITS.
```

```
 1870  4552                                          QMSGM   TEST05
  (1)  4552  CD  1E  28        18.0                 CALL    QUEM
  (1)       001A                                    MSGN=MSGN+1
  (1)  4555  1A                                     .BYTE   MSGN
  (1)  4556  DA  E2  45        10.0                 JC      TEST05
 1871                                    ;>SET THE TU78 #3 PORT SELECT SWITCH TO MASSBUS PORT 0
 1872  4559  3E  82                7.0              MVI     A,$82           ;SELECT MTA 3 REGISTER 2
 1873  455B  D3  40               10.0              OUT     TCMD
 1874  455D  DB  40               10.0              IN      TSTS            ;GET THE TU PORT STATUS
 1875  455F  E6  07                7.0              ANI     $07             ;SAVE ONLY PORT SELECT BITS
 1876  4561  FE  06                7.0              CPI     BIT1!BIT2       ;COMPARE WITH EXPECTED
 1877  4563  CA  73  45           10.0              JZ      T4CN1           ;BRANCH IF EQUAL
 1878  4566                                          ROUT    ADATA           ;SAVE THE ACTUAL DATA
  (1)  4566  D3  94               10.0                      OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  4568  7F                    4.0                      MOV     A,A             ;RETRY LINK
 1879  4569  3E  06                7.0              MVI     A,BIT1!BIT2     ;SAVE THE EXPECTED DATA
 1880  456B                                          ROUT    EDATA
  (1)  456B  D3  95               10.0                      OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  456D  7F                    4.0                      MOV     A,A             ;RETRY LINK
 1881  456E                                          ERRB    TST04X,T4CN1
  (1)                                             ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  456E  CD  12  28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       001B                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4571  1B                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4572  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  4573  CD  15  28        18.0                T4CN1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  4576  DA  37  45        10.0                        JC      TST04X          ;LOOP ADDRESS IF LOOP SPECIFIED
 1882                                    ;>PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 0
 1883  4579                                          QMSGM   TEST05
  (1)  4579  CD  1E  28        18.0                 CALL    QUEM
  (1)       001C                                    MSGN=MSGN+1
  (1)  457C  1C                                     .BYTE   MSGN
  (1)  457D  DA  E2  45        10.0                 JC      TEST05
 1884                                    ;>SET THE TU78 #3 PORT SELECT SWITCH TO MASSBUS PORT 1
 1885  4580  DB  40               10.0              IN      TSTS
 1886  4582  E6  07                7.0              ANI     $07
 1887  4584  FE  05                7.0              CPI     BIT0!BIT2
 1888  4586  CA  96  45           10.0              JZ      T4CN2
 1889  4589                                          ROUT    ADATA
  (1)  4589  D3  94               10.0                      OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  458B  7F                    4.0                      MOV     A,A             ;RETRY LINK
 1890  458C  3E  05                7.0              MVI     A,BIT0!BIT2
 1891  458E                                          ROUT    EDATA
  (1)  458E  D3  95               10.0                      OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  4590  7F                    4.0                      MOV     A,A             ;RETRY LINK
 1892  4591                                          ERRB    TST04X,T4CN2
  (1)                                             ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  4591  CD  12  28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       001D                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4594  1D                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4595  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  4596  CD  15  28        18.0                T4CN2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  4599  DA  37  45        10.0                        JC      TST04X          ;LOOP ADDRESS IF LOOP SPECIFIED
 1893                                    ;>PORT SELECT SWITCH DOES NOT INDICATE MASSBUS PORT 1
```

```
1894  459C                                          QMSGM   TEST05
 (1)  459C  CD  1E  28        18.0                 CALL    QUEM
 (1)        001E                                   MSGN=MSGN+1
 (1)  459F  1E                                     .BYTE   MSGN
 (1)  45A0  DA  E2  45        10.0                 JC      TEST05
1895                                    ;>SET THE TU78 #3 PORT SELECT SWITCH TO BOTH PORTS 0 AND 1
1896  45A3  DB  40              10.0                 IN      TSTS
1897  45A5  E6  07               7.0                 ANI     $07
1898  45A7  FE  03               7.0                 CPI     BIT0.BIT1
1899  45A9  CA  B9  45        10.0                 JZ      T4CN3
1900  45AC                                          ROUT    ADATA
 (1)  45AC  D3  94              10.0                         OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  45AE  7F                   4.0                         MOV     A,A             ;RETRY LINK
1901  45AF  3E  03               7.0                 MVI     A,BIT0!BIT1
1902  45B1                                          ROUT    EDATA
 (1)  45B1  D3  95              10.0                         OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  45B3  7F                   4.0                         MOV     A,A             ;RETRY LINK
1903  45B4                                          ERRB    TST04X,T4CN3
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  45B4  CD  12  28        18.0                         CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        001F                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45B7  1F                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45B8  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45B9  CD  15  28        18.0                 T4CN3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45BC  DA  37  45        10.0                         JC      TST04X          ;LOOP ADDRESS IF LOOP SPECIFIED
1904                                    ;>PORT SELECT SWITCH DOES NOT INDICATE BOTH MASSBUS PORTS 0 AND 1
1905  45BF                                          QMSGM   TEST05
 (1)  45BF  CD  1E  28        18.0                 CALL    QUEM
 (1)        0020                                   MSGN=MSGN+1
 (1)  45C2  20                                     .BYTE   MSGN
 (1)  45C3  DA  E2  45        10.0                 JC      TEST05
1906                                    ;>SET THE TU78 #3 PORT SELECT SWITCH TO MAINTENANCE
1907  45C6  DB  40              10.0                 IN      TSTS
1908  45C8  E6  07               7.0                 ANI     $07
1909  45CA  FE  07               7.0                 CPI     BIT0!BIT1!BIT2
1910  45CC  CA  DC  45        10.0                 JZ      T4CN4
1911  45CF                                          ROUT    ADATA
 (1)  45CF  D3  94              10.0                         OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  45D1  7F                   4.0                         MOV     A,A             ;RETRY LINK
1912  45D2  3E  07               7.0                 MVI     A,BIT0!BIT1!BIT2
1913  45D4                                          ROUT    EDATA
 (1)  45D4  D3  95              10.0                         OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  45D6  7F                   4.0                         MOV     A,A             ;RETRY LINK
1914  45D7                                          ERRB    TST04X,T4CN4
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  45D7  CD  12  28        18.0                         CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0021                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45DA  21                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45DB  00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45DC  CD  15  28        18.0                 T4CN4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45DF  DA  37  45        10.0                         JC      TST04X          ;LOOP ADDRESS IF LOOP SPECIFIED
1915                                    ;>PORT SELECT SWITCH DOES NOT INDICATE MAINTENANCE
```

```
                                .SBTTL  TEST 05 - LOOP THRESHOLD CONTROL BITS - MTA #0

45E2                            ST
 (1)                            ;*****************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*----- -----
                                ;*LOOP THRESHOLD CONTROL BITS - MTA #0
45E2                            SD
 (1)                            ;*****************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE THRESHOLD CONTROL BITS FOR MTA #0 USEING
                                ;*THE THRESHOLD CONTROL BIT LOOP BACK LOGIC ON THE MTA BOARD.
45E2                            SP
 (1)                            ;*****************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #0 NOT SELECTED BY USER
                                ;*     :    THEN-NEXT TEST
                                ;*     :    ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT PORT 0
                                ;*     CLEAR THE THRESHOLD DATA
                                ;*     BGNDO
                                ;*     :    ISSUE 3 BITS OF THE CONTENTS OF THRESHOLD DATA TO MTA #0 REGISTER #3
                                ;*     :    INPUT MTA #0 REGISTER #5
                                ;*     :    IF THRESHOLD DATA READ=THRESHOLD DATA
                                ;*     :    :    THEN-CONTINUE
                                ;*     :    :    ELSE-ERROR
                                ;*     :    ENDIF
                                ;*     :    INCREMENT THE THRESHOLD DATA
                                ;*     :    DO UNTIL THE THRESHOLD DATA-10(8)
                                ;*     ENDDO
                                ;*     CALL SUBROUTINE CLEAR
                                ;*ENDTST
45E2                            SE
 (1)                            ;*****************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA2 MICRO TEST 05
                                ;*MTA2 MICRO ERROR 42
                                ;*MTA LOOP THRESHOLD CONTROL BITS - MTA #0
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN TO MTA #0 THRESHOLD NOT = DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
45E2                            S
 (1)                            ;*****************************************************************************
```

```
1955  45E2                                  TEST05: TESTX   @05              ;SET UP THE TEST NUMBER
 (1)  45E2  3E  05              7.0                        MVI     A,@05        ;DEFINE THE TEST NUMBER
 (1)  45E4  CD  03  28         18.0                        CALL    TSET         ;SETUP THE TEST
1956
1957                                       ;%MTA LOOP THRESHOLD CONTROL BITS - MTA #0
1958                                       ;&M8954, M8955'S
1958  45E7  3A  BA  49         13.0 TST05X: LDA     UNITMP          ;GET THE UNIT MAP
1959  45EA  E6  01                7.0          ANI     @001            ;TEST UNIT 0?
1960  45EC  CA  40  46          10.0          JZ      TEST06          ;NO-CHECK FOR TEST06
1961                                                                   ;YES-RUN THE TEST
1962  45EF  CD  20  48          18.0          CALL    CLEAR           ;CLEAN UP THE PORTS
1963  45F2  DB  E0               10.0          IN      INTSTA          ;GET SELECT STATUS
1964  45F4  E6  80                7.0          ANI     BIT7
1965  45F6  D3  E0               10.0          OUT     MBSEL           ;SELECT PORT 0
1966  45F8  21  B9  49           10.0          LXI     H,DATA          ;LOAD ADDRESS OF DATA
1967  45FB  AF                    4.0          XRA     A               ;CLEAR THE ACCUMULATOR
1968  45FC  77                    7.0          MOV     M,A             ;STORE IN DATA
1969                                                                   ;
1970  45FD  3E  83                7.0 ALP01:   MVI     A,@203          ;SELECT MTA #0 REGISTER 3
1971  45FF  D3  40               10.0          OUT     TCMD
1972  4601  7E                    7.0          MOV     A,M             ;GET THE THRESHOLD DATA
1973  4602  D3  40               10.0          OUT     TCMD            ;LOAD MTA REGISTER 0
1974  4604  3E  85                7.0          MVI     A,@205          ;SELECT MTA #0 REGISTER 5
1975  4606  D3  40               10.0          OUT     TCMD
1976  4608  7E                    7.0          MOV     A,M             ;GET THE THRESHOLD DATA
1977  4609  47                    4.0          MOV     B,A             ;SAVE FOR DATA COMPARE
1978  460A  DB  40               10.0          IN      TSTS            ;READ THE THRESHOLD REGISTER
1979  460C  E6  07                7.0          ANI     $07             ;REMOVE THE JUNK BITS
1980  460E  B8                    4.0          CMP     B               ;COMPARE ACTUAL AND EXP.
1981  460F  CA  1E  46           10.0          JZ      ALP05           ;EQUAL - CONTINUE
1982  4612                                     ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  4612  D3  94               10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4614  7F                    4.0                  MOV     A,A             ;RETRY LINK
1983  4615  7E                    7.0          MOV     A,M             ;GET EXP. DATA
1984  4616                                     ROUT    EDATA           ;STORE EXP. DATA
 (1)  4616  D3  95               10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4618  7F                    4.0                  MOV     A,A             ;RETRY LINK
1985  4619                                     ERRB    ALP01,ALP05,0
 (1)                                         ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4619  CD  12  28          18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0022                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  461C  22                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  461D  00                                        .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  461E  CD  15  28          18.0          ALP05:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4621  DA  FD  45          10.0                  JC      ALP01           ;LOOP ADDRESS IF LOOP SPECIFIED
1986                                       ;>DATA WRITTEN TO MTA #0 THRESHOLD NOT = DATA READ
1987  4624  7E                    7.0          MOV     A,M             ;GET DATA
1988  4625  3C                    4.0          INR     A               ;UPDATE THE DATA BYTE
1989  4626  77                    7.0          MOV     M,A             ;SAVE UPDATED DATA
1990  4627  FE  08                7.0          CPI     @10             ;DONE?
1991  4629  C2  FD  45           10.0          JNZ     ALP01           ;NO - CONTINUE UNTIL DONE
1992
```

```
 1994  462C                                        ENDTST  TST05X
  (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  462C                                       REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  462C  CD  06  28        18.0                                 CALL    REQST
  (2)  462F  00                                                     .BYTE                   ;DATA PATTERN NUMBER
  (2)  4630  00  00                                                 .WORD                   ;SYSTEM "" COUNT
  (2)  4632  00  00                                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  4634  00                                                     .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  4635  07                                                     .BYTE   7               ;REQUEST CODE
  (1)  4636  3A  9A  4F        13.0               LDA     ITERA           ;GET ITERATION COUNT
  (1)  4639  3D                 4.0               DCR     A               ;DOWNCOUNT
  (1)  463A  32  9A  4F        13.0               STA     ITERA           ;SAVE COUNT
  (1)  463D  F2  E7  45        10.0               JP      TST05X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 06 - LOOP THRESHOLD CONTROL BITS - MTA #1

4640                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP THRESHOLD CONTROL BITS - MTA #1
4640                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE THRESHOLD CONTROL BITS FOR MTA #1 USEING
                                ;*THE THRESHOLD CONTROL BIT LOOP BACK LOGIC ON THE MTA BOARD.
4640                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #1 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT PORT 1
                                ;*    CLEAR THE THRESHOLD DATA
                                ;*    BGNDO
                                ;*    :   ISSUE 3 BITS OF THE CONTENTS OF THRESHOLD DATA TO MTA #1 REGISTE
                                ;*    :   INPUT MTA #1 REGISTER #5
                                ;*    :   IF THRESHOLD DATA READ=THRESHOLD DATA
                                ;*    :   :   THEN-CONTINUE
                                ;*    :   :   ELSE-ERROR
                                ;*    :   ENDIF
                                ;*    :   INCREMENT THE THRESHOLD DATA
                                ;*    :   DO UNTIL THE THRESHOLD DATA=10(8)
                                ;*    ENDDO
                                ;*    CALL SUBROUTINE CLEAR
                                ;*ENDTST
4640                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA2 MICRO TEST 06
                                ;*MTA2 MICRO ERROR 43
                                ;*MTA LOOP THRESHOLD CONTROL BITS - MTA #1
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN TO MTA #1 THRESHOLD NOT = DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
4640                            S
                                ; ***************************************************************************
```

```
2034  4640                              TEST06: TESTX   @06             ;SET UP THE TEST NUMBER
 (1)  4640  3E  06            7.0                      MVI     A,@06           ;DEFINE THE TEST NUMBER
 (1)  4642  CD  03  28       18.0                      CALL    TSET            ;SETUP THE TEST
2035                              ;%MTA LOOP THRESHOLD CONTROL BITS - MTA #1
2036                              ;&M8954, M8955'S
2037  4645  3A  BA  49       13.0 TST06X: LDA     UNITMP          ;GET THE UNIT MAP
2038  4648  E6  02             7.0          ANI     @002            ;TEST UNIT 1?
2039  464A  CA  A0  46       10.0          JZ      TEST07          ;NO-CHECK FOR TEST07
2040                                                               ;YES-RUN THE TEST
2041  464D  CD  20  48       18.0          CALL    CLEAR           ;CLEAN UP THE PORTS
2042  4650  DB  E0            10.0          IN      INTSTA          ;GET SELECT STATUS
2043  4652  E6  80             7.0          ANI     BIT7
2044  4654  F6  01             7.0          ORI     @1              ;OR IN THE PORT SELECT BITS
2045  4656  D3  E0            10.0          OUT     MBSEL           ;SELECT PORT 1
2046  4658  21  B9  49       10.0          LXI     H,DATA          ;LOAD ADDRESS OF DATA
2047  465B  AF             4.0          XRA     A               ;CLEAR THE ACCUMULATOR
2048  465C  77             7.0          MOV     M,A             ;STORE IN DATA
2049                                                               ;
2050  465D  3E  83             7.0 ALP02:  MVI     A,@203          ;SELECT MTA #1 REGISTER 3
2051  465F  D3  40            10.0          OUT     TCMD
2052  4661  7E             7.0          MOV     A,M             ;GET THE THRESHOLD DATA
2053  4662  D3  40            10.0          OUT     TCMD            ;LOAD MTA REGISTER 0
2054  4664  3E  85             7.0          MVI     A,@205          ;SELECT MTA #1 REGISTER 5
2055  4666  D3  40            10.0          OUT     TCMD
2056  4668  7E             7.0          MOV     A,M             ;GET THE THRESHOLD DATA
2057  4669  47             4.0          MOV     B,A             ;SAVE FOR DATA COMPARE
2058  466A  DB  40            10.0          IN      TSTS            ;READ THE THRESHOLD REGISTER
2059  466C  E6  07             7.0          ANI     $07             ;REMOVE THE JUNK BITS
2060  466E  B8             4.0          CMP     B               ;COMPARE ACTUAL AND EXP.
2061  466F  CA  7E  46       10.0          JZ      ALP06           ;EQUAL - CONTINUE
2062  4672                               ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  4672  D3  94            10.0                      OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4674  7F             4.0                      MOV     A,A             ;RETRY LINK
2063  4675  7E             7.0          MOV     A,M             ;GET EXP. DATA
2064  4676                               ROUT    EDATA           ;STORE EXP. DATA
 (1)  4676  D3  95            10.0                      OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4678  7F             4.0                      MOV     A,A             ;RETRY LINK
2065  4679                               ERRB    ALP02,ALP06,0
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4679  CD  12  28       18.0                      CALL    ERLPB           ;PROCESS ERROR - DC 2.3
 (1)       0023                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  467C  23                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  467D  00                                        .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  467E  CD  15  28       18.0                ALP06:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4681  DA  5D  46       10.0                      JC      ALP02           ;LOOP ADDRESS IF LOOP SPECIFIED
2066                              ;>DATA WRITTEN TO MTA #1 THRESHOLD NOT = DATA READ
2067  4684  7E             7.0          MOV     A,M             ;GET DATA
2068  4685  3C             4.0          INR     A               ;UPDATE THE DATA BYTE
2069  4686  77             7.0          MOV     M,A             ;SAVE UPDATED DATA
2070  4687  FE  08             7.0          CPI     @10             ;DONE?
2071  4689  C2  5D  46       10 0          JNZ     ALP02           ;NO - CONTINUE UNTIL DONE
2072
```

```
2074  468C                                    ENDTST  TST06X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  468C                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  468C  CD  06  28        18.0                          CALL    REQST
 (2)  468F  00                                              .BYTE                   ;DATA PATTERN NUMBER
 (2)  4690  00  00                                          .WORD                   ;SYSTEM '"' COUNT
 (2)  4692  00  00                                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4694  00                                              .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4695  07                                              .BYTE   7               ;REQUEST CODE
 (1)  4696  3A  9A  4F        13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4699  3D                 4.0          DCR     A               ;DOWNCOUNT
 (1)  469A  32  9A  4F        13.0          STA     ITERA           ;SAVE COUNT
 (1)  469D  F2  45  46        10.0          JP      TST06X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 07 - LOOP THRESHOLD CONTROL BITS - MTA #2

                                ST
                                ; ******************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*LOOP THRESHOLD CONTROL BITS - MTA #2
                                SD
                                ; ******************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE THRESHOLD CONTROL BITS FOR MTA #2 USEING
                                ;*THE THRESHOLD CONTROL BIT LOOP BACK LOGIC ON THE MTA BOARD.
                                SP
                                ; ******************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #2 NOT SELECTED BY USER
                                ;*    :   THEN-NEXT TEST
                                ;*    :   ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SELECT PORT 2
                                ;*    CLEAR THE THRESHOLD DATA
                                ;*    BGNDO
                                ;*    :   ISSUE 3 BITS OF THE CONTENTS OF THRESHOLD DATA TO MTA #2 REGISTER #3
                                ;*    :   INPUT MTA #2 REGISTER #5
                                ;*    :   IF THRESHOLD DATA READ=THRESHOLD DATA
                                ;*    :   :   THEN-CONTINUE
                                ;*    :   :   ELSE-ERROR
                                ;*    :   ENDIF
                                ;*    :   INCREMENT THE THRESHOLD DATA
                                ;*    :   DO UNTIL THE THRESHOLD DATA=10(8)
                                ;*    ENDDO
                                ;*    CALL SUBROUTINE CLEAR
                                ;*ENDTST
                                SE
                                ; ******************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA2 MICRO TEST 07
                                ;*MTA2 MICRO ERROR 44
                                ;*MTA LOOP THRESHOLD CONTROL BITS - MTA #2
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN TO MTA #2 THRESHOLD NOT = DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                S
                                ; ******************************************************************************
```

```
2114  46A0                                  TEST07: TEST%   @07             ;SET UP THE TEST NUMBER
 (1)  46A0  3E  07              7.0                          MVI     A,@07           ;DEFINE THE TEST NUMBER
 (1)  46A2  CD  03  28          18.0                         CALL    TSET            ;SETUP THE TEST
2115                                  ;%MTA LOOP THRESHOLD CONTROL BITS - MTA #2
2116                                  ;&M8954, M8955'S
2117  46A5  3A  BA  49          13.0 TST07X: LDA     UNITMP          ;GET THE UNIT MAP
2118  46A8  E6  04                7.0          ANI     @004            ;TEST UNIT 2?
2119  46AA  CA  00  47           10.0          JZ      TEST10          ;NO-CHECK FOR TEST10
2120                                                                   ;YES-RUN THE TEST
2121  46AD  CD  20  48           18.0          CALL    CLEAR           ;CLEAN UP THE PORTS
2122  46B0  DB  E0                10.0          IN      INTSTA          ;GET SELECT STATUS
2123  46B2  E6  80                 7.0          ANI     BIT7
2124  46B4  F6  02                 7.0          ORI     @2
2125  46B6  D3  E0                10.0          OUT     MBSEL           ;SELECT PORT 2
2126  46B8  21  B9  49            10.0          LXI     H,DATA          ;LOAD ADDRESS OF DATA
2127  46BB  AF                     4.0          XRA     A               ;CLEAR THE ACCUMULATOR
2128  46BC  77                     7.0          MOV     M,A             ;STORE IN DATA
2129                                                                   ;
2130  46BD  3E  83                 7.0 ALP03:   MVI     A,@203          ;SELECT MTA #2 REGISTER 3
2131  46BF  D3  40                10.0          OUT     TCMD
2132  46C1  7E                     7.0          MOV     A,M             ;GET THE THRESHOLD DATA
2133  46C2  D3  40                10.0          OUT     TCMD            ;LOAD MTA REGISTER 0
2134  46C4  3E  85                 7.0          MVI     A,@205          ;SELECT MTA #2 REGISTER 5
2135  46C6  D3  40                10.0          OUT     TCMD
2136  46C8  7E                     7.0          MOV     A,M             ;GET THE THRESHOLD DATA
2137  46C9  47                     4.0          MOV     B,A             ;SAVE FOR DATA COMPARE
2138  46CA  DB  40                10.0          IN      TSTS            ;READ THE THRESHOLD REGISTER
2139  46CC  E6  07                 7.0          ANI     $07             ;REMOVE THE JUNK BITS
2140  46CE  B8                     4.0          CMP     B               ;COMPARE ACTUAL AND EXP.
2141  46CF  CA  DE  46            10.0          JZ      ALP07           ;EQUAL - CONTINUE
2142  46D2                                     ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  46D2  D3  94              10.0                         OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46D4  7F                   4.0                         MOV     A,A             ;RETRY LINK
2143  46D5  7E                     7.0          MOV     A,M             ;GET EXP. DATA
2144  46D6                                     ROUT    EDATA           ;STORE EXP. DATA
 (1)  46D6  D3  95              10.0                         OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  46D8  7F                   4.0                         MOV     A,A             ;RETRY LINK
2145  46D9                                     ERRB    ALP03,ALP07,0
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46D9  CD  12  28          18.0                         CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0024                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46DC  24                                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46DD  00                                             .BYTE   0               ;PRINT ROUTINE NUMBER
 (1)  46DE  CD  15  28          18.0                ALP07:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46E1  DA  BD  46          10.0                        JC      ALP03           ;LOOP ADDRESS IF LOOP SPECIFIED
2146                                  ;>DATA WRITTEN TO MTA #2 THRESHOLD NOT = DATA READ
2147  46E4  7E                     7.0          MOV     A,M             ;GET DATA
2148  46E5  3C                     4.0          INR     A               ;UPDATE THE DATA BYTE
2149  46E6  77                     7.0          MOV     M,A             ;SAVE UPDATED DATA
2150  46E7  FE  08                 7.0          CPI     @10             ;DONE?
2151  46E9  C2  BD  46            10.0          JNZ     ALP03           ;NO - CONTINUE UNTIL DONE
2152
```

```
2154  46EC                                    ENDTST  TST07X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  46EC                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  46EC  CD  06  28      18.0                              CALL    REQST
 (2)  46EF  00                                                .BYTE                   ;DATA PATTERN NUMBER
 (2)  46F0  00  00                                            .WORD                   ;SYSTEM '"' COUNT
 (2)  46F2  00  00                                            .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  46F4  00                                                .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  46F5  07                                                .BYTE   7               ;REQUEST CODE
 (1)  46F6  3A  9A  4F      13.0          LDA     ITERA                   ;GET ITERATION COUNT
 (1)  46F9  3D                4.0          DCR     A                       ;DOWNCOUNT
 (1)  46FA  32  9A  4F      13.0          STA     ITERA                   ;SAVE COUNT
 (1)  46FD  F2  A5  46      10.0          JP      TST07X                  ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 10 - LOOP THRESHOLD CONTROL BITS - MTA #3

4700                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*LOOP THRESHOLD CONTROL BITS - MTA #3
4700                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE THRESHOLD CONTROL BITS FOR MTA #3 USEING
                                ;*THE THRESHOLD CONTROL BIT LOOP BACK LOGIC ON THE MTA BOARD.
4700                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*     IF TU PORT #3 NOT SELECTED BY USER
                                ;*     :   THEN-NEXT TEST
                                ;*     :   ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SELECT PORT 3
                                ;*     CLEAR THE THRESHOLD DATA
                                ;*     BGNDO
                                ;*     :   ISSUE 3 BITS OF THE CONTENTS OF THRESHOLD DATA TO MTA #3 REGISTER #3
                                ;*     :   INPUT MTA #3 REGISTER #5
                                ;*     :   IF THRESHOLD DATA READ=THRESHOLD DATA
                                ;*     :   :   THEN-CONTINUE
                                ;*     :   :   ELSE-ERROR
                                ;*     :   ENDIF
                                ;*     :   INCREMENT THE THRESHOLD DATA
                                ;*     :   DO UNTIL THE THRESHOLD DATA=10(8)
                                ;*     ENDDO
                                ;*     CALL SUBROUTINE CLEAR
                                ;*ENDTST
4700                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA2 MICRO TEST 10
                                ;*MTA2 MICRO ERROR 45
                                ;*MTA LOOP THRESHOLD CONTROL BITS - MTA #3
                                ;*M8954, M8955'S
                                ;*DATA WRITTEN TO MTA #3 THRESHOLD NOT - DATA READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
4700                            S
 (1)                            ; ************************************************************************
```

```
2194  4700                              TEST10: TESTX   @10              ;SET UP THE TEST NUMBER
 (1)  4700  3E  08            7.0                        MVI     A,@10          ;DEFINE THE TEST NUMBER
 (1)  4702  CD  03  28       18.0                        CALL    TSET           ;SETUP THE TEST
2195                                ;%MTA LOOP THRESHOLD CONTROL BITS - MTA #3
2196                                ;&M8954, M8955'S
2197  4705  3A  BA  49       13.0  TST10X: LDA     UNITMP           ;GET THE UNIT MAP
2198  4708  E6  08            7.0         ANI     @010             ;TEST UNIT 3?
2199  470A  CA  60  47        10.0         JZ      TEST11           ;NO-CHECK FOR TEST11
2200                                                                 ;YES-RUN THE TEST
2201  470D  CD  20  48        18.0         CALL    CLEAR            ;CLEAN UP THE PORTS
2202  4710  DB  E0             10.0         IN      INTSTA           ;GET SELECT STATUS
2203  4712  E6  80              7.0         ANI     BIT7
2204  4714  F6  03              7.0         ORI     @3
2205  4716  D3  E0             10.0         OUT     MBSEL            ;SELECT PORT 3
2206  4718  21  B9  49         10.0         LXI     H,DATA           ;LOAD ADDRESS OF DATA
2207  471B  AF                  4.0         XRA     A                ;CLEAR THE ACCUMULATOR
2208  471C  77                  7.0         MOV     M,A              ;STORE IN DATA
2209                                                                 ;
2210  471D  3E  83              7.0  ALP04:  MVI     A,@203           ;SELECT MTA #3 REGISTER 3
2211  471F  D3  40             10.0         OUT     TCMD
2212  4721  7E                  7.0         MOV     A,M              ;GET THE THRESHOLD DATA
2213  4722  D3  40             10.0         OUT     TCMD             ;LOAD MTA REGISTER 0
2214  4724  3E  85              7.0         MVI     A,@205           ;SELECT MTA #3 REGISTER 5
2215  4726  D3  40             10.0         OUT     TCMD
2216  4728  7E                  7.0         MOV     A,M              ;GET THE THRESHOLD DATA
2217  4729  47                  4.0         MOV     B,A              ;SAVE FOR DATA COMPARE
2218  472A  DB  40             10.0         IN      TSTS             ;READ THE THRFSHOLD REGISTER
2219  472C  E6  07              7.0         ANI     $07              ;REMOVE THE JUNK BITS
2220  472E  B8                  4.0         CMP     B                ;COMPARE ACTUAL AND EXP.
2221  472F  CA  3E  47        10.0         JZ      ALP08            ;EQUAL - CONTINUE
2222  4732                               ROUT    ADATA              ;STORE ACTUAL DATA
 (1)  4732  D3  94            10.0                        OUT     ADATA          ;WRITE AC INTO ADATA
 (1)  4734  7F                 4.0                        MOV     A,A            ;RETRY LINK
2223  4735  7E                  7.0         MOV     A,M              ;GET EXP. DATA
2224  4736                               ROUT    EDATA              ;STORE EXP. DATA
 (1)  4736  D3  95            10.0                        OUT     EDATA          ;WRITE AC INTO EDATA
 (1)  4738  7F                 4.0                        MOV     A,A            ;RETRY LINK
2225  4739                               ERRB    ALP04,ALP08,0
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4739  CD  12  28        18.0                        CALL    ERLPB          ;PROCESS ERROR - DO 2.3
 (1)       0025                                          MSGN    =       MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  473C  25                                           .BYTE   MSGN           ;MESSAGE NUMBER ID
 (1)  473D  00                                           .BYTE   0              ;PRINT ROUTINE NUMBER
 (1)  473E  CD  15  28        18.0                ALP08:: CALL    CKLOP          ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4741  DA  1D  47        10.0                        JC      ALP04          ;LOOP ADDRESS IF LOOP SPECIFIED
2226                          ;>DATA WRITTEN TO MTA #3 THRESHOLD NOT = DATA READ
2227  4744  7E                  7.0         MOV     A,M              ;GET DATA
2228  4745  3C                  4.0         INR     A                ;UPDATE THE DATA BYTE
2229  4746  77                  7.0         MOV     M,A              ;SAVE UPDATED DATA
2230  4747  FE  08              7.0         CPI     @10              ;DONE?
2231  4749  C2  1D  47        10.0         JNZ     ALP04            ;NO - CONTINUE UNTIL DONE
2232
```

```
2234  474C                                        ENDTST  TST10X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  474C                                       REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  474C  CD  06  28          18.0                             CALL    REQST
 (2)  474F  00                                                   .BYTE                   ;DATA PATTERN NUMBER
 (2)  4750  00  00                                               .WORD                   ;SYSTEM "" COUNT
 (2)  4752  00  00                                               .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4754  00                                                   .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4755  07                                                   .BYTE   7               ;REQUEST CODE
 (1)  4756  3A  9A  4F          13.0             LDA     ITERA           ;GET ITERATION COUNT
 (1)  4759  3D                   4.0             DCR     A               ;DOWNCOUNT
 (1)  475A  32  9A  4F          13.0             STA     ITERA           ;SAVE COUNT
 (1)  475D  F2  05  47          10.0             JP      TST10X          ;DO TEST UNTIL TILL = 0
```

```
                        .SBTTL  TEST 11 - COMMAND FUNCTION TEST - MTA #0
4760                    ST
                        ;*********************************************************************
                        ;*TEST TITLE
                        ;*---- -----
                        ;*COMMAND FUNCTION TEST MTA #0
4760                    SD
                        ;*********************************************************************
                        ;*DESCRIPTION
                        ;*-----------
                        ;*THIS TEST CHECKS THE CLEAR TU, SET PE, SET GCR, DSE AND TEST COMMANDS
                        ;*BY ISSUEING EACH COMMAND AND VERIFYING THAT THE PROPER STATUS INDICATORS
                        ;*SET AND/OR CLEAR.
4760                    SP
                        ;*********************************************************************
                        ;*PROCEDURE
                        ;*---------
                        ;*BGNTST
                        ;*  IF TU PORT #0 NOT SELECTED BY USER
                        ;*  :  THEN-NEXT TEST
                        ;*  :  ELSE-CONTINUE
                        ;*  ENDIF
                        ;*  CALL SUBROUTINE CLEAR
                        ;*  SELECT TU PORT #0
                        ;*  CALL SUBROUTINE CMDCOM
                        ;*ENDTST
4760                    SE
                        ;*********************************************************************
                        ;*ERRORS
                        ;*------
                        ;*MTA2 MICRO TEST 11
                        ;*MTA2 MICRO ERROR 47
                        ;*MTA COMMAND FUNCTION TEST - MTA #0
                        ;*M8954, M8955'S
                        ;*MTA STATUS REGISTER 1 NOT=0 AFTER CLEAR CMD.
                        ;*ACTUAL = NNNN
                        ;*
                        ;*MTA2 MICRO TEST 11
                        ;*MTA2 MICRO ERROR 50
                        ;*MTA COMMAND FUNCTION TEST - MTA #0
                        ;*M8954, M8955'S
                        ;*''READY ON'' IN MTA REG. 0 NOT=0 AFTER CLEAR CMD.
                        ;*
                        ;*MTA2 MICRO TEST 11
                        ;*MTA2 MICRO ERROR 51
                        ;*MTA COMMAND FUNCTION TEST - MTA #0
                        ;*M8954, M8955'S
                        ;*''CMD PE'' IN MTA REG. 2 NOT-0 AFTER CLEAR CMD.
                        ;*
                        ;*MTA2 MICRO TEST 11
                        ;*MTA2 MICRO ERROR 52
                        ;*MTA COMMAND FUNCTION TEST - MTA #0
                        ;*M8954, M8955'S
                        ;*''PES'' IN MTA REG. 0 NOT SET AFTER SET PE CMD.
```

```
2278                                    ;*
2279                                    ;*MTA2 MICRO TEST 11
2280                                    ;*MTA2 MICRO ERROR 53
2281                                    ;*MTA COMMAND FUNCTION TEST - MTA #0
2282                                    ;*M8954, M8955'S
2283                                    ;*''PEC'' IN MTA REGISTER 2 NOT SET AFTER SET PE CMD.
2284                                    ;*
2285                                    ;*MTA2 MICRO TEST 11
2286                                    ;*MTA2 MICRO ERROR 54
2287                                    ;*MTA COMMAND FUNCTION TEST - MTA #0
2288                                    ;*M8954, M8955'S
2289                                    ;*''PES'' IN MTA REGISTER 0 NOT RESET AFTER SET GCR CMD.
2290                                    ;*
2291                                    ;*MTA2 MICRO TEST 11
2292                                    ;*MTA2 MICRO ERROR 55
2293                                    ;*MTA COMMAND FUNCTION TEST - MTA #0
2294                                    ;*M8954, M8955'S
2295                                    ;*''PEC'' IN MTA REGISTER 2 NOT RESET AFTER SET GCR CMD.
2296                                    ;*
2297                                    ;*MTA2 MICRO TEST 11
2298                                    ;*MTA2 MICRO ERROR 56
2299                                    ;*MTA COMMAND FUNCTION TEST - MTA #0
2300                                    ;*M8954, M8955'S
2301                                    ;*''DSE'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
2302                                    ;*
2303                                    ;*MTA2 MICRO TEST 11
2304                                    ;*MTA2 MICRO ERROR 57
2305                                    ;*MTA COMMAND FUNCTION TEST - MTA #0
2306                                    ;*M8954, M8955'S
2307                                    ;*''FWD'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
2308                                    ;*
2309                                    ;*MTA2 MICRO TEST 11
2310                                    ;*MTA2 MICRO ERROR 60
2311                                    ;*MTA COMMAND FUNCTION TEST - MTA #0
2312                                    ;*M8954, M8955'S
2313                                    ;*''RDY ON'' IN MTA REGISTER 0 NOT SET AFTER TEST CMD.
2314                                    ;*
2315                                    ;*MTA2 MICRO TEST 11
2316                                    ;*MTA2 MICRO ERROR 61
2317                                    ;*MTA COMMAND FUNCTION TEST - MTA #0
2318                                    ;*M8954, M8955'S
2319                                    ;*''CMD PE'' IN MTA REGISTER 2 NOT SET AFTER TEST CMD
2320                                    ;*
2321                                    ;*MTA2 MICRO TEST 11
2322                                    ;*MTA2 MICRO ERROR 62
2323                                    ;*MTA COMMAND FUNCTION TEST - MTA #0
2324                                    ;*M8954, M8955'S
2325                                    ;*''TU78'' IN MTA REGISTER 2 NOT SET
2326  4760                              S
 (1)                                    ; ***************************************************************************
2327  4760                              TEST11: TESTX   @011            ;INITIALIZE THE TEST
 (1)  4760  3E  09           7.0                     MVI     A,@011          ;DEFINE THE TEST NUMBER
 (1)  4762  CD  03  28      18.0                     CALL    TSET            ;SETUP THE TEST
2328                                    ;%MTA COMMAND FUNCTION TEST - MTA #0
```

```
2329                                    ;&M8954, M8955'S
2330  4765  3A BA 49      13.0          LDA     UNITMP          ;GET THE UNIT MAP
2331  4768  E6 01          7.0          ANI     @001            ;TEST UNIT 0
2332  476A  CA 8D 47      10.0          JZ      TEST12          ;NO-GO CHECK TEST 12
2333  476D  CD 20 48      18.0  TST11X: CALL    CLEAR           ;CLEAN UP THE PORTS
2334  4770  DB E0         10.0          IN      INTSTA          ;GET MASSBUS SELECT
2335  4772  E6 80          7.0          ANI     BIT7
2336  4774  D3 E0         10.0          OUT     MBSEL           ;SELECT TU PORT #0
2337  4776  CD 4F 48      18.0          CALL    CMDCOM
2338  4779                              ENDTST  TST11X
 (1)                      ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4779                            REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4779  CD 06 28      18.0                        CALL    REQST
 (2)  477C  00                                        .BYTE                   ;DATA PATTERN NUMBER
 (2)  477D  00 00                                     .WORD                   ;SYSTEM '"' COUNT
 (2)  477F  00 00                                     .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4781  00                                        .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4782  07                                        .BYTE   7               ;REQUEST CODE
 (1)  4783  3A 9A 4F      13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4786  3D             4.0          DCR     A               ;DOWNCOUNT
 (1)  4787  32 9A 4F      13.0          STA     ITERA           ;SAVE COUNT
 (1)  478A  F2 6D 47      10.0          JP      TST11X          ;DO TEST UNTIL TILL - 0
```

```
                                .SBTTL  TEST 12 - COMMAND FUNCTION TEST - MTA #1
478D                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*COMMAND FUNCTION TEST MTA #1
478D                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE CLEAR TU, SET PE, SET GCR, DSE AND TEST COMMANDS
                                ;*BY ISSUEING EACH COMMAND AND VERIFYING THAT THE PROPER STATUS INDICATORS
                                ;*SET AND/OR CLEAR.
478D                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  IF TU PORT #1 NOT SELECTED BY USER
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  CALL SUBROUTINE CLEAR
                                ;*  SELECT TU PORT #1
                                ;*  CALL SUBROUTINE CMDCOM
                                ;*ENDTST
478D                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA2 MICRO TEST 12
                                ;*MTA2 MICRO ERROR 47
                                ;*MTA COMMAND FUNCTION TEST - MTA #1
                                ;*M8954, M8955'S
                                ;*MTA STATUS REGISTER 1 NOT=0 AFTER CLEAR CMD.
                                ;*ACTUAL = NNNN
                                ;*
                                ;*MTA2 MICRO TEST 12
                                ;*MTA2 MICRO ERROR 50
                                ;*MTA COMMAND FUNCTION TEST - MTA #1
                                ;*M8954, M8955'S
                                ;*''READY ON'' IN MTA REG. 0 NOT-0 AFTER CLEAR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 12
                                ;*MTA2 MICRO ERROR 51
                                ;*MTA COMMAND FUNCTION TEST - MTA #1
                                ;*M8954, M8955'S
                                ;*''CMD PE'' IN MTA REG. 2 NOT=0 AFTER CLEAR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 12
                                ;*MTA2 MICRO ERROR 52
                                ;*MTA COMMAND FUNCTION TEST - MTA #1
                                ;*M8954, M8955'S
                                ;*''PES'' IN MTA REG. 0 NOT SET AFTER SET PE CMD.
```

```
                                ;*
                                ;*MTA2 MICRO TEST 12
                                ;*MTA2 MICRO ERROR 53
                                ;*MTA COMMAND FUNCTION TEST - MTA #1
                                ;*M8954, M8955'S
                                ;*''PEC'' IN MTA REGISTER 2 NOT SET AFTER SET PE CMD.
                                ;*
                                ;*MTA2 MICRO TEST 12
                                ;*MTA2 MICRO ERROR 54
                                ;*MTA COMMAND FUNCTION TEST - MTA #1
                                ;*M8954, M8955'S
                                ;*''PES'' IN MTA REGISTER 0 NOT RESET AFTER SET GCR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 12
                                ;*MTA2 MICRO ERROR 55
                                ;*MTA COMMAND FUNCTION TEST - MTA #1
                                ;*M8954, M8955'S
                                ;*''PEC'' IN MTA REGISTER 2 NOT RESET AFTER SET GCR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 12
                                ;*MTA2 MICRO ERROR 56
                                ;*MTA COMMAND FUNCTION TEST - MTA #1
                                ;*M8954, M8955'S
                                ;*''DSE'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
                                ;*
                                ;*MTA2 MICRO TEST 12
                                ;*MTA2 MICRO ERROR 57
                                ;*MTA COMMAND FUNCTION TEST - MTA #1
                                ;*M8954, M8955'S
                                ;*''FWD'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
                                ;*
                                ;*MTA2 MICRO TEST 12
                                ;*MTA2 MICRO ERROR 60
                                ;*MTA COMMAND FUNCTION TEST - MTA #1
                                ;*M8954, M8955'S
                                ;*''RDY ON'' IN MTA REGISTER 0 NOT SET AFTER TEST CMD.
                                ;*
                                ;*MTA2 MICRO TEST 12
                                ;*MTA2 MICRO ERROR 61
                                ;*MTA COMMAND FUNCTION TEST - MTA #1
                                ;*M8954, M8955'S
                                ;*''CMD PE'' IN MTA REGISTER 2 NOT SET AFTER TEST CMD
                                ;*
                                ;*MTA2 MICRO TEST 12
                                ;*MTA2 MICRO ERROR 62
                                ;*MTA COMMAND FUNCTION TEST - MTA #1
                                ;*M8954, M8955'S
                                ;*''TU78'' IN MTA REGISTER 2 NOT SET
478D                            S
 (1)                            ; *****************************************************************************
478D                            TEST12: TESTX   @012            ;INITIALIZE THE TEST
 (1) 478D  3E  0A        7.0                    MVI     A,@012          ;DEFINE THE TEST NUMBER
 (1) 478F  CD  03  28    18.0                   CALL    TSET            ;SETUP THE TEST
                                ;%MTA COMMAND FUNCTION TEST - MTA #1
```

```
2433                                    ;&M8954, M8955'S
2434  4792  3A  BA  49      13.0        LDA     UNITMP          ;GET THE UNIT MAP
2435  4795  E6  02              7.0     ANI     @002            ;TEST UNIT 1
2436  4797  CA  BC  47      10.0        JZ      TEST13          ;NO-GO CHECK TEST 13
2437  479A  CD  20  48      18.0  TST12X: CALL    CLEAR           ;CLEAN UP THE PORTS
2438  479D  DB  E0            10.0      IN      INTSTA          ;GET MASSBUS SELECT
2439  479F  E6  80              7.0     ANI     BIT7
2440  47A1  F6  01              7.0     ORI     $01
2441  47A3  D3  E0            10.0      OUT     MBSEL           ;SELECT TU PORT #1
2442  47A5  CD  4F  48        18.0      CALL    CMDCOM
2443  47A8                              ENDTST  TST12X
 (1)                            ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  47A8                      REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  47A8  CD  06  28  18.0                            CALL    REQST
 (2)  47AB  00                                          .BYTE                   ;DATA PATTERN NUMBER
 (2)  47AC  00  00                                      .WORD                   ;SYSTEM '"' COUNT
 (2)  47AE  00  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  47B0  00                                          .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  47B1  07                                          .BYTE   7               ;REQUEST CODE
 (1)  47B2  3A  9A  4F  13.0            LDA     ITERA           ;GET ITERATION COUNT
 (1)  47B5  3D            4.0           DCR     A               ;DOWNCOUNT
 (1)  47B6  32  9A  4F  13.0            STA     ITERA           ;SAVE COUNT
 (1)  47B9  F2  9A  47  10.0            JP      TST12X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 13 - COMMAND FUNCTION TEST - MTA #2
47BC                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*COMMAND FUNCTION TEST MTA #2
47BC                            SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE CLEAR TU, SET PE, SET GCR, DSE AND TEST COMMANDS
                                ;*BY ISSUEING EACH COMMAND AND VERIFYING THAT THE PROPER STATUS INDICATORS
                                ;*SET AND/OR CLEAR.
47BC                            SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  IF TU PORT #2 NOT SELECTED BY USER
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  CALL SUBROUTINE CLEAR
                                ;*  SELECT TU PORT #2
                                ;*  CALL SUBROUTINE CMDCOM
                                ;*ENDTST
47BC                            SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA2 MICRO TEST 13
                                ;*MTA2 MICRO ERROR 47
                                ;*MTA COMMAND FUNCTION TEST - MTA #2
                                ;*M8954, M8955'S
                                ;*MTA STATUS REGISTER 1 NOT=0 AFTER CLEAR CMD.
                                ;*ACTUAL = NNNN
                                ;*
                                ;*MTA2 MICRO TEST 13
                                ;*MTA2 MICRO ERROR 50
                                ;*MTA COMMAND FUNCTION TEST - MTA #2
                                ;*M8954, M8955'S
                                ;*''READY ON'' IN MTA REG. 0 NOT=0 AFTER CLEAR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 13
                                ;*MTA2 MICRO ERROR 51
                                ;*MTA COMMAND FUNCTION TEST - MTA #2
                                ;*M8954, M8955'S
                                ;*''CMD PE'' IN MTA REG. 2 NOT=0 AFTER CLEAR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 13
                                ;*MTA2 MICRO ERROR 52
                                ;*MTA COMMAND FUNCTION TEST - MTA #2
                                ;*M8954, M8955'S
                                ;*''PES'' IN MTA REG. 0 NOT SET AFTER SET PE CMD.
```

```
2487                                    ;*
2488                                    ;*MTA2 MICRO TEST 13
2489                                    ;*MTA2 MICRO ERROR 53
2490                                    ;*MTA COMMAND FUNCTION TEST - MTA #2
2491                                    ;*M8954, M8955'S
2492                                    ;*''PEC'' IN MTA REGISTER 2 NOT SET AFTER SET PE CMD.
2493                                    ;*
2494                                    ;*MTA2 MICRO TEST 13
2495                                    ;*MTA2 MICRO ERROR 54
2496                                    ;*MTA COMMAND FUNCTION TEST - MTA #2
2497                                    ;*M8954, M8955'S
2498                                    ;*''PES'' IN MTA REGISTER 0 NOT RESET AFTER SET GCR CMD.
2499                                    ;*
2500                                    ;*MTA2 MICRO TEST 13
2501                                    ;*MTA2 MICRO ERROR 55
2502                                    ;*MTA COMMAND FUNCTION TEST - MTA #2
2503                                    ;*M8954, M8955'S
2504                                    ;*''PEC'' IN MTA REGISTER 2 NOT RESET AFTER SET GCR CMD.
2505                                    ;*
2506                                    ;*MTA2 MICRO TEST 13
2507                                    ;*MTA2 MICRO ERROR 56
2508                                    ;*MTA COMMAND FUNCTION TEST - MTA #2
2509                                    ;*M8954, M8955'S
2510                                    ;*''DSE'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
2511                                    ;*
2512                                    ;*MTA2 MICRO TEST 13
2513                                    ;*MTA2 MICRO ERROR 57
2514                                    ;*MTA COMMAND FUNCTION TEST - MTA #2
2515                                    ;*M8954, M8955'S
2516                                    ;*''FWD'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
2517                                    ;*
2518                                    ;*MTA2 MICRO TEST 13
2519                                    ;*MTA2 MICRO ERROR 60
2520                                    ;*MTA COMMAND FUNCTION TEST - MTA #2
2521                                    ;*M8954, M8955'S
2522                                    ;*''RDY ON'' IN MTA REGISTER 0 NOT SET AFTER TEST CMD.
2523                                    ;*
2524                                    ;*MTA2 MICRO TEST 13
2525                                    ;*MTA2 MICRO ERROR 61
2526                                    ;*MTA COMMAND FUNCTION TEST - MTA #2
2527                                    ;*M8954, M8955'S
2528                                    ;*''CMD PE'' IN MTA REGISTER 2 NOT SET AFTER TEST CMD
2529                                    ;*
2530                                    ;*MTA2 MICRO TEST 13
2531                                    ;*MTA2 MICRO ERROR 62
2532                                    ;*MTA COMMAND FUNCTION TEST - MTA #2
2533                                    ;*M8954, M8955'S
2534                                    ;*''TU78'' IN MTA REGISTER 2 NOT SET
2535  47BC                              S
 (1)                                    ; ***************************************************************************
2536  47BC                              TEST13: TESTX   @013            ;INITIALIZE THE TEST
 (1)  47BC  3E  0B           7.0                     MVI     A,@013          ;DEFINE THE TEST NUMBER
 (1)  47BE  CD  03  28      18.0                     CALL    TSET            ;SETUP THE TEST
2537                                    ;%MTA COMMAND FUNCTION TEST - MTA #2
```

```
2538                                    ;&M8954, M8955'S
2539  47C1  3A  BA  49      13.0         LDA     UNITMP          ;GET THE UNIT MAP
2540  47C4  E6  04           7.0         ANI     @004            ;TEST UNIT 2
2541  47C6  CA  EB  47      10.0         JZ      TEST14          ;NO-GO CHECK TEST 14
2542  47C9  CD  20  48      18.0  TST13X: CALL    CLEAR           ;CLEAN UP THE PORTS
2543  47CC  DB  E0          10.0         IN      INTSTA          ;GET MASSBUS SELECT
2544  47CE  E6  80           7.0         ANI     BIT7
2545  47D0  F6  02           7.0         ORI     $02
2546  47D2  D3  E0          10.0         OUT     MBSEL           ;SELECT TU PORT #2
2547  47D4  CD  4F  48      18.0         CALL    CMDCOM
2548  47D7                               ENDTST  TST13X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  47D7                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  47D7  CD  06  28      18.0                        CALL    REQST
 (2)  47DA  00                                          .BYTE                   ;DATA PATTERN NUMBER
 (2)  47DB  00  00                                      .WORD                   ;SYSTEM "" COUNT
 (2)  47DD  00  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  47DF  00                                          .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  47E0  07                                          .BYTE   7               ;REQUEST CODE
 (1)  47E1  3A  9A  4F      13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1)  47E4  3D               4.0         DCR     A               ;DOWNCOUNT
 (1)  47E5  32  9A  4F      13.0         STA     ITERA           ;SAVE COUNT
 (1)  47E8  F2  C9  47      10.0         JP      TST13X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 14 - COMMAND FUNCTION TEST - MTA #3
47EB                            ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*COMMAND FUNCTION TEST MTA #3
47EB                            SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE CLEAR TU, SET PE, SET GCR, DSE AND TEST COMMANDS
                                ;*BY ISSUEING EACH COMMAND AND VERIFYING THAT THE PROPER STATUS INDICATORS
                                ;*SET AND/OR CLEAR.
47EB                            SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  IF TU PORT #3 NOT SELECTED BY USER
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  CALL SUBROUTINE CLEAR
                                ;*  SELECT TU PORT #3
                                ;*  CALL SUBROUTINE CMDCOM
                                ;*ENDIST
47EB                            SE
 (1)                            ;*********************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA2 MICRO TEST 14
                                ;*MTA2 MICRO ERROR 47
                                ;*MTA COMMAND FUNCTION TEST - MTA #3
                                ;*M8954, M8955'S
                                ;*MTA STATUS REGISTER 1 NOT=0 AFTER CLEAR CMD.
                                ;*ACTUAL = NNNN
                                ;*
                                ;*MTA2 MICRO TEST 14
                                ;*MTA2 MICRO ERROR 50
                                ;*MTA COMMAND FUNCTION TEST - MTA #3
                                ;*M8954, M8955'S
                                ;*''READY ON'' IN MTA REG. 0 NOT=0 AFTER CLEAR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 14
                                ;*MTA2 MICRO ERROR 51
                                ;*MTA COMMAND FUNCTION TEST - MTA #3
                                ;*M8954, M8955'S
                                ;*''CMD PE'' IN MTA REG. 2 NOT=0 AFTER CLEAR CMD.
                                ;*
                                ;*MTA2 MICRO TEST 14
                                ;*MTA2 MICRO ERROR 52
                                ;*MTA COMMAND FUNCTION TEST - MTA #3
                                ;*M8954, M8955'S
                                ;*''PES'' IN MTA REG. 0 NOT SET AFTER SET PE CMD.
```

```
2592                                    ;*
2593                                    ;*MTA2 MICRO TEST 14
2594                                    ;*MTA2 MICRO ERROR 53
2595                                    ;*MTA COMMAND FUNCTION TEST - MTA #3
2596                                    ;*M8954, M8955'S
2597                                    ;*"PEC" IN MTA REGISTER 2 NOT SET AFTER SET PE CMD.
2598                                    ;*
2599                                    ;*MTA2 MICRO TEST 14
2600                                    ;*MTA2 MICRO ERROR 54
2601                                    ;*MTA COMMAND FUNCTION TEST - MTA #3
2602                                    ;*M8954, M8955'S
2603                                    ;*"PES" IN MTA REGISTER 0 NOT RESET AFTER SET GCR CMD.
2604                                    ;*
2605                                    ;*MTA2 MICRO TEST 14
2606                                    ;*MTA2 MICRO ERROR 55
2607                                    ;*MTA COMMAND FUNCTION TEST - MTA #3
2608                                    ;*M8954, M8955'S
2609                                    ;*"PEC" IN MTA REGISTER 2 NOT RESET AFTER SET GCR CMD.
2610                                    ;*
2611                                    ;*MTA2 MICRO TEST 14
2612                                    ;*MTA2 MICRO ERROR 56
2613                                    ;*MTA COMMAND FUNCTION TEST - MTA #3
2614                                    ;*M8954, M8955'S
2615                                    ;*"DSE" IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
2616                                    ;*
2617                                    ;*MTA2 MICRO TEST 14
2618                                    ;*MTA2 MICRO ERROR 57
2619                                    ;*MTA COMMAND FUNCTION TEST - MTA #3
2620                                    ;*M8954, M8955'S
2621                                    ;*"FWD" IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
2622                                    ;*
2623                                    ;*MTA2 MICRO TEST 14
2624                                    ;*MTA2 MICRO ERROR 60
2625                                    ;*MTA COMMAND FUNCTION TEST - MTA #3
2626                                    ;*M8954, M8955'S
2627                                    ;*"RDY ON" IN MTA REGISTER 0 NOT SET AFTER TEST CMD.
2628                                    ;*
2629                                    ;*MTA2 MICRO TEST 14
2630                                    ;*MTA2 MICRO ERROR 61
2631                                    ;*MTA COMMAND FUNCTION TEST - MTA #3
2632                                    ;*M8954, M8955'S
2633                                    ;*"CMD PE" IN MTA REGISTER 2 NOT SET AFTER TEST CMD
2634                                    ;*
2635                                    ;*MTA2 MICRO TEST 14
2636                                    ;*MTA2 MICRO ERROR 62
2637                                    ;*MTA COMMAND FUNCTION TEST - MTA #3
2638                                    ;*M8954, M8955'S
2639                                    ;*"TU78" IN MTA REGISTER 2 NOT SET
2640  47EB                              $
 (1)                                    ; *****************************************************************************
2641  47EB                              TEST14: TESTX   @014            ;INITIALIZE THE TEST
 (1)  47EB  3E  0C            7.0                  MVI     A,@014          ;DEFINE THE TEST NUMBER
 (1)  47ED  CD  03  28       18.0                  CALL    TSET            ;SETUP THE TEST
2642                                    ;%MTA COMMAND FUNCTION TEST - MTA #3
```

```
2643                                    ;&M8954, M8955'S
2644  47F0  3A  BA  49      13.0       LDA     UNITMP          ;GET THE UNIT MAP
2645  47F3  E6  08           7.0       ANI     @0'0            ;TEST UNIT 3
2646  47F5  CA  1A  48      10.0       JZ      TEST15          ;NO-GO CHECK TEST 15
2647  47F8  CD  20  48      18.0  TST14X: CALL    CLEAR           ;CLEAN UP THE PORTS
2648  47FB  DB  E0          10.0       IN      INTSTA          ;GET MASSBUS SELECT
2649  47FD  E6  80           7.0       ANI     BIT7
2650  47FF  F6  03           7.0       ORI     $03
2651  4801  D3  E0          10.0       OUT     MBSEL           ;SELECT TU PORT #3
2652  4803  CD  4F  48      18.0       CALL    CMDCOM
2653  4806                             ENDTST  TST14X
 (1)                             ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4806                       REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4806  CD  06  28      18.0                         CALL    REQST
 (2)  4809  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  480A  00  00                                       .WORD                   ;SYSTEM '"' COUNT
 (2)  480C  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  480E  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  480F  07                                           .BYTE   7               ;REQUEST CODE
 (1)  4810  3A  9A  4F      13.0       LDA     ITERA           ;GET ITERATION COUNT
 (1)  4813  3D           4.0       DCR     A               ;DOWNCOUNT
 (1)  4814  32  9A  4F      13.0       STA     ITERA           ;SAVE COUNT
 (1)  4817  F2  F8  47      10.0       JP      TST14X          ;DO TEST UNTIL TILL = 0
2654
2655  481A  C3  1D  48      10.0  TEST15: JMP     EXIT
2656  481D  C3  18  28      10.0  EXIT:   JMP     TSTEND
```

```
                                .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
4820                            SSUB
 (1)                            ; ************************************************************************
 (1)                            ;*SUBROUTINE TITLE
 (1)                            ;*---------- -----
                                ;*CLEAR ALL TU PORTS
4820                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                ;*AND LOOP MODES.
4820                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNSUB
                                ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                ;*     CLEAR PORT SELECT FOR TRANSPORT
                                ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                ;*     CLEAR PORT AMTIE WORD
                                ;*ENDSUB
4820                            S
 (1)                            ; ************************************************************************
4820  F5              12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
4821  C5              12.0          PUSH    B
4822  06  00           7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
4824  DB  E0         10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
4826  E6  80          7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
4828  B0              4.0          ORA     B               ;ADD IN THE SELECTED PORT #
4829  D3  E0         10.0          OUT     MBSEL           ;RESET TO THIS PORT
482B  3E  80          7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
482D  D3  40         10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
482F  AF              4.0          XRA     A               ;CLEAR TU COMMAND A
4830  D3  40         10.0          OUT     TCMD            ;
4832  3E  81          7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
4834  D3  40         10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
4836  3E  00          7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
4838  D3  40         10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
483A  3E  03          7.0          MVI     A,EOTCLR        ;LOAD TU "EOT STATUS CLEAR" COMMAND
483C  D3  40         10.0          OUT     TCMD            ;ISSUE TU "EOT STATUS CLEAR" COMMAND
483F  AF              4.0          XRA     A
483F  D3  44         10.0          OUT     TAMT            ;CLEAR AMTIE WORD
4841  D3  48         10.0          OUT     PDIAG           ;CLEAR DJAG CONTROL WORD
4843  D3  4C         10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
4845  04              4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
4846  78              4.0          MOV     A,B
```

```
2699  4847  FE  04          7.0       CPI   4          ;DONE?
2700  4849  C2  24  48      10.0      JNZ   CLRLP      ;NO - CLEAR THIS PORT ALSO
2701  484C  C1              10.0      POP   B          ;RESET B & C
2702  484D  F1              10.0      POP   PSW        ;ALL DONE
2703  484E  C9              10.0      RET              ;EXIT
```

```
                                .SBTTL  SUBROUTINE COMMAND COMMON
484F                            $SUB
 (1)                            ;*****************************************************************************
 (1)                            ;*SUBROUTINE TITLE
 (1)                            ;*---------- -----
                                ;*COMMAND COMMON CODE
484F                            $D
 (1)                            ;*****************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS SUBROUTINE IS USED TO TEST THE FUNCTIONALITY OF AN MTA BOARD
                                ;*COMMAND REPERTOIRE BY ISSUING THE CLEAR TU, SET PE, SET GCR, DSE, AND
                                ;*TEST COMMANDS AND MONITORING THE RESPONSE.
484F                            $P
 (1)                            ;*****************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNSUB
                                ;*  IF MANUAL INTERVENTION NOT ALLOWED
                                ;*  :  THEN - EXIT THE SUBROUTINE
                                ;*  :  ELSE - CONTINUE
                                ;*  ENDIF
                                ;*  ASK THE USER TO PUT THE TU78 OFF-LINE
                                ;*  SET THE FWD, REV, WRT, WRT INH AND LWR COMMAND BITS
                                ;*  ISSUE MTA CLEAR TU COMMAND
                                ;*  IF MTA STATUS BITS FWD, REV, WRT, WRT INH, LWR, DSE, CMD PE, AND RDY ON=0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  ISSUE MTA SET PE COMMAND
                                ;*  IF MTA STATUS BITS PES AND PEC=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  ISSUE MTA SET GCR COMMAND
                                ;*  IF MTA STATUS BITS PES AND PEC=0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  ISSUE MTA DSE COMMAND
                                ;*  IF MTA STATUS BITS DSE AND FWD=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  ISSUE MTA TEST COMMAND
                                ;*  IF RDY ON AND CMD PE-1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  IF MTA STATUS BIT TU78=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  CALL SUBROUTINE CLEAR
```

```
  2750                                  ;*ENDSUB
  2751  484F                            S
   (1)                                  ; ************************************************************************
```

```
2753  484F                                  CMDCOM: QMSGM   OFFOUT          ;ASK THE USER TO PUT THE DRIVE OFF-LINE
 (1)  484F  CD  1E  28        18.0                CALL    QUEM
 (1)        0026                                MSGN=MSGN+1
 (1)  4852  26                                  .BYTE   MSGN
 (1)  4853  DA  B5  49        10.0                JC      OFFOUT
2754                                    ;>PUT THE TU78 OFF LINE
2755  4856  3E  80                7.0   CMDCM:  MVI     A,$80            ;LOAD MTA REGISTER 0 SELECT CODE
2756  4858  D3  40               10.0           OUT     TCMD
2757  485A  DB  40               10.0           IN      TSTS             ;GET TU STATUS
2758  485C  E6  20                7.0           ANI     T.ONL            ;IS ONLINE BIT SET
2759  485E  CA  69  48          10.0           JZ      CMDCNO           ;NO , GOOD CONTINUE
2760  4861                                     ROUT    ADATA            ;YES , ERROR
 (1)  4861  D3  94               10.0                   OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  4863  7F                    4.0                   MOV     A,A              ;RETRY LINK
2761  4864                                     ERRA    CMDCOM,CMDCNO
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4864  CD  0F  28        18.0                           CALL    ERLPA            ;PROCESS ERROR - DO 2.3
 (1)        0027                                             MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4867  27                                               .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  4868  00                                               .BYTE                    ;PRINT ROUTINE NUMBER
 (1)  4869  CD  15  28        18.0                   CMDCNO::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  486C  DA  4F  48        10.0                           JC      CMDCOM           ;LOOP ADDRESS IF LOOP SPECIFIED
2762                                    ;>"ONL" BIT SET IN MTA 0
2763  486F                                     QMSGM   OFFOUT
 (1)  486F  CD  1E  28        18.0                CALL    QUEM
 (1)        0028                                MSGN=MSGN+1
 (1)  4872  28                                  .BYTE   MSGN
 (1)  4873  DA  B5  49        10.0                JC      OFFOUT
2764                                    ;>PUT THE TU78 ONLINE
2765  4876  DB  40               10.0           IN      TSTS             ;GET TU STATUS
2766  4878  E6  20                7.0           ANI     T.ONL            ;TU78 ONLINE ?
2767  487A  C2  85  48          10.0           JNZ     CMDCNZ           ;YES , CONTINUE
2768  487D                                     ROUT    ADATA            ;NO , ERROR TAPE SHOULD BE ONLINE
 (1)  487D  D3  94               10.0                   OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  487F  7F                    4.0                   MOV     A,A              ;RETRY LINK
2769  4880                                     ERRA    CMDCOM,CMDCNZ
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4880  CD  0F  28        18.0                           CALL    ERLPA            ;PROCESS ERROR - DO 2.3
 (1)        0029                                             MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4883  29                                               .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  4884  00                                               .BYTE                    ;PRINT ROUTINE NUMBER
 (1)  4885  CD  15  28        18.0                   CMDCNZ::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4888  DA  4F  48        10.0                           JC      CMDCOM           ;LOOP ADDRESS IF LOOP SPECIFIED
2770                                    ;> "ONL" BIT NOT SET IN MTA 0
2771  488B  3E  7C                7.0           MVI     A,$7C            ;SET THE MTA REGISTER 0 COMMAND BITS
2772  488D  D3  40               10.0           OUT     TCMD
2773
2774  488F  3E  81                7.0           MVI     A,$81            ;LOAD MTA REGISTER 1 SELECT CODE
2775  4891  D3  40               10.0           OUT     TCMD
2776  4893  AF                    4.0           XRA     A                ;ISSUE CLEAR TU COMMAND
2777  4894  D3  40               10.0           OUT     TCMD
2778
2779  4896  DB  40               10.0           IN      TSTS             ;READ MTA REGISTER 1
2780  4898  E6  FD                7.0           ANI     $FD              ;REMOVE JUNK
```

```
2781  489A  CA  A5  48  10.0          JZ      CMDCN1          ;=0? - CONTINUE
2782  489D                           ROUT    ADATA           ;SAVE THE ACTUAL DATA
 (1)  489D  D3  94      10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  489F  7F          4.0                    MOV     A,A             ;RETRY LINK
2783  48A0                           ERRA    CMDCM,CMDCN1
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  48A0  CD  0F  28  18.0                           CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)      002A                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48A3  2A                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48A4  00                                        .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  48A5  CD  15  28  18.0                   CMDCN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48A8  DA  56  48  10.0                           JC      CMDCM           ;LOOP ADDRESS IF LOOP SPECIFIED
2784                           ;>MTA STATUS REGISTER 1 NOT=0 AFTER CLEAR CMD.
2785
2786  48AB  3E  80           7.0     MVI     A,$80           ;LOAD MTA REGISTER 0 SELECT CODE
2787  48AD  D3  40           10.0    OUT     TCMD
2788  48AF  DB  40           10.0    IN      TSTS            ;READ MTA REGISTER 0
2789  48B1  E6  40           7.0     ANI     $40
2790  48B3  CA  BB  48       10.0    JZ      CMDCN2
2791  48B6                   ERR     CMDCM,CMDCN2
 (1)                           ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  48B6  CD  09  28  18.0                   CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)      002B                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48B9  2B                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48BA  00                                .BYTE
 (1)  48BB  CD  15  28  18.0           CMDCN2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  48BE  DA  56  48  10.0                   JC      CMDCM           ;LOOP ADDRESS IF LOOP SPECIFIED
2792                           ;>"READY ON" IN MTA REG. 0 NOT=0 AFTER CLEAR CMD.
2793  48C1  3E  82           7.0     MVI     A,$82           ;LOAD MTA REGISTER 2 SELECT CODE
2794  48C3  D3  40           10.0    OUT     TCMD
2795  48C5  DB  40           10.0    IN      TSTS
2796  48C7  E6  80           7.0     ANI     $80
2797  48C9  CA  D1  48       10.0    JZ      CMDCN3
2798  48CC                   ERR     CMDCM,CMDCN3
 (1)                           ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  48CC  CD  09  28  18.0                   CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)      002C                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48CF  2C                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48D0  00                                .BYTE
 (1)  48D1  CD  15  28  18.0           CMDCN3::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  48D4  DA  56  48  10.0                   JC      CMDCM           ;LOOP ADDRESS IF LOOP SPECIFIED
2799                           ;>"CMD PE" IN MTA REG. 2 NOT=0 AFTER CLEAR CMD.
```

```
2801  48D7  3E  81        7.0  CMDCOA: MVI   A,$81          ;LOAD MTA REGISTER 1 SELECT CODE
2802  48D9  D3  40       10.0          OUT   TCMD
2803  48DB  3E  01        7.0          MVI   A,$01          ;ISSUE SET PE COMMAND
2804  48DD  D3  40       10.0          OUT   TCMD
2805
2806  48DF  3E  82        7.0          MVI   A,$82          ;LOAD MTA REGISTER 2 SELECT CODE
2807  48E1  D3  40       10.0          OUT   TCMD
2808  48E3  DB  40       10.0          IN    TSTS
2809  48E5  E6  40        7.0          ANI   $40
2810  48E7  C2  EF  48   10.0          JNZ   CMDCN4
2811  48EA                             ERR   CMDCOA,CMDCN4
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  48EA  CD  09  28   18.0                CALL  ERLP          ;PROCESS ERROR - DO 2.3
 (1)      002D                              MSGN  =     MSGN+1        ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48ED  2D                              .BYTE MSGN          ;MESSAGE NUMBER ID
 (1)  48EE  00                              .BYTE
 (1)  48EF  CD  15  28   18.0       CMDCN4::       CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  48F2  DA  D7  48   10.0                JC    CMDCOA        ;LOOP ADDRESS IF LOOP SPECIFIED
2812                          ;>''PEC'' IN MTA REGISTER 2 NOT SET AFTER SET PE CMD.
2813  48F5  3E  82        7.0          MVI   A,$82
2814  48F7  D3  40       10.0          OUT   TCMD
2815  48F9  DB  40       10.0          IN    TSTS
2816  48FB  E6  40        7.0          ANI   $40
2817  48FD  C2  05  49   10.0          JNZ   CMDCN5
2818  4900                             ERR   CMDCOA,CMDCN5
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4900  CD  09  28   18.0                CALL  ERLP          ;PROCESS ERROR - DO 2.3
 (1)      002E                              MSGN  =     MSGN+1        ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4903  2E                              .BYTE MSGN          ;MESSAGE NUMBER ID
 (1)  4904  00                              .BYTE
 (1)  4905  CD  15  28   18.0       CMDCN5::       CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4908  DA  D7  48   10.0                JC    CMDCOA        ;LOOP ADDRESS IF LOOP SPECIFIED
2819                          ;>''PEC'' IN MTA REGISTER 2 NOT SET AFTER SET PE CMD.
2820  490B  3E  81        7.0  CMDCOB: MVI   A,$81          ;LOAD MTA REGISTER 1 SELECT CODE
2821  490D  D3  40       10.0          OUT   TCMD
2822  490F  3E  02        7.0          MVI   A,$02          ;ISSUE SET GCR COMMAND
2823  4911  D3  40       10.0          OUT   TCMD
2824
2825  4913  3E  80        7.0          MVI   A,$80          ;LOAD MTA REGISTER 0 SELECT CODE
2826  4915  D3  40       10.0          OUT   TCMD
2827  4917  DB  40       10.0          IN    TSTS
2828  4919  E6  08        7.0          ANI   $08
2829  491B  CA  23  49   10.0          JZ    CMDCN6
```

```
2831  491E                                        ERR     CMDCOB,CMDCN6
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  491E  CD  09  28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        002F                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4921  2F                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4922  00                                           .BYTE
 (1)  4923  CD  15  28        18.0               CMDCN6::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4926  DA  0B  49        10.0                       JC      CMDCOB          ;LOOP ADDRESS IF LOOP SPECIFIED
2832                                    ;>''PES'' IN MTA REGISTER 0 NOT RESET AFTER SET GCR CMD.
2833  4929  3E  82                7.0         MVI     A,$82
2834  492B  D3  40               10.0         OUT     TCMD
2835  492D  DB  40               10.0         IN      TSTS
2836  492F  E6  40                7.0         ANI     $40
2837  4931  CA  39  49           10.0         JZ      CMDCN7
2838  4934                                    ERR     CMDCOB,CMDCN7
 (1)                                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4934  CD  09  28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0030                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4937  30                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4938  00                                           .BYTE
 (1)  4939  CD  15  28        18.0               CMDCN7::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  493C  DA  0B  49        10.0                       JC      CMDCOB          ;LOOP ADDRESS IF LOOP SPECIFIED
2839                                    ;>''PEC'' IN MTA REGISTER 2 NOT RESET AFTER SET GCR CMD.
2840  493F  3E  81                7.0 CMDCOC: MVI     A,$81           ;LOAD MTA REG. 1 SELECT CODE
2841  4941  D3  40               10.0         OUT     TCMD
2842  4943  3E  06                7.0         MVI     A,$06           ;ISSUE THE DSE COMMAND
2843  4945  D3  40               10.0         OUT     TCMD
2844  4947  DB  40               10.0         IN      TSTS
2845  4949  E6  01                7.0         ANI     $01
2846  494B  C2  53  49           10.0         JNZ     CMDCN8
2847  494E                                    ERR     CMDCOC,CMDCN8
 (1)                                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  494E  CD  09  28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0031                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4951  31                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4952  00                                           .BYTE
 (1)  4953  CD  15  28        18.0               CMDCN8::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4956  DA  3F  49        10.0                       JC      CMDCOC          ;LOOP ADDRESS IF LOOP SPECIFIED
2848                                    ;>''DSE'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
2849
2850  4959  DB  40               10.0         IN      TSTS
2851  495B  E6  40                7.0         ANI     $40
2852  495D  C2  65  49           10.0         JNZ     CMDCN9
2853  4960                                    ERR     CMDCOC,CMDCN9
 (1)                                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4960  CD  09  28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0032                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4963  32                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4964  00                                           .BYTE
 (1)  4965  CD  15  28        18.0               CMDCN9::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4968  DA  3F  49        10.0                       JC      CMDCOC          ;LOOP ADDRESS IF LOOP SPECIFIED
2854                                    ;>''FWD'' IN MTA REGISTER 1 NOT SET AFTER DSE CMD.
2855
2856  496B  3E  81                7.0 CMDCOD: MVI     A,$81           ;LOAD MTA REGISTER 1 SELECT CODE
```

```
2857  496D  D3  40          10.0          OUT     TCMD
2858  496F  3E  07           7.0          MVI     A,$07           ;ISSUE THE TEST COMMAND
2859  4971  D3  40          10.0          OUT     TCMD
2860  4973  3E  80           7.0          MVI     A,$80           ;LOAD MTA REGISTER 0 SELECT CODE
2861  4975  D3  40          10.0          OUT     TCMD
2862  4977  DB  40          10.0          IN      TSTS
2863  4979  E6  40           7.0          ANI     $40             ;READY ON=1
2864  497B  C2  83  49       10.0          JNZ     CMDCNB
2865  497E                                ERR     CMDCOD,CMDCNB
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  497E  CD  09  28       18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0033                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4981  33                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4982  00                             .BYTE
 (1)  4983  CD  15  28       18.0          CMDCNB::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4986  DA  6B  49       10.0                  JC      CMDCOD          ;LOOP ADDRESS IF LOOP SPECIFIED
2866                        ;>''RDY ON'' IN MTA REGISTER 0 NOT SET AFTER TEST CMD.
2867  4989  3E  82           7.0          MVI     A,$82           ;LOAD MTA REGISTER 2 SELECT CODE
2868  498B  D3  40          10.0          OUT     TCMD
2869  498D  DB  40          10.0          IN      TSTS
2870  498F  E6  80           7.0          ANI     $80
2871  4991  C2  99  49       10.0          JNZ     CMDCNA
2872  4994                                ERR     CMDCOD,CMDCNA
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4994  CD  09  28       18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0034                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4997  34                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4998  00                             .BYTE
 (1)  4999  CD  15  28       18.0          CMDCNA::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  499C  DA  6B  49       10.0                  JC      CMDCOD          ;LOOP ADDRESS IF LOOP SPECIFIED
2873                        ;>''CMD PE'' IN MTA REGISTER 2 NOT SET AFTER TEST CMD
2874
2875  499F  3E  82           7.0  CMDCOE: MVI     A,$82           ;LOAD MTA REGISTER 2 SELECT CODE
2876  49A1  D3  40          10.0          OUT     TCMD
2877  49A3  DB  40          10.0          IN      TSTS            ;TU78 BIT=1?
2878  49A5  E6  10           7.0          ANI     $10
2879  49A7  C2  AF  49       10.0          JNZ     CMDCNG          ;YES-CONTINUE
2880  49AA                                ERR     CMDCOE,CMDCNG
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  49AA  CD  09  28       18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0035                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  49AD  35                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  49AE  00                             .BYTE
 (1)  49AF  CD  15  28       18.0          CMDCNG::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  49B2  DA  9F  49       10.0                  JC      CMDCOE          ;LOOP ADDRESS IF LOOP SPECIFIED
2881                        ;>''TU78'' IN MTA REGISTER 2 NOT SET
2882
2883  49B5  CD  20  48       18.0  OFFOUT: CALL    CLEAR
2884  49B8  C9                10.0          RET                     ;RETURN TO IN-LINE
2885
```

```
                                .SBTTL  PROGRAM VARIABLES

49B9  00                        DATA:   .BYTE   0               ;DATA FOR THE THRESHOLD TEST
49BA  00                        UNITMP: .BYTE   0               ;UNIT MAP
      0000                              .END
```

```
A      =%0007          ADATA  = 0094          ALP01  45FD          ALP02  465D
ALP03  46BD            ALP04  471D          ALP05  461E  G       ALP06  467E  G
ALP07  46DE  G         ALP08  473E  G       AMTIEP= 0001         AMTIE7= 0002
ARAIDF= 0098           ASAVE  4F98          ATTCD  4F97          AXNUM  4F91
B      =%0000          BADST  = 0090        BIT0   = 0001        BIT1   = 0002
BIT15  = 8000          BIT2   = 0004        BIT3   = 0008        BIT4   = 0010
BIT5   = 0020          BIT6   = 0040        BIT7   = 0080        BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A         BRKRAM= 4F10         BRKSTR= 4E60
BRKXCT= 4F00           BSAVE  4F9C          BYTCNT= 00D4         BYTEH  4F24
BYTEL  4F23            C      =%0001        CASCT  4F21          CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089        CATTL  = 0088        CBUSST= 00A1
CBYTH  = 008B          CBYTL  = 008A        CDG1H  = 0087        CDG1L  = 0086
CDG2H  = 0093          CDG2L  = 0092        CDG3H  = 0095        CDG3L  = 0094
CDVTH  = 008D          CDVTL  = 008C        CHPTIE= 0028         CHOTIE= 0020
CH1TIE= 0021           CH2TIE= 0022         CH3TIE= 0023         CH4TIE= 0024
CH5TIE= 0025           CH6TIE= 0026         CH7TIE= 0027         CKLOP  = 2815
CLEAR  4820            CLKCTL= 00F0         CLOCK  4F26          CLRLP  4824
CMCOH  = 0099          CMCOL  = 0098        CMC1H  = 009B        CMC1L  = 009A
CMC2H  = 009D          CMC2L  = 009C        CMC3H  = 009F        CMC3L  = 009E
CMDCM  4856            CMDCNA 4999  G       CMDCNB 4983  G       CMDCNG 49AF  G
CMDCNZ 4885  G         CMDCN0 4869  G       CMDCN1 48A5  G       CMDCN2 48BB  G
CMDCN3 48D1  G         CMDCN4 48EF  G       CMDCN5 4905  G       CMDCN6 4923  G
CMDCN7 4939  G         CMDCN8 4953  G       CMDCN9 4965  G       CMDCOA 48D7
CMDCOB 490B            CMDCOC 493F          CMDCOD 496B          CMDCOE 499F
CMDCOM 484F            CMINH  = 0097        CMINL  = 0096        CNTCTL= 00D7
CRCWRD= 0018           CSAVE  4F9D          CSRLH  = 0091        CSRLL  = 0090
CTCH   = 0085          CTCL   = 0084        CTSTH  = 008F        CTSTL  = 008E
CXCTH  = 0081          CXCTL  = 0080        CXINH  = 0083        CXINL  = 0082
C.     = 0001          C.AVAI= 008C         C.DP   = 0008        C.DSE  = 0010
C.DTU  = 0003          C.DVA  = 0008        C.FAIL= 00FC         C.FMT  = 0070
C.FNCT= 003E           C.GO   = 0001        C.INTC= 00FE         C.MAIN= 0020
C.NSA  = 0080          C.RCT  = 00FC        C.SER  = 0080        C.SHR  = 0040
C.SKPC= 000F           C.TAPE= 0040         C.WCS  = 0002        D      =%0002
DATA   49B9            DATACT= 00D0         DBUS   4F28          DBUSCT= 00C0
DBUSST= 00C0           DDRA   = 00D8        DDRAIN= 0010         DDRB   = 00D9
DDRBIN= 0002           DDRC   = 00DA        DDRCIN= 0001         DDRCO  = 0088
DDRCTL= 00DB           DIAFLG 4F22          DIAGPG= 4300         DIAGRM= 4F90
DIARD  = 000B          DONE1  = 0045        DONINT= 0010         DSAVE  4F9E
DSE    = 0006          DUMMY  431E  G       D.ATH0= 0001         D.ATH1= 0002
D.EOTD= 0010           D.LAGC= 0020         D.NOTW= 0040         D.NTHR= 0004
D.TACH= 0008           D.WR4  = 0080        E      =%0003        ECCBAD= 0042
ECCCOR= 0019           ECCOK  = 0041        ECCSTA= 001A         ECCTST= 00CE
EDATA  = 0095          EOTCLR= 0003         ERFLG  4F93          ERLP   = 2809
ERLPA  = 280F          ERLPB  = 2812        ERLPE  = 280C        ERNUM  4F90
ERRCNT= 00D6           ESAVE  4F9F          EXIT   481D          E.ACRC= 0010
E.AMT  = 0020          E.CDP  = 0080        E.CRC  = 0080        E.PNTR= 0008
E.RPE  = 0040          E.STEC= 0001         E.TTEC= 0002         E.UNC  = 0004
FIFORD= 006A           FORMAT 4F25          FWDTST= 0061         GCRID  = 0089
GCRSET= 0002           GOODTM= 0092         H      =%0004        HLSAVE 4FA0
IE     = 0008          INTSTA= 00E0         ITERA  4F9A          I.PWR  = 0020
I.RMPE= 0040           I5.5   = 0010        I6.5   = 0020        I7.5   = 0040
KCALL  = 005F          KCLR   = 007B        KDEP   = 003F        KENAB  = 0078
KEXAM  = 003E          KEYBRD= 00C8         KEY1   = 0078        KEY10  = 006D
KEY11  = 006E          KEY12  = 006F        KEY13  = 005C        KEY14  = 005D
```

```
KEY15 = 005E            KEY16 = 005F            KEY17 = 003C            KEY18 = 003D
KEY19 = 003E            KEY2  = 0079            KEY20 = 003F            KEY3  = 007A
KEY4  = 007B            KEY5  = 0074            KEY6  = 0075            KEY7  = 0076
KEY8  = 0077            KEY9  = 006C            KINTA = 006F            KLDAD = 003D
KN0   = 003C            KN1   = 005C            KN2   = 005D            KN3   = 005E
KN4   = 006C            KN5   = 006D            KN6   = 006E            KN7   = 0074
KN8   = 0C75            KN9   = 0076            KU2   = 0079            KU3   = 007A
KU8   = 0077            L     =%0005            LBLANK= 000F            LCE   = 000B
LCH   = 000C            LCL   = 000D            LCP   = 000E            LC0   = 0000
LC1   = 0001            LC2   = 0002            LC3   = 0003            LC4   = 0004
LC5   = 0005            LC6   = 0006            LC7   = 0007            LC8   = 0008
LC9   = 0009            LDLEDA= 00CA            LDLEDB= 00CB            LDLEDC= 00CC
LDLEDD= 00CD            LDLEDE= 00CE            LDLEDF= 00CF            LKDIAG= 2800
LKKBD = 004C            LKKEY = 0049            LKLWMG= 0058            LKLWMP= 0055
LKLWPG= 0052            LKLWPP= 004F            LKMOD7= 0046            LKOPR = 0046
LPFLG   4F94            LPNUM   4F92            M     =%0006            MBSEL = 00E0
MB.A  = 0008            MB.B  = 0004            MEMTOP= 4FFF            MINUS = 000A
MM    = 8000            MSE   = 0008            MSGN  = 0035            MTACLR= 0000
MT.ARA= 0020            MT.CPE= 0080            MT.DSE= 0001            MT.FWD= 0040
MT.INH= 0008            MT.LWR= 0004            MT.MOT= 0002            MT.NWT= 0080
MT.PEC= 0040            MT.PSB= 0004            MT.PS0= 0001            MT.PS1= 0002
MT.REV= 0020            MT.WRT= 0010            MT.Z  = 0008            M.ATA = 0080
M.CAPE= 0020            M.CONT= 0080            M.DEM = 0020            M.EBL = 0010
M.EXC = 0008            M.FAIL= 0008            M.ILR = 0010            M.INIT= 0010
M.OCC = 0020            M.ONLI= 0001            M.PE  = 0040            M.PORT= 0080
M.RDEN= 0002            M.RDPE= 0008            M.RUN = 0004            M.SCLK= 0001
M.TRA = 0040            M.UNIT= 0007            M.WCLK= 0040            M.WCLN= 0080
M.WREN= 0080            M5.5  = 0001            M6.5  = 0002            M7.5  = 0004
NOTCAP= 0088            OFFOUT  49B5            OKAY  = 00FF            OPRRAM= 4300
OPSTRT= 0058            OPVER = 0040            PADCNT= 00D5            PADCRC= 0080
PDIAG = 0048            PEID  = 008A            PENAB = 004C            PESET = 0001
PL    = 00B1            PRDD  = 004C            PRENF = 009C            PS    = 00B2
PSTAT = 0048            PSW   =%0009            P.AMTP= 0001            P.BCTC= 0040
P.CMDP= 0020            P.INTE= 0080            P.LCS = 0040            P.LWR = 0020
P.RDP = 0002            P.RPEN= 0004            P.RPST= 0002            P.RPOE= 0020
P.RP1E= 0010            P.RP2E= 0020            P.RP3E= 0010            P.SING= 0080
P.STAT= 0002            P.STPE= 0080            P.TACH= 0008            P.TUPR= 0010
P.WCSP= 0004            P.WDS = 0040            P.WFLP= 0001            P.WPEN= 0010
P.WPOE= 0008            P.WP1E= 0004            P.WP2E= 0008            P.WP3E= 0004
P.5VOK= 0002            QUE   = 281B            QUEM  = 281E            RAMT  = 0010
RARA  = 0006            RARAI = 0004            RCHBD0= 0048            RCHBD1= 0047
RCHOK = 0046            RCHTST= 000C            RCLRT = 000D            RCMD  = 000B
RCMLP = 0003            RCONT = 0080            RDATA = 0017            RDCLK = 0010
RDON  = 0011            READG = 0007            REND  = 0014            REQST = 2806
RESCHR= 00D1            REVTST= 0064            REWIND= 0004            RFIFOL= 0008
RGCLK = 0002            RGCRI = 0003            RIBG  = 0001            RILL  = 0012
RINST = 000C            RMCTST= 0008            RMK2  = 0013            RNOP  = 0000
RPATH = 0001            RPBAD = 0044            RPCHI = 00C1            RPCLK = 0003
RPCTL = 0009            RPEI  = 0002            RPFAIL= 0000            RPF1  = 009D
RPF2  = 009E            RPOK  = 0043            RPOSTN= 0016            RPSTA = 0015
RRCMT = 000A            RSTAT = 0002            RTIEB = 000A            RTIER = 0030
RTM   = 0005            RUNKI = 0009            RUPTST= 005E            RWDUNL  0005
R.AMT = 0001            R.BOP = 0008            R.DATA= 0040            R.DON = 0002
R.DRDY= 0010            R.END = 0010            R.ILL = 0004            R.JVOK= 0004
```

```
R.MK2 = 0008          R.PLOD= 0008          R.PLO0= 0010          R.PLO1= 0020
R.PUST= 0020          R.STNM= 0002          R.STOP= 0004          R.STPC= 0001
R.TBJN= 0080          R.TSTD= 0040          R.VOK = 0080          R00H  = 0081
R00L  = 0080          R01H  = 0083          R01L  = 0082          R02H  = 0085
R02L  = 0084          R03H  = 0087          R03L  = 0086          R04H  = 0089
R04L  = 0088          R05H  = 008B          R05L  = 008A          R06H  = 008D
R06L  = 008C          R07H  = 008F          R07L  = 008E          R10H  = 0091
R10L  = 0090          R11H  = 0093          R11L  = 0092          R12H  = 0095
R12L  = 0094          R13H  = 0097          R13L  = 0096          R14H  = 0099
R14L  = 0098          R15H  = 009B          R15L  = 009A          R16H  = 009D
R16L  = 009C          R17H  = 009F          R17L  = 009E          R7.5  = 0010
SELCLR= 0000          SETATA= 00A1          SID   = 0080          SOD   = 0080
SOE   = 0040          SP    =%0008          SSCLK = 0040          SSTEP = 0005
STACK = 4FFF          STATRM= 4F20          STPCT   4F20          STRSP = 5000
TADR00= 0080          TADR01= 0081          TADR02= 0082          TADR03= 0083
TADR04= 0084          TADR05= 0085          TADR06= 0086          TADR07= 0087
TADR10= 0088          TADR11= 0089          TADR12= 008A          TADR13= 008B
TAMT  = 0044          TASEL = 0080          TCMD  = 0040          TC.FWD= 0040
TC.INH= 0008          TC.LWR= 0004          TC.REV= 0020          TC.WRT= 0010
TEMP    4F99          TEST01  4300          TEST02  43D2          TEST03  4482
TEST04  4532          TEST05  45E2          TEST06  4640          TEST07  46A0
TEST10  4700          TEST11  4760          TEST12  478D          TEST13  47BC
TEST14  47EB          TEST15  481A          TMF   = 0099          TMRDY = 0040
TRKENA= 00D2          TSET  = 2803          TSTEND= 2818          TSTS  = 0040
TST01X  4327          TST02X  43D7          TST03X  4487          TST04X  4537
TST05X  45E7          TST06X  4645          TST07X  46A5          TST10X  4705
TST11X  476D          TST12X  479A          TST13X  47C9          TST14X  47F8
TUSEL0= 00D1          TUSEL1= 00D2          TU78  = 0010          T.ATH0= 0001
T.ATH1= 0002          T.BOT = 0004          T.EOT = 0002          T.FPT = 0001
T.NTHR= 0004          T.ONL = 0020          T.PES = 0008          T.PSBJ= 0020
T.PSOJ= 0008          T.PS1J= 0010          T.RDY = 0080          T.RDYO= 0040
T.RWD = 0010          T.SCLK= 0002          T1CN1   4363  G       T1CN2   4386  G
T1CN3   43A9  G       T1CN4   43CC  G       T2CN1   4413  G       T2CN2   4436  G
T2CN3   4459  G       T2CN4   447C  G       T3CN1   44C3  G       T3CN2   44E6  G
T3CN3   4509  G       T3CN4   452C  G       T4CN1   4573  G       T4CN2   4596  G
T4CN3   45B9  G       T4CN4   45DC  G       UIBG  = 00A1          UNITMP  49BA
VALFC   4F98          VALTB   4F95          VELTST= 005B          WDR.P = 0010
WMCCTL= 00D3          WMCERR= 00DA          WMCSTA= 00D0          WRTCLK= 0000
WRTDAT= 00D3          W.ACRC= 0004          W.CRC = 0008          W.DIAG= 0002
W.DONN= 0040          W.ECC = 0010          W.ENAB= 0080          W.ERR = 0020
W.FMT = 0070          W.GCR = 0010          W.LEFT= 0004          W.ONES= 0020
W.RESI= 0002          W.REV = 0004          W.ROME= 0010          W.RST = 0001
W.SKIP= 000F          W.WRIT= 0008          W.XFER= 0020          X     =%000A
X.DONN= 0080          X.ENAB= 0040          X.PEPE= 0002          X.ROME= 0001
X.WCLK= 0001          Y     =%000B          .       49BB


ERRORS DETECTED:  0

*MTA2.A78/PTP,MTA2=NLIST,PARAM,MACRO,LIST,MTA2
RUN-TIME: 5 8 0 SECONDS
CORE USED:  10K
```

3 "MEMORY MAP" EXTENDED ADDRESS BIT
19 READ MICRO CONTROLLER CONTROL & STATUS WORDS
139 TRANSPORT/PORT COMMAND + STATUS WORDS
380 COMMON ADDRESS SPACE REGISTERS
444 MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485 MASSBUS CONTROL WORD DEFINITIONS
557 WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709 KEYBOARD/DISPLAY CONTROL + STATUS WORD
797 MISCELLANEOUS VARIABLE DEFINITIONS
801 RAM ADDRESS DEFINITIONS
819 LINKAGE TABLE ADDRESSES
842 BIT DEFINITIONS
856 RIM AND SIM INSTRUCTION BIT DEFINITIONS
877 INTERRUPTABLE INSTRUCTION MACROS
924 DIAGNOSTIC VARIABLES
945 COMMON VARIABLE STORAGE
959 MACRO CALL DEFINITIONS
1047 TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101 8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1334 TEST 01 - TACH LINE CONTINUITY TEST - MTA #0
1391 TEST 02 - TACH LINE CONTINUITY TEST - MTA #1
1432 TEST 03 - TACH LINE CONTINUITY TEST - MTA #2
1473 TEST 04 - TACH LINE CONTINUITY TEST - MTA #3
1517 SUBROUTINE CLEAR ALL TU PORTS
1561 SUBROUTINE TACH COMMON
1604 PROGRAM VARIABLES

```
                ;*************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;               "BYTE/SCLK COUNT NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS PEGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               MM = DATA FORMAT FROM CAS REGISTER 2
                ;               NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;               BYTE-SCLK COUNT = LLL
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               LLL = AS ABOVE
                ;               MM  = AS ABOVE
                ;               NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;               TRANSITION COUNT = LLL
                ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;               EXPECTED 18 BITS =      E EEEEEE
                ;               ACTUAL 18 BITS =        A AAAAAA
                ;
                ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;               OF CAS REG 15 LOW BYTE.
                ;
                ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;               17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;               AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;               "SUBGROUP NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       10 -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;*************************************************************************
```

```
1016                    ;*****************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEP  HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    RFAD XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ;*****************************************************************************
1035                    ;*****************************************************************************
1036                    ;DATA PATTERN CODES
1037
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ;*****************************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                        .TITLE  MTA3 - MAG TAPE ADAPTER TEST PART #3
                                        ;ID MTA3-MAG TAPE ADAPTER CONTROLLER PART #3
                                        .SBTTL  TEST 01 - TACH LINE CONTINUITY TEST - MTA #0
4300                                    ST
 (1)                                    ; ************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
                                        ;*TACH LINE CONTINUITY TEST - MTA #0
4300                                    SD
 (1)                                    ; ************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS TEST CHECKS THAT THE TACH VALUE READ FROM THE MTA IS THE SAME AS
                                        ;*THE TACH VALUE READ FROM THE TU PORT.
4300                                    SP
 (1)                                    ; ************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNTST
                                        ;*  IF TU PORT #0 NOT SELECTED BY USER
                                        ;*  :  THEN-NEXT TEST
                                        ;*  :  ELSE-CONTINUE
                                        ;*  ENDIF
                                        ;*  CALL SUBROUTINE CLEAR
                                        ;*  SELECT TU PORT #0
                                        ;*  CALL SUBROUTINE TACOM
                                        ;*ENDTST
4300                                    SE
 (1)                                    ; ************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
                                        ;*MTA3 MICRO TEST 01
                                        ;*MTA3 MICRO ERROR 01
                                        ;*MTA TACH LINE CONTINUITY TEST - MTA #0
                                        ;*TU CABLE
                                        ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                        ;*FATAL ERROR - TEST ABORTED
                                        ;*
                                        ;*MTA3 MICRO TEST 01
                                        ;*MTA3 MICRO ERROR 02
                                        ;*MTA TACH LINE CONTINUITY TEST - MTA #0
                                        ;*TU CABLE
                                        ;*MTA TACH BIT VALUE NOT=TUP TACH BIT VALUE
                                        ;*ACTUAL DATA = TUP TACH BIT VALUE
                                        ;*EXPECTED DATA = MTA TACH BIT VALUE
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4300                                    S
 (1)                                    ; ************************************************************************
4300                                    TEST01: TESTX   @01             ;INITIALIZE THE TEST
 (1)  4300  3E  01            7.0                MVI     A,@01           ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                CALL    TSET            ;SETUP THE TEST
                                        ;%MTA TACH LINE CONTINUITY TEST - MTA #0
                                        ;&TU CABLE
```

```
1371  4305                                    REQ     @7,0,0,0,0
 (1)  4305  CD  06  28        18.0                     CALL    REQST
 (1)  4303  00                                         .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4309  00  00                                     .WORD   0               ;SYSTEM "0" COUNT
 (1)  430B  00  00                                     .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                         .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1)  430E  07                                         .BYTE   @7              ;REQUEST CODE
1372  430F                                    RIN     R12L                    ;GET THE UNITS DESIRED
 (1)  430F  DB  94            10.0                     IN      R12L                    ;READ R12L INTO AC
 (1)  4311  7F                 4.0                     MOV     A,A                     ;RETRY LINK
1373  4312  32  31  44        13.0             STA     UNITMP                  ;STORE IN MEMORY
1374  4315  A7                 4.0             ANA     A                       ;SET THE CONDITION CODE
1375  4316  C2  2F  43        10.0             JNZ     TST01X                  ;GO TEST THE SPECIFIED UNITS
1376                                                                          ;NO UNITS SPECIFIED
1377  4319                                    ERR     EXIT,DUMMY
 (1)                                          ;FLAG AN ERROR - NO EXFECTED OR ACTUAL
 (1)  4319  CD  09  28        18.0                     CALL    ERLP                    ;PROCESS ERROR - DO 2.3
 (1)        0001                                       MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                         .BYTE   MSGN                    ;MESSAGE NUMBER ID
 (1)  431D  00                                         .BYTE
 (1)  431E  CD  15  28        18.0             DUMMY:: CALL    CKLOP                   ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  DC  43        10.0                     JC      EXIT                    ;LOOP ADDRESS IF LOOP SPECIFIED
1378                                     ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1379                                     ;>FATAL ERROR - TEST ABORTED
1380  4324  C3  DC  43        10.0             JMP     EXIT                    ;EXIT THE TEST
1381  4327  3A  31  44        13.0             LDA     UNITMP                  ;GET THE UNIT MAP
1382  432A  E6  01                 7.0             ANI     @001                    ;TEST UNIT 0?
1383  432C  CA  4F  43        10.0             JZ      TEST02                  ;NO-GO CHECK TEST 2
1384  432F  CD  DF  43        18.0     TST01X: CALL    CLEAR
1385  4332  DB  E0                10.0             IN      INTSTA
1386  4334  E6  80                 7.0             ANI     BIT7
1387  4336  D3  E0                10.0             OUT     MBSEL
1388  4338  CD  0A  44        18.0             CALL    TACOM
1389  433B                                    ENDTST  TST01X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  433B                                    REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  433B  CD  06  28        18.0                     CALL    REQST
 (2)  433E  00                                         .BYTE                   ;DATA PATTERN NUMBER
 (2)  433F  00  00                                     .WORD                   ;SYSTEM " " COUNT
 (2)  4341  00  00                                     .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4343  00                                         .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4344  07                                         .BYTE   7               ;REQUEST CODE
 (1)  4345  3A  9A  4F        13.0                     LDA     ITERA                   ;GET ITERATION COUNT
 (1)  4348  3D                 4.0                     DCR     A                       ;DOWNCOUNT
 (1)  4349  32  9A  4F        13.0                     STA     ITERA                   ;SAVE COUNT
 (1)  434C  F2  2F  43        10.0                     JP      TST01X                  ;DO TEST UNTIL TILL = 0
```

```
                                        .SBTTL  TEST 02 - TACH LINE CONTINUITY TEST - MTA #1
434F                                    ST
 (1)                                    ; **********************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
                                        ;*TACH LINE CONTINUITY TEST - MTA #1
434F                                    SD
 (1)                                    ; **********************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS TEST CHECKS THAT THE TACH VALUE READ FROM THE MTA IS THE SAME AS
                                        ;*THE TACH VALUE READ FROM THE TU PORT.
434F                                    SP
 (1)                                    ; **********************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNTST
                                        ;*  IF TU PORT #1 NOT SELECTED BY USER
                                        ;*  :  THEN-NEXT TEST
                                        ;*  :  ELSE-CONTINUE
                                        ;*  ENDIF
                                        ;*  CALL SUBROUTINE CLEAR
                                        ;*  SELECT TU PORT #1
                                        ;*  CALL SUBROUTINE TACOM
                                        ;*ENDTST
434F                                    SE
 (1)                                    ; **********************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
                                        ;*MTA3 MICRO TEST 02
                                        ;*MTA3 MICRO ERROR 02
                                        ;*MTA TACH LINE CONTINUITY TEST - MTA #1
                                        ;*TU CABLE
                                        ;*MTA TACH BIT VALUE NOT=TUP TACH BIT VALUE
                                        ;*ACTUAL DATA = TUP TACH BIT VALUE
                                        ;*EXPECTED DATA = MTA TACH BIT VALUE
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
434F                                    S
 (1)                                    ; **********************************************************************
434F                                    TEST02: TESTX   @02             ;INITIALIZE THE TEST
 (1) 434F  3E  02            7.0                    MVI     A,@02       ;DEFINE THE TEST NUMBER
 (1) 4351  CD  03  28       18.0                    CALL    TSET        ;SETUP THE TEST
                                        ;%MTA TACH LINE CONTINUITY TEST - MTA #1
                                        ;&TU CABLE
4354  3A  31  44           13.0            LDA     UNITMP          ;GET THE UNIT MAP
4357  E6  02                7.0            ANI     @002            ;TEST UNIT 1?
4359  CA  7E  43           10.0            JZ      TEST03          ;NO-GO CHECK TEST 3
435C  CD  DF  43           18.0  TST02X:   CALL    CLEAR
435F  DB  E0               10.0            IN      INTSTA
4361  E6  80                7.0            ANI     BIT7
4363  F6  01                7.0            ORI     $01
4365  D3  E0               10.0            OUT     MBSFL
4367  CD  0A  44           18.0            CALL    TACOM
```

```
1430  436A                                        ENDTST  TST02X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  436A                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  436A    CD   06   28        18.0                                CALL    REQST
 (2)  436D    00                                                   .BYTE                   ;DATA PATTERN NUMBER
 (2)  436E    00   00                                              .WORD                   ;SYSTEM '*' COUNT
 (2)  4370    00   00                                              .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4372    00                                                   .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4373    07                                                   .BYTE   7               ;REQUEST CODE
 (1)  4374    3A   9A   4F        13.0            LDA     ITERA           ;GET ITERATION COUNT
 (1)  4377    3D                   4.0            DCR     A               ;DOWNCOUNT
 (1)  4378    32   9A   4F        13.0            STA     ITERA           ;SAVE COUNT
 (1)  437B    F2   5C   43        10.0            JP      TST02X          ;DO TEST UNTIL TILL = 0
```

```
                                        .SBTTL  TEST 03 - TACH LINE CONTINUITY TEST - MTA #2
437E                                    ST
 (1)                                    ; ***************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
                                        ;*TACH LINE CONTINUITY TEST - MTA #2
437E                                    SD
 (1)                                    ; ***************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS TEST CHECKS THAT THE TACH VALUE READ FROM THE MTA IS THE SAME AS
                                        ;*THE TACH VALUE READ FROM THE TU PORT.
437E                                    SP
 (1)                                    ; ***************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNTST
                                        ;*  IF TU PORT #2 NOT SELECTED BY USER
                                        ;*  :  THEN-NEXT TEST
                                        ;*  :  ELSE-CONTINUE
                                        ;*  ENDIF
                                        ;*  CALL SUBROUTINE CLEAR
                                        ;*  SELECT TU PORT #2
                                        ;*  CALL SUBROUTINE TACOM
                                        ;*ENDTST
437E                                    SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
                                        ;*MTA3 MICRO TEST 03
                                        ;*MTA3 MICRO ERROR 02
                                        ;*MTA TACH LINE CONTINUITY TEST - MTA #2
                                        ;*TU CABLE
                                        ;*MTA TACH BIT VALUE NOT=TUP TACH BIT VALUE
                                        ;*ACTUAL DATA = TUP TACH BIT VALUE
                                        ;*EXPECTED DATA = MTA TACH BIT VALUE
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
437E                                    S
 (1)                                    ; ***************************************************************************
437E                                    TEST03: TESTX   @03             ;INITIALIZE THE TEST
 (1) 437E  3E  03          7.0                          MVI     A,@03           ;DEFINE THE TEST NUMBER
 (1) 4380  CD  03  28     18.0                          CALL    TSET            ;SETUP THE TEST
                                        ;%MTA TACH LINE CONTINUITY TEST - MTA #2
                                        ;&TU CABLE
4383  3A  31  44           13.0                  LDA     UNITMP          ;GET THE UNIT MAP
4386  E6  04                7.0                  ANI     @004            ;TEST UNIT 2?
4388  CA  AD  43           10.0                  JZ      TEST04          ;NO-GO CHECK TEST 4
438B  CD  DF  43           18.0  TST03X: CALL    CLEAR
438E  DB  E0               10.0                  IN      INTSTA
4390  E6  80                7.0                  ANI     BIT7
4392  F6  02                7.0                  ORI     $02
4394  D3  E0               10.0                  OUT     MBSEL
4396  CD  0A  44           18.0                  CALL    TACOM
```

```
1471  4399                                        ENDTST  TST03X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4399                                       REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4399  CD  06  28        18.0                                 CALL    REQST
 (2)  439C  00                                                     .BYTE                   ;DATA PATTERN NUMBER
 (2)  439D  00  00                                                 .WORD                   ;SYSTEM "" COUNT
 (2)  439F  00  00                                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43A1  00                                                     .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  43A2  07                                                     .BYTE   7               ;REQUEST CODE
 (1)  43A3  3A  9A  4F        13.0               LDA     ITERA           ;GET ITERATION COUNT
 (1)  43A6  3D                 4.0               DCR     A               ;DOWNCOUNT
 (1)  43A7  32  9A  4F        13.0               STA     ITERA           ;SAVE COUNT
 (1)  43AA  F2  8B  43        10.0               JP      TST03X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 04 - TACH LINE CONTINUITY TES' - MTA #3
43AD                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*TACH LINE CONTINUITY TEST - MTA #3
43AD                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THAT THE TACH VALUE READ FROM THE MTA IS THE SAME AS
                                ;*THE TACH VALUE READ FROM THE TU PORT.
43AD                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  IF TU PORT #3 NOT SELECTED BY USER
                                ;*  :  THEN-NEXT TEST
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  CALL SUBROUTINE CLEAR
                                ;*  SELECT TU PORT #3
                                ;*  CALL SUBROUTINE TACOM
                                ;*ENDTST
43AD                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA3 MICRO TEST 04
                                ;*MTA3 MICRO ERROR 02
                                ;*MTA TACH LINE CONTINUITY TEST - MTA #3
                                ;*TU CABLE
                                ;*MTA TACH BIT VALUE NOT=TUP TACH BIT VALUE
                                ;*ACTUAL DATA = TUP TACH BIT VALUE
                                ;*EXPECTED DATA = MTA TACH BIT VALUE
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
43AD                            S
 (1)                            ; ***************************************************************************
43AD                            TEST04: TESTX   @04             ;INITIALIZE THE TEST
 (1) 43AD  3E  04        7.0                    MVI     A,@04           ;DEFINE THE TEST NUMBER
 (1) 43AF  CD  03  28   18.0                    CALL    TSET            ;SETUP THE TEST
                                ;%MTA TACH LINE CONTINUITY TEST - MTA #3
                                ;&TU CABLE
43B2  3A  31  44        13.0            LDA     UNITMP          ;GET THE UNIT MAP
43B5  E6  08             7.0            ANI     @010            ;TEST UNIT 3?
43B7  CA  DC  43        10.0            JZ      EXIT            ;NO-GO CHECK TEST 5
43BA  CD  DF  43        18.0    TST04X: CALL    CLEAR
43BD  DB  E0            10.0            IN      INTSTA
43BF  E6  80             7.0            ANI     BIT7
43C1  F6  03             7.0            ORI     $03
43C3  D3  E0            10.0            OUT     MBSEL
43C5  CD  0A  44        18.0            CALL    TACOM
```

```
1512  43C8                                    ENDTST  TST04X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43C8                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43C8  CD  06  28       18.0                             CALL    REQST
 (2)  43CB  00                                                .BYTE                   ;DATA PATTERN NUMBER
 (2)  43CC  00  00                                            .WORD                   ;SYSTEM '"' COUNT
 (2)  43CE  00  00                                            .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43D0  00                                                .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  43D1  07                                                .BYTE   7               ;REQUEST CODE
 (1)  43D2  3A  9A  4F       13.0             LDA     ITERA           ;GET ITERATION COUNT
 (1)  43D5  3D                4.0             DCR     A               ;DOWNCOUNT
 (1)  43D6  32  9A  4F       13.0             STA     ITERA           ;SAVE COUNT
 (1)  43D9  F2  BA  43       10.0             JP      TST04X          ;DO TEST UNTIL TILL = 0
1513
1514
1515  43DC  C3  18  28       10.0  EXIT:      JMP     TSTEND
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ; ************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ; ************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ; ************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*    CLEAR PORT SELECT FOR TRANSPORT
                                        ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*    CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ; ************************************************************************
                                        ;
                        12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
                        12.0          PUSH    B
                         7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
                        10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
                         7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
                         4.0          ORA     B               ;ADD IN THE SELECTED PORT #
                        10.0          OUT     MBSEL           ;RESET TO THIS PORT
                         7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
                        10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
                         4.0          XRA     A               ;CLEAR TU COMMAND A
                        10.0          OUT     TCMD            ;
                         7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
                        10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #
                         7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
                        10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
                         4.0          XRA     A
                        10.0          OUT     TAMT            ;CLEAR AMTIE WORD
                        10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
                        10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
                         4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
                         4.0          MOV     A,B
                         7.0          CPI     4               ;DONE?
                        10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
                        10.0          POP     B               ;RESET B & C
                        10.0          POP     PSW             ;ALL DONE
                        10.0          RET                     ;EXIT
```

```
                                .SBTTL  SUBROUTINE TACH COMMON
                                SSUB
                                ;*********************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*TACH CONTINUITY COMMON CODE
                                SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE CHECKS THE VALUE OF THE TACH BIT ON THE MTA BOARD AND
                                ;*EXPECTES THE TACH BIT ON THE TUP BOARD TO MATCH.
                                SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*  SELECT MTA REGISTER 5
                                ;*  INPUT THE MTA TACH BIT
                                ;*  INPUT THE TUP TACH BIT
                                ;*  IF MTA TACH BIT=TUP TACH BIT
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*ENDSUB
                                S
                                ;*********************************************************************
440A 3E 85          7.0 TACOM:  MVI     A,$85           ;LOAD MTA REGISTER 5 SELECT CODE
440C D3 40         10.0        OUT     TCMD
440E DB 40         10.0        IN      TSTS            ;INPUT THE TU DIAG REGISTER
4410 E6 08          7.0        ANI     BIT3            ;SAVE THE MTA TACH BIT
4412 1F             4.0        RAR
4413 1F             4.0        RAR
4414 1F             4.0        RAR
4415 47             4.0        MOV     B,A             ;SAVE IN B
4416 D3 95         10.0        OUT     EDATA           ;EDATA=MTA TACH BIT
4418 DB 48         10.0        IN      PSTAT           ;INPUT THE TU PORT TACH BIT
441A E6 08          7.0        ANI     BIT3
441C 1F             4.0        RAR
441D 1F             4.0        RAR
441E 1F             4.0        RAR
441F B8             4.0        CMP     B               ;BOTH THE SAME?
4420 CA 2A 44      10.0        JZ      TACON           ;YES-CONTINUE
4423 D3 94         10.0        OUT     ADATA           ;ADATA=TUP TACH BIT
```

```
1596  4425                                          ERRB    TACOM,TACON
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4425  CD   12   28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0002                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4428  02                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4429  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  442A  CD   15   28        18.0                TACON:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  442D  DA   0A   44        10.0                        JC      TACOM           ;LOOP ADDRESS IF LOOP SPECIFIED
1597                                          ;>MTA TACH BIT VALUE NOT=TUP TACH BIT VALUE
1598                                          ;>ACTUAL DATA = TUP TACH BIT VALUE
1599                                          ;>EXPECTED DATA = MTA TACH BIT VALUE
1600  4430  C9                    10.0                RET
1601
1602
1603
1604                                          .SBTTL  PROGRAM VARIABLES
1605  4431  00                                UNITMP: .BYTE   0
1606       0000                                       .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE   4F9B           ATTCD   4F97          A.NUM   4F91
B      =%0000          BADST  = 0090          BIT0   = 0001          BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008          BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080          BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00           BSAVE   4F9C           BYTCNT= 00D4          BYTEH   4F24
BYTEL   4F23           C      =%0001          CASCT   4F21          CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1
CBYTH  = 008B          CBYTL  = 008A          CDG1H  = 0087          CDG1L  = 0086
CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095          CDG3L  = 0094
CDVTH  = 008D          CDVTL  = 008C          CHPTIE= 0028          CHOTIE= 0020
CH1TIE= 0021           CH2TIE= 0022           CH3TIE= 0023          CH4TIE= 0024
CH5TIE= 0025           CH6TIE= 0026           CH7TIE= 0027          CKLOP  = 2815
CLEAR   43DF           CLKCTL= 00F0           CLOCK   4F26          CLRLP   43E3
CMC0H  = 0099          CMC0L  = 0098          CMC1H  = 009B          CMC1L  = 009A
CMC2H  = 009D          CMC2L  = 009C          CMC3H  = 009F          CMC3L  = 009E
CMINH  = 0097          CMINL  = 0096          CNTCTL= 00D7          CRCWRD= 0018
CSAVE   4F9D           CSRLH  = 0091          CSRLL  = 0090          CTCH   = 0085
CTCL   = 0084          CTSTH  = 008F          CTSTL  = 008E          CXCTH  = 0081
CXCTL  = 0080          CXINH  = 0083          CXINL  = 0082          C.     = 0001
C.AVAI= 0080           C.DP   = 0008          C.DSE  = 0010          C.DTU  = 0003
C.DVA  = 0008          C.FAIL= 00FC           C.FMT  = 0070          C.FNCT= 003E
C.GO   = 0001          C.INTC= 00FE           C.MAIN= 0020          C.NSA  = 0080
C.RCT  = 00FC          C.SER  = 0080          C.SHR  = 0040          C.SKPC= 000F
C.TAPE= 0040           C.WCS  = 0002          D      =%0002          DATACT= 00D0
DBUS    4F28           DBUSCT= 00C0           DBUSST= 00C0          DDRA   = 00D8
DDRAIN= 0010           DDRB   = 00D9          DDRBIN= 0002          DDRC   = 00DA
DDRCIN= 0001           DDRCO  = 0088          DDRCTL= 00DB          DIAFLG  4F22
DIAGPG= 4300           DIAGRM= 4F90           DIARD  = 000B          DONE1  = 0045
DONINT= 0010           DSAVE   4F9E           DSE    = 0006          DUMMY   431E  G
D.ATH0= 0001           D.ATH1= 0002           D.EOTD= 0010          D.LAGC= 0020
D.NOIW= 0040           D.NTHR= 0004           D.TACH= 0008          D.WR4  = 0080
E      =%0003          ECCBAD= 0042           ECCCOR= 0019          ECCOK  = 0041
ECCSTA= 001A           ECCTST= 000E           EDATA  = 0095         EOTCLR= 0003
ERFLG   4F93           ERLP   = 2809          ERLPA  = 280F         ERLPB  = 2812
ERLPE  = 280C          ERNUM   4F90           ERRCNT= 00D6          ESAVE   4F9F
EXIT    43DC           E.ACRC= 0010           E.AMT  = 0020         E.CDP  = 0080
E.CRC  = 0080          E.PNTR= 0008           E.RPE  = 0040         E.STEC= 0001
E.TTEC= 0002           E.UNC  = 0004          FIFORD= 006A          FORMAT  4F25
FWDTST= 0061           GCRID  = 0089          GCRSET= 0002          GOODTM= 0092
H      =%0004          HLSAVE  4FA0           IE     = 0008         INTSTA= 00E0
ITERA   4F9A           I.PWR  = 0020          I.RMPE= 0040          I5.5   = 0010
I6.5   = 0020          I7.5   = 0040          KCALL  = 005F         KCLR   = 007B
KDEP   = 003F          KENAB  = 0078          KEXAM  = 003E         KEYBRD= 00C8
KEY1   = 0078          KEY10  = 006D          KEY11  = 006E         KEY12  = 006F
KEY13  = 005C          KEY14  = 005D          KEY15  = 005E         KEY16  = 005F
KEY17  = 003C          KEY18  = 003D          KEY19  = 003E         KEY2   = 0079
KEY20  = 003F          KEY3   = 007A          KEY4   = 007B         KEY5   = 0074
KEY6   = 0075          KEY7   = 0076          KEY8   = 0077         KEY9   = 006C
KINTA  = 006F          KLDAD  = 003D          KN0    = 003C         KN1    = 005C
KN2    = 005D          KN3    = 005E          KN4    = 006C         KN5    = 006D
KN6    = 006E          KN7    = 0074          KN8    = 0075         KN9    = 0076
KU2    = 0079          KU3    = 007A          KU8    = 0077         L      =%0005
```

```
LBLANK= 000F        LCE   = 000B        LCH   = 000C        LCL   = 000D
LCP   = 000E        LC0   = 0000        LC1   = 0001        LC2   = 0002
LC3   = 0003        LC4   = 0004        LC5   = 0005        LC6   = 0006
LC7   = 0007        LC8   = 0008        LC9   = 0009        LDLEDA= 00CA
LDLEDB= 00CB        LDLEDC= 00CC        LDLEDD= 00CD        LDLEDE= 00CE
LDLEDF= 00CF        LKDIAG= 2800        LKKBD = 004C        LKKEY = 0049
LKLWMG= 0058        LKLWMP= 0055        LKLWPG= 0052        LKLWPP= 004F
LKMOD7= 0046        LKOPR = 0046        LPFLG   4F94        LPNUM   4F92
M     =%0006        MBSEL = 00E0        MB.A  = 0008        MB.B  = 0004
MEMTOP= 4FFF        MINUS = 000A        MM    = 8000        MSE   = 0008
MSGN  = 0002        MTACLR= 0000        MT.ARA= 0020        MT.CPE= 0080
MT.DSE= 0001        MT.FWD= 0040        MT.INH= 0008        MT.LWR= 0004
MT.MOT= 0002        MT.NWT= 0080        MT.PEC= 0040        MT.PSB= 0004
MT.PS0= 0001        MT.PS1= 0002        MT.REV= 0020        MT.WRT= 0010
MT.Z  = 0008        M.ATA = 0080        M.CAPE= 0020        M.CONT= 0080
M.DEM = 0020        M.EBL = 0010        M.EXC = 0008        M.FAIL= 0008
M.ILR = 0010        M.INIT= 0010        M.OCC = 0020        M.ONLI= 0001
M.PE  = 0040        M.PORT= 0080        M.RDEN= 0002        M.RDPE= 0008
M.RUN = 0004        M.SCLK= 0001        M.TRA = 0040        M.UNIT= 0007
M.WCLK= 0040        M.WCLN= 0080        M.WREN= 0080        M5.5  = 0001
M6.5  = 0002        M7.5  = 0004        NOTCAP= 0088        OKAY  = 00FF
OPRRAM= 4300        OPSTRT= 0058        OPVER = 0040        PADCNT= 00D5
PADCRC= 0080        PDIAG = 0048        PEID  = 008A        PENAB = 004C
PESET = 0001        PL    = 00B1        PRDD  = 004C        PRENF = 009C
PS    = 00B2        PSTAT = 0048        PSW   =%0009        P.AMTP= 0001
P.BCTC= 0040        P.CMDP= 0020        P.INTE= 0080        P.LCS = 0040
P.LWR = 0020        P.RDP = 0002        P.RPEN= 0004        P.RPST= 0002
P.RPOE= 0020        P.RP1E= 0010        P.RP2E= 0020        P.RP3E= 0010
P.SING= 0080        P.STAT= 0002        P.STPE= 0080        P.TACH= 0008
P.TUPR= 0010        P.WCSP= 0004        P.WDS = 0040        P.WFLP= 0001
P.WPEN= 0010        P.WPOE= 0008        P.WP1E= 0004        P.WP2E= 0008
P.WP3E= 0004        P.5VOK= 0002        QUE   = 281B        QUEM  = 281E
RAMT  = 0010        RARA  = 0006        RARAI = 0004        RCHBD0= 0048
RCHBD1= 0047        RCHOK = 0046        RCHTST= 000C        RCLRT = 000D
RCMD  = 000B        RCMLP = 0003        RCONT = 0080        RDATA = 0017
RDCLK = 0010        RDON  = 0011        READG = 0007        REND  = 0014
REQST = 2806        RESCHR= 00D1        REVTST= 0064        REWIND= 0004
RFIFOL= 0008        RGCLK = 0002        RGCRI = 0003        RIBG  = 0001
RILL  = 0012        RINST = 000C        RMCTST= 0008        RMK2  = 0013
RNOP  = 0000        RPATH = 0001        RPEAD = 0044        RPCHI = 0001
RPCLK = 0003        RPCTL = 0009        RPEI  = 0002        RPFAIL= 0000
RPF1  = 009D        RPF2  = 009E        RPOK  = 0043        RPOSTN= 0016
RPSTA = 0015        RRCMT = 000A        RSTAT = 0002        RTIEB = 000A
RTIER = 0030        RTM   = 0005        RUNKI = 0009        RUPTST= 005E
RWDUNL= 0005        R.AMT = 0001        R.BOP = 0008        R.DATA= 0040
R.DON = 0002        R.DRDY= 0010        R.END = 0010        R.ILL = 0004
R.JVOK= 0004        R.MK2 = 0008        R.PLOD= 0008        R.PLOO= 0010
R.PLO1= 0020        R.POST= 0020        R.STNM= 0002        R.STOP= 0004
R.STPC= 0001        R.TBJN= 0080        R.TSTD= 0040        R.VOK = 0080
R00H  = 0081        R00L  = 0080        R01H  = 0083        R01L  = 0082
R02H  = 0085        R02L  = 0084        R03H  = 0087        R03L  = 0086
R04H  = 0089        R04L  = 0088        R05H  = 008B        R05L  = 008A
R06H  = 008D        R06L  = 008C        R07H  = 008F        R07L  = 008E
R10H  = 0091        R10L  = 0090        R11H  = 0093        R11L  = 0092
```

```
R12H  = 0095          R12L  = 0094          R13H  = 0097          R13L  = 0096
R14H  = 0099          R14L  = 0098          R15H  = 009B          R15L  = 009A
R16H  = 009D          R16L  = 009C          R17H  = 009F          R17L  = 009E
R7.5  = 0010          SELCLR= 0000          SETATA= 00A1          SID   = 0080
SOD   = 0080          SOE   = 0040          SP    =%0008          SSCLK = 0040
SSTEP = 0005          STACK = 4FFF          STATRM= 4F20          STPCT   4F20
STRSP = 5000          TACOM   440A          TACON   442A  G       TADR00= 0080
TADR01= 0081          TADR02= 0082          TADR03= 0083          TADR04= 0084
TADR05= 0085          TADR06= 0086          TADR07= 0087          TADR10= 0088
TADR11= 0089          TADR12= 008A          TADR13= 008B          TAMT  = 0044
TASEL = 0080          TCMD  = 0040          TC.FWD= 0040          TC.INH= 0008
TC.LWR= 0004          TC.REV= 0020          TC.WRT= 0010          TEMP    4F99
TEST01  4300          TEST02  434F          TEST03  437E          TEST04  43AD
TMF   = 0099          TMRDY = 0040          TRKENA= 00D2          TSET  = 2803
TSTEND= 2818          TSTS  = 0040          TST01X  432F          TST02X  435C
TST03X  438B          TST04X  43BA          TUSEL0= 00D1          TUSEL1= 00D2
TU78  = 0010          T.ATH0= 0001          T.ATH1= 0002          T.BOT = 0004
T.EOT = 0002          T.FPT = 0001          T.NTHR= 0004          T.ONL = 0020
T.PES = 0008          T.PSBJ= 0020          T.PS0J= 0008          T.PS1J= 0010
T.RDY = 0080          T.RDY0= 0040          T.RWD = 0010          T.SCLK= 0002
UIBG  = 00A1          UNITMP  4431          VALFC   4F98          VALTB   4F95
VELTST= 005B          WDR.P = 0010          WMCCTL= 00D3          WMCERR= 00DA
WMCSTA= 00D0          WRTCLK= 0000          WRTDAT= 00D3          W.ACRC= 0004
W.CRC = 0008          W.DIAG= 0002          W.DONN= 0040          W.ECC = 0010
W.ENAB= 0080          W.ERR = 0020          W.FMT = 0070          W.GCR = 0010
W.LEFT= 0004          W.ONES= 0020          W.RESI= 0002          W.REV = 0004
W.ROME= 0010          W.RST = 0001          W.SKIP= 000F          W.WRIT= 0008
W.XFER= 0020          X     =%000A          X.DONN= 0080          X.ENAB= 0040
X.PEPE= 0002          X.ROME= 0001          X.WCLK= 0001          Y     =%000B
.     = 4432


ERRORS DETECTED:  0

*MTA3.A78/PTP,MTA3=NLIST,PARAM,MACRO,LIST,MTA3
RUN-TIME: 3 3 0 SECONDS
CORE USED:  10K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(25) 15-OCT-80 09:35 SEQ 0424
PARAM.M80 TABLE OF CONTENTS

3	"MEMORY MAP" EXTENDED ADDRESS BIT
19	READ MICRO CONTROLLER CONTROL & STATUS WORDS
139	TRANSPORT/PORT COMMAND + STATUS WORDS
380	COMMON ADDRESS SPACE REGISTERS
444	MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485	MASSBUS CONTROL WORD DEFINITIONS
557	WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709	KEYBOARD/DISPLAY CONTROL + STATUS WORD
797	MISCELLANEOUS VARIABLE DEFINITIONS
801	RAM ADDRESS DEFINITIONS
819	LINKAGE TABLE ADDRESSES
842	BIT DEFINITIONS
856	RIM AND SIM INSTRUCTION BIT DEFINITIONS
877	INTERRUPTABLE INSTRUCTION MACROS
924	DIAGNOSTIC VARIABLES
945	COMMON VARIABLE STORAGE
959	MACRO CALL DEFINITIONS
1047	TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101	8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1333	TEST 1 - XMC - SELF TEST
1397	TEST 2 - XMC - TRACK ENABLE TEST
2226	SUBROUTINE CLEAR ALL TU PORTS

```
961                     ;*************************************************************************
962                     ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
963                     ;ERROR MACRO CALLS
964                     ;
965                     ;       1  -    REQUEST HOST CPU TO PRINT:
966                     ;               "BYTE/SCLK COUNT NUMBER = LLL"
967                     ;               WHERE:
968                     ;               LLL = THE VALUE STORED IN CAS REGISTER 5
969                     ;                     (THE BYTE COUNT REGISTER 16 BITS).
970                     ;
971                     ;       2  -    REQUEST HOST CPU TO PRINT:
972                     ;               DATA FORMAT = MM
973                     ;               SKIP COUNT  = NN
974                     ;               WHERE:
975                     ;               MM = DATA FORMAT FPOM CAS REGISTER 2
976                     ;               NN = SKIP COUNT FROM CAS REGISTER 2
977                     ;
978                     ;       3  -    REQUEST HOST CPU TO PRINT:
979                     ;               BYTE-SCLK COUNT = LLL
980                     ;               DATA FORMAT = MM
981                     ;               SKIP COUNT  = NN
982                     ;               WHERE:
983                     ;               LLL = AS ABOVE
984                     ;               MM  = AS ABOVE
985                     ;               NN  = AS ABOVE
986                     ;
987                     ;       4  -    REQUEST HOST TO PRINT:
988                     ;               TRANSITION COUNT = LLL
989                     ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
990                     ;
991                     ;       5  -    REQUEST HOST CPU TO PRINT:
992                     ;               EXPECTED 18 BITS =      E EEEEEE
993                     ;               ACTUAL 18 BITS =        A AAAAAA
994                     ;
995                     ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
996                     ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
997                     ;               OF CAS REG 15 LOW BYTE.
998                     ;
999                     ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
1000                    ;               AND IN REG 17 (FITS 17-16)...PRINT FLAG IS CAS REG
1001                    ;               17 LOW BYTE SIG. BIT.
1002                    ;
1003                    ;       6  -    REQUEST THE HOS  CPU TO PRINT BOTH ROUTINE #1 AND #5
1004                    ;               TO REPORT THE B/TE/SCLK COUNT AND 18 BITS OF EXPECTED
1005                    ;               AND/OR ACTUAL DATA.
1006                    ;
1007                    ;       7  -    REQUEST HOST CPU TO PRINT:
1008                    ;               "SUBGROUP NUMBER = LLL"
1009                    ;               WHERE:
1010                    ;               LLL = THE VALUE STORED IN CAS REGISTER 5
1011                    ;                     (THE BYTE COUNT REGISTER 16 BITS).
1012                    ;
1013                    ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
1014                    ;*************************************************************************
```

```
1016                    ;*************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ;*************************************************************************
1035                    ;*************************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ;*************************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                        .TITLE  XMC1 - TRANSLATOR MICRO CONTROLLER PART #1
                                        .SBTTL  TEST 1 - XMC - SELF TEST
                                        ;ID     XMC1-TRANSLATOR MICRO CONTROLLER PART #1
 4300                                   ST
                                        ; ************************************************************************
                                        ;*TEST TITLE
                                        ;*---- -----
                                        ;*XMC - SELF TEST
 4300                                   SD
                                        ; ************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS TEST COMMANDS THE TRANSLATOR MICRO CONTROLLER TO RUN ITS INTERNAL
                                        ;*SELF TEST DIAGNOSTIC PACKAGE.  A SOFTWARE DELAY IS PERFORMED AND THE
                                        ;*XMC ROM PARITY BIT IS TESTED.  IF THE PARITY BIT IS RESET NO SELF TEST
                                        ;*ERROR OCCURED, BUT IF SET EITHER A SELF TEST ERROR WAS DETECTED OR A
                                        ;*ROM PARITY ERROR DETECTED.
 4300                                   SP
                                        ; ************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNTST
                                        ;*  RESET THE MTA IF ONE SELECTED
                                        ;*  SET THE WMC/XMC RESTART BIT
                                        ;*  CLEAR CLOCK CONTROL BYTE
                                        ;*  CLEAR THE WMC/XMC RESTART BIT
                                        ;*  DELAY 1 MS
                                        ;*  IF XMC ROM PARITY ERROR=1
                                        ;*  :  THEN-ERROR
                                        ;*  :  ELSE-CONTINUE
                                        ;*  ENDIF
                                        ;*ENDTST
 4300                                   SE
                                        ; ************************************************************************
                                        ;*ERRORS
                                        ;*------
                                        ;*XMC1 MICRO TEST 01
                                        ;*XMC1 MICRO ERROR 01
                                        ;*XMC1-XMC-SELF TEST
                                        ;*M8958, M8959, M8960
                                        ;*XMC ROM PARITY ERROR SET AFTER XMC DISABLE INDICATES XMC SELF
                                        ;*DIAG. FAILURE OR XMC ROM PARITY ERROR.
 4300                                   S
                                        ; ************************************************************************

 4300                                   TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 4300 3E 01              7.0                            MVI     A,@1            ;DEFINE THE TEST NUMBER
 4302 CD 03 28          18.0                            CALL    TSET            ;SETUP THE TEST
                                        ;%XMC1-XMC-SELF TEST
                                        ;&M8958, M8959, M8960

 4305 AF                 4.0            XRA     A               ;CLEAR ALL TRACK ENABLE LINES
 4306 D3 D2             10.0            OUT     TRKENA
 4308 D3 D2             10.0            OUT     TRKENA
```

```
1371  43??  3E  40        7.0          MVI   A,P.LCS          ;GET RID OF MTA ON THE SYSTEM
1372  4???  D3  48       10.0          OUT   PDIAG
1373  430E  3E  00        7.0          MVI   A,@0             ;SET TO 0
1374  4310  D3  40       10.0          OUT   TCMD             ;CLEAR FOR ANY INTERFEARENCE
1375
1376  4??2  3E  01        7.0  XLLP1:  MVI   A,W.RST          ;CLEAR THE XMC ENABLE BIT
1377  431?  D3  D3       10.0          OUT   WMCCTL           ;DISABLE THE XMC
1378  4316  3E  00        7.0          MVI   A,0
1379  4318  D3  F0       10.0          OUT   CLKCTL           ;SET CLOCK CONTROL TO NORMAL OPERATION
1380  431A  AF           4.0          XRA   A
1381  431B  D3  D3       10.0          OUT   WMCCTL           ;FINISH THE RESTART OF WMC/XMC
1382                                                         ;XMC SHOULD START SELF TEST
1383  431D  3E  FF        7.0          MVI   A,@377           ;LOAD A COUNT VALUE
1384  431F  00            4.0  1$:     NOP                    ;NOP
1385  4320  3D            4.0          DCR   A                ;DECREMENT COUNT
1386  4321  C2  1F  43    10.0          JNZ   1$               ;CONTINUE UNTIL COUNT = 0
1387                                                         ;
1388  4324  DB  E0       10.0          IN    INTSTA           ;GET THE INTERRUPT STATUS BYTE
1389  4326  E6  01        7.0          ANI   X.ROME           ;XMC ROM PARITY ERROR?
1390  4328  CA  30  43    10.0          JZ    XLCN1            ;CONTINUE IF CLEAR
1391                                                         ;
1392  432B                               ERR   XLLP1,XLCN1
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  432B  CD  09  28    18.0                CALL  ERLP             ;PROCESS ERROR - DO 2.3
 (1)       0001                             MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  432E  01                              .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  432F  00                              .BYTE
 (1)  4330  CD  15  28    18.0          XLCN1:: CALL  CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4333  DA  12  43    10.0                JC    XLLP1            ;LOOP ADDRESS IF LOOP SPECIFIED
1393                                   ;>XMC ROM PARITY ERROR SET AFTER XMC DISABLE INDICATES XMC SELF
1394                                   ;>DIAG. FAILURE OR XMC ROM PARITY ERROR.
1395  4336                               ENDTST  XLLP1
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4336                             REQ   7                ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4336  CD  06  28    18.0                CALL  REQST
 (2)  4339  00                              .BYTE                  ;DATA PATTERN NUMBER
 (2)  433A  00  00                          .WORD                  ;SYSTEM '"' COUNT
 (2)  433C  00  00                          .WORD                  ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  433E  00                              .BYTE                  ;DATA COMPARE FLAG IF -1
 (2)  433F  07                              .BYTE 7                ;REQUEST CODE
 (1)  4340  3A  9A  4F    13.0          LDA   ITERA            ;GET ITERATION COUNT
 (1)  4343  3D            4.0          DCR   A                ;DOWNCOUNT
 (1)  4344  32  9A  4F    13.0          STA   ITERA            ;SAVE COUNT
 (1)  4347  F2  12  43    10.0          JP    XLLP1            ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 2 - XMC - TRACK ENABLE TEST
                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - TRACK ENABLE TEST
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST LOADS ALL ONES IN THE TRACK ENABLE REGISTER AND COMMANDS THE
                                ;*TRANSLATOR MICRO CONTROLLER TO WRITE GCR ALL ONES.  THE TRANSLATOR MICRO
                                ;*CONTROLLER IS THEN SINGLE STEPPED UNTIL THE TRANSLATOR MICRO CONTROLLER
                                ;*(XMC) STATUS BIT "XL WCLK" SETS.  THEN THE OUTPUT OF THE TRANSLATOR IS
                                ;*CHECKED AT THE TU PORT BOARD FOR THE FIRST DATA STATE OF THE NRZI FORMAT.
                                ;*THEN THE XMC IS CLOCKED 8 MORE TIMES AND THE OUTPUT OF THE TRANSLATOR
                                ;*IS CHECKED AT THE TU PORT BOARD FOR THE SECOND DATA STATE OF THE NRZI
                                ;*FORMAT.
                                ;*
                                ;*ONCE THE ABOVE SEQUENCE IS COMPLETE THE TEST PROCEEDS TO DISABLE EACH
                                ;*OF THE NINE TRACKS ONE AT A TIME UNTIL ALL NINE ARE DISABLED.  AFTER
                                ;*EACH TRACK IS DISABLED THE NRZI DATA STATES ARE CHECKED AT THE TJ PORT
                                ;*BOARD BEFORE THE NEXT TRACK IS DISABLED.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  REQUEST TU PORTS TO USE FOR TEST
                                ;*  SET WRITE PATH ENABLE FOR ALL SELECTED TU PORTS
                                ;*  SET THE TRACK ENABLE MASK TO ALL ONES (9 BITS)
                                ;*  SET THE PASSCOUNT TO 1
                                ;*  SET THE EXPECTED GCR DATA TO ALL ONES (9 BITS)
                                ;*  SET THE WMC/XMC RESTAR TBIT
                                ;*  SELECT THE SINGLE STEP WRITE CLOCK
                                ;*  CLEAR THE WMC/XMC RESTART BIT
                                ;*  CLEAR THE TRACK ENABLE REGISTER
                                ;*  SET WMC CONTROL BITS X.ENABLE, ONES, GCR AND WRITE
                                ;   BGNDO
                                ;*  :  CALL SUBROUTINE ADJDAT
                                ;*  :  ISSUE THE TRACK ENABLE MASK TO THE TRACK ENABLE REGISTER
                                ;*  :  SET THE LOOP COUNT TO 177(8)
                                ;*  :  BGNDO
                                ;*  :  :  CALL SUBROUTINE CLKSYS
                                ;*  :  :  DECREMENT THE LOOP COUNT
                                ;*  :  :  DO UNTIL THE LOOP COUNT=0 OR XL WRITE CLOCK SET
                                ;*  :  ENDDO
                                ;*  :  IF LOOP COUNT=0
                                ;*  :  :  THEN-ERROR-TIMEOUT WAITING FOR XL WRITE CLOCK
                                ;*  :  :  ELSE-CONTINUE
                                ;*  :  ENDIF
                                ;*  :  CALL SUBROUTINE CLKSYS
                                ;*  :  CALL SUBROUTINE CLKSYS
                                ;*  :  IF TU PORT #0 SELECTED BY THE USER
```

```
1442                    ;*  :  :  THEN-INPUT THE TRANSLATED DATA FROM TU PORT #0
1443                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA=ACTUAL TRANSLATED DATA
1444                    ;*  :  :  :  :  THEN-CONTINUE
1445                    ;*  :  :  :  :  ELSE-ERROR-DATA STATE NOT=EXPECTED
1446                    ;*  :  :  :  ENDIF
1447                    ;*  :  :  :  INPUT THE TRANSLATED DATA PARITY FROM TU PORT #0
1448                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA PARITY=ACTUAL TRANSLATED DATA PARITY
1449                    ;*  :  :  :  :  THEN-CONTINUE
1450                    ;*  :  :  :  :  ELSE-ERROR-PARITY DATA STATE NOT=EXPECTED
1451                    ;*  :  :  :  ENDIF
1452                    ;*  :  :  ELSE-CONTINUE
1453                    ;*  :  ENDIF
1454                    ;*  :  IF TU PORT #1 SELECTED BY THE USER
1455                    ;*  :  :  THEN-INPUT THE TRANSLATED DATA FROM TU PORT #1
1456                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA=ACTUAL TRANSLATED DATA
1457                    ;*  :  :  :  :  THEN-CONTINUE
1458                    ;*  :  :  :  :  ELSE-ERROR-DATA STATE NOT=EXPECTED
1459                    ;*  :  :  :  ENDIF
1460                    ;*  :  :  :  INPUT THE TRANSLATED DATA PARITY FROM TU PORT #1
1461                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA PARITY=ACTUAL TRANSLATED DATA PARITY
1462                    ;*  :  :  :  :  THEN-CONTINUE
1463                    ;*  :  :  :  :  ELSE-ERROR-PARITY DATA STATE NOT=EXPECTED
1464                    ;*  :  :  :  ENDIF
1465                    ;*  :  :  ELSE-CONTINUE
1466                    ;*  :  ENDIF
1467                    ;*  :  IF TU PORT #2 SELECTED BY THE USER
1468                    ;*  :  :  THEN-INPUT THE TRANSLATED DATA FROM TU PORT #2
1469                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA=ACTUAL TRANSLATED DATA
1470                    ;*  :  :  :  :  THEN-CONTINUE
1471                    ;*  :  :  :  :  ELSE-ERROR-DATA STATE NOT=EXPECTED
1472                    ;*  :  :  :  ENDIF
1473                    ;*  :  :  :  INPUT THE TRANSLATED DATA PARITY FROM TU PORT #2
1474                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA PARITY=ACTUAL TRANSLATED DATA PARITY
1475                    ;*  :  :  :  :  THEN-CONTINUE
1476                    ;*  :  :  :  :  ELSE-ERROR-PARITY DATA STATE NOT=EXPECTED
1477                    ;*  :  :  :  ENDIF
1478                    ;*  :  :  ELSE-CONTINUE
1479                    ;*  :  ENDIF
1480                    ;*  :  IF TU PORT #3 SELECTED BY THE USER
1481                    ;*  :  :  THEN-INPUT THE TRANSLATED DATA FROM TU PORT #3
1482                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA=ACTUAL TRANSLATED DATA
1483                    ;*  :  :  :  :  THEN-CONTINUE
1484                    ;*  :  :  :  :  ELSE-ERROR-DATA STATE NOT=EXPECTED
1485                    ;*  :  :  :  ENDIF
1486                    ;*  :  :  :  INPUT THE TRANSLATED DATA PARITY FROM TU PORT #3
1487                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA PARITY=ACTUAL TRANSLATED DATA PARITY
1488                    ;*  :  :  :  :  THEN-CONTINUE
1489                    ;*  :  :  :  :  ELSE-ERROR-PARITY DATA STATE NOT=EXPECTED
1490                    ;*  :  :  :  ENDIF
1491                    ;*  :  :  ELSE-CONTINUE
1492                    ;*  :  ENDIF
1493                    ;*  :  CALL SUBROUTINE COMP
1494                    ;*  :  CALL SUBROUTINE ADJDAT
1495                    ;*  :  SET THE LOOP COUNT TO 14(8)
```

```
1496                                    ;*  :  BGNDO
1497                                    ;*  :  :  CALL SUBROUTINE CLKSYS
1498                                    ;*  :  :  DECREMENT THE LOOP COUNT
1499                                    ;*  :  :  DO UNTIL THE LOOP COUNT=0
1500                                    ;*  :  ENDDO
1501                                    ;*  :  IF XL WRITE CLOCK=1
1502                                    ;*  :  :  THEN-CONTINUE
1503                                    ;*  :  :  ELSE-ERROR-XL WRITE CLOCK LATE
1504                                    ;*  :  ENDIF
1505                                    ;*  :  CALL SUBROUTINE CLKSYS
1506                                    ;*  :  CALL SUBROUTINE CLKSYS
1507                                    ;*  :  IF TU PORT #0 SELECTED BY THE USER
1508                                    ;*  :  :  THEN-INPUT THE TRANSLATED DATA FROM TU PORT #0
1509                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA=ACTUAL TRANSLATED DATA
1510                                    ;*  :  :  :  :  THEN-CONTINUE
1511                                    ;*  :  :  :  :  ELSE-ERROR-DATA STATE NOT=EXPECTED
1512                                    ;*  :  :  :  ENDIF
1513                                    ;*  :  :  :  INPUT THE TRANSLATED DATA PARITY FROM TU PORT #0
1514                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA PARITY=ACTUAL TRANSLATED DATA PARITY
1515                                    ;*  :  :  :  :  THEN-CONTINUE
1516                                    ;*  :  :  :  :  ELSE-ERROR-PARITY DATA STATE NOT=EXPECTED
1517                                    ;*  :  :  :  ENDIF
1518                                    ;*  :  :  ELSE-CONTINUE
1519                                    ;*  :  ENDIF
1520                                    ;*  :  IF TU PORT #1 SELECTED BY THE USER
1521                                    ;*  :  :  THEN-INPUT THE TRANSLATED DATA FROM TU PORT #1
1522                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA=ACTUAL TRANSLATED DATA
1523                                    ;*  :  :  :  :  THEN-CONTINUE
1524                                    ;*  :  :  :  :  ELSE-ERROR-DATA STATE NOT=EXPECTED
1525                                    ;*  :  :  :  ENDIF
1526                                    ;*  :  :  :  INPUT THE TRANSLATED DATA PARITY FROM TU PORT #1
1527                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA PARITY=ACTUAL TRANSLATED DATA PARITY
1528                                    ;*  :  :  :  :  THEN-CONTINUE
1529                                    ;*  :  :  :  :  ELSE-ERROR-PARITY DATA STATE NOT=EXPECTED
1530                                    ;*  :  :  :  ENDIF
1531                                    ;*  :  :  ELSE-CONTINUE
1532                                    ;*  :  ENDIF
1533                                    ;*  :  IF TU PORT #2 SELECTED BY THE USER
1534                                    ;*  :  :  THEN-INPUT THE TRANSLATED DATA FROM TU PORT #2
1535                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA=ACTUAL TRANSLATED DATA
1536                                    ;*  :  :  :  :  THEN-CONTINUE
1537                                    ;*  :  :  :  :  ELSE-ERROR-DATA STATE NOT=EXPECTED
1538                                    ;*  :  :  :  ENDIF
1539                                    ;*  :  :  :  INPUT THE TRANSLATED DATA PARITY FROM TU PORT #2
1540                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA PARITY=ACTUAL TRANSLATED DATA PARITY
1541                                    ;*  :  :  :  :  THEN-CONTINUE
1542                                    ;*  :  :  :  :  ELSE-ERROR-PARITY DATA STATE NOT=EXPECTED
1543                                    ;*  :  :  :  ENDIF
1544                                    ;*  :  :  ELSE-CONTINUE
1545                                    ;*  :  ENDIF
1546                                    ;*  :  IF TU PORT #3 SELECTED BY THE USER
1547                                    ;*  :  :  THEN-INPUT THE TRANSLATED DATA FROM TU PORT #3
1548                                    ;*  :  :  :  IF EXPECTED TRANSLATED DATA=ACTUAL TRANSLATED DATA
1549                                    ;*  :  :  :  :  THEN-CONTINUE
```

```
1550                                ;*  :  :  :  : ELSE-ERROR-DATA STATE NOT=EXPECTED
1551                                ;*  :  :  :  ENDIF
1552                                ;*  :  :  :  INPUT THE TRANSLATED DATA PARITY FROM TU PORT #3
1553                                ;*  :  :  :  IF EXPECTED TRANSLATED DATA PARITY=ACTUAL TRANSLATED DATA PARITY
1554                                ;*  :  :  :  : THEN-CONTINUE
1555                                ;*  :  :  :  : ELSE-ERROR-PARITY DATA STATE NOT=EXPECTED
1556                                ;*  :  :  : ENDIF
1557                                ;*  :  : ELSE-CONTINUE
1558                                ;*  : ENDIF
1559                                ;*  : CALL SUBROUTINE COMP
1560                                ;*  : INCREMENT THE PASS COUNT
1561                                ;*  : ROTATE THE TRACK ENABLE MASK RIGHT ONE BIT POSITION
1562                                ;*  : DO UNTIL THE TRACK ENABLE MASK=0
1563                                ;* ENDDO
1564                                ;*ENDTST
1565  434A                         SE
 (1)                                ; ************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1566                                ;*XMC1 MICRO TEST 02
1567                                ;*XMC1 MICRO ERROR 02
1568                                ;*XMC1-XMC TRACK ENABLE TEST-GCR
1569                                ;*M8958, M8959, M8960, M8955'S
1570                                ;*OPERATOR ERROR - NO TM78 UNITS SELECTED
1571                                ;*FATAL ERROR - TEST ABORTED
1572                                ;*
1573                                ;*XMC1 MICRO TEST 02
1574                                ;*XMC1 MICRO ERROR 03
1575                                ;*XMC1-XMC TRACK ENABLE TEST-GCR
1576                                ;*M8958, M8959, M8960, M8955'S
1577                                ;*XMC TIMEOUT WHILE WAITING FOR "XL WCLK" TO SET
1578                                ;*FATAL ERROR - TEST ABORTED
1579                                ;*
1580                                ;*XMC1 MICRO TEST 02
1581                                ;*XMC1 MICRO ERROR 04
1582                                ;*XMC1-XMC TRACK ENABLE TEST-GCR
1583                                ;*M8958, M8959, M8960, M8955'S
1584                                ;*FIRST GCR DATA STATE NOT AS EXPECTED
1585                                ;*WHEN READ FROM TUO STATUS BYTE
1586                                ;*ACTUAL = NNNN
1587                                ;*EXPECTED = NNNN
1588                                ;*
1589                                ;*XMC1 MICRO TEST 02
1590                                ;*XMC1 MICRO ERROR 05
1591                                ;*XMC1-XMC TRACK ENABLE TEST-GCR
1592                                ;*M8958, M8959, M8960, M8955'S
1593                                ;*FIRST GCR PARITY STATE NOT AS EXPECTED
1594                                ;*WHEN READ FROM "WCSP" BIT OF PORT 0 STATUS BYTE
1595                                ;*ACTUAL = NNNN
1596                                ;*EXPECTED = NNNN
1597                                ;*
1598                                ;*XMC1 MICRO TEST 02
1599                                ;*XMC1 MICRO ERROR 06
1600                                ;*XMC1-XMC TRACK ENABLE TEST-GCR
```

```
1601                    ;*M8958, M8959, M8960, M8955'S
1602                    ;*FIRST GCR DATA STATE NOT AS EXPECTED
1603                    ;*WHEN READ FROM TU1 STATUS BYTE
1604                    ;*ACTUAL = NNNN
1605                    ;*EXPECTED = NNNN
1606                    ;*
1607                    ;*XMC1 MICRO TEST 02
1608                    ;*XMC1 MICRO ERROR 07
1609                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1610                    ;*M8958, M8959, M8960, M8955'S
1611                    ;*FIRST GCR PARITY STATE NOT AS EXPECTED
1612                    ;*WHEN READ FROM ''WCSP'' BIT OF PORT 1 STATUS BYTE
1613                    ;*ACTUAL = NNNN
1614                    ;*EXPECTED = NNNN
1615                    ;*
1616                    ;*XMC1 MICRO TEST 02
1617                    ;*XMC1 MICRO ERROR 10
1618                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1619                    ;*M8958, M8959, M8960, M8955'S
1620                    ;*FIRST GCR DATA STATE NOT AS EXPECTED
1621                    ;*WHEN READ FROM TU2 STATUS BYTE
1622                    ;*ACTUAL = NNNN
1623                    ;*EXPECTED = NNNN
1624                    ;*
1625                    ;*XMC1 MICRO TEST 02
1626                    ;*XMC1 MICRO ERROR 11
1627                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1628                    ;*M8958, M8959, M8960, M8955'S
1629                    ;*FIRST GCR PARITY STATE NOT AS EXPECTED
1630                    ;*WHEN READ FROM ''WCSP'' BIT OF PORT 2 STATUS BYTE
1631                    ;*ACTUAL = NNNN
1632                    ;*EXPECTED = NNNN
1633                    ;*
1634                    ;*XMC1 MICRO TEST 02
1635                    ;*XMC1 MICRO ERROR 12
1636                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1637                    ;*M8958, M8959, M8960, M8955'S
1638                    ;*FIRST GCR DATA STATE NOT AS EXPECTED
1639                    ;*WHEN READ FROM TU3 STATUS BYTE
1640                    ;*ACTUAL = NNNN
1641                    ;*EXPECTED = NNNN
1642                    ;*
1643                    ;*XMC1 MICRO TEST 02
1644                    ;*XMC1 MICRO ERROR 13
1645                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1646                    ;*M8958, M8959, M8960, M8955'S
1647                    ;*FIRST GCR PARITY STATE NOT AS EXPECTED
1648                    ;*WHEN READ FROM ''WCSP'' BIT OF PORT 3 STATUS BYTE
1649                    ;*ACTUAL = NNNN
1650                    ;*EXPECTED = NNNN
1651                    ;*
1652                    ;*XMC1 MICRO TEST 02
1653                    ;*XMC1 MICRO ERROR 14
1654                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
```

```
1655                    ;*M8958, M8959, M8960, M8955'S
1656                    ;*XMC "XL WCLK" NOT SET 8 CLOCKS AFTER FIRST DATA STATE
1657                    ;*
1658                    ;*XMC1 MICRO TEST 02
1659                    ;*XMC1 MICRO ERROR 15
1660                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1661                    ;*M8958, M8959, M8960, M8955'S
1662                    ;*SECOND GCR DATA STATE NOT AS EXPECTED
1663                    ;*WHEN READ FROM TU0 STATUS BYTE
1664                    ;*ACTUAL = NNNN
1665                    ;*EXPECTED = NNNN
1666                    ;*
1667                    ;*XMC1 MICRO TEST 02
1668                    ;*XMC1 MICRO ERROR 16
1669                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1670                    ;*M8958, M8959, M8960, M8955'S
1671                    ;*SECOND GCR PARITY STATE NOT AS EXPECTED
1672                    ;*WHEN READ FROM "WCSP" BIT OF PORT 0 STATUS BYTE
1673                    ;*ACTUAL = NNNN
1674                    ;*EXPECTED = NNNN
1675                    ;*
1676                    ;*XMC1 MICRO TEST 02
1677                    ;*XMC1 MICRO ERROR 17
1678                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1679                    ;*M8958, M8959, M8960, M8955'S
1680                    ;*SECOND GCR DATA STATE NOT AS EXPECTED
1681                    ;*WHEN READ FROM TU1 STATUS BYTE
1682                    ;*ACTUAL = NNNN
1683                    ;*EXPECTED = NNNN
1684                    ;*
1685                    ;*XMC1 MICRO TEST 02
1686                    ;*XMC1 MICRO ERROR 20
1687                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1688                    ;*M8958, M8959, M8960, M8955'S
1689                    ;*SECOND GCR PARITY STATE NOT AS EXPECTED
1690                    ;*WHEN READ FROM "WCSP" BIT OF PORT 1 STATUS BYTE
1691                    ;*ACTUAL = NNNN
1692                    ;*EXPECTED = NNNN
1693                    ;*
1694                    ;*XMC1 MICRO TEST 02
1695                    ;*XMC1 MICRO ERROR 21
1696                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1697                    ;*M8958, M8959, M8960, M8955'S
1698                    ;*SECOND GCR DATA STATE NOT AS EXPECTED
1699                    ;*WHEN READ FROM TU2 STATUS BYTE
1700                    ;*ACTUAL = NNNN
1701                    ;*EXPECTED = NNNN
1702                    ;*
1703                    ;*XMC1 MICRO TEST 02
1704                    ;*XMC1 MICRO ERROR 22
1705                    ;*XMC1-XMC TRACK ENABLE TEST-GCR
1706                    ;*M8958, M8959, M8960, M8955'S
1707                    ;*SECOND GCR PARITY STATE NOT AS EXPECTED
1708                    ;*WHEN READ FROM "WCSP" BIT OF PORT 2 STATUS BYTE
```

```
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*XMC1 MICRO TEST 02
                                        ;*XMC1 MICRO ERROR 23
                                        ;*XMC1-XMC TRACK ENABLE TEST-GCR
                                        ;*M8958, M8959, M8960, M8955'S
                                        ;*SECOND GCR DATA STATE NOT AS EXPECTED
                                        ;*WHEN READ FROM TU3 STATUS BYTE
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*XMC1 MICRO TEST 02
                                        ;*XMC1 MICRO ERROR 24
                                        ;*XMC1-XMC TRACK ENABLE TEST-GCR
                                        ;*M8958, M8959, M8960, M8955'S
                                        ;*SECOND GCR PARITY STATE NOT AS EXPECTED
                                        ;*WHEN READ FROM "WCSP" BIT OF PORT 3 STATUS BYTE
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
434A                                    S
 (1)                                    ; ************************************************************************
434A                                    TEST2:  TESTX   @2              ;INITIALIZE THE TEST
 (1) 434A  3E  02          7.0                   MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1) 434C  CD  03  28     18.0                   CALL    TSET            ;SETUP THE TEST
                                        ,%XMC1-XMC TRACK ENABLE TEST-GCR
                                        ;&M8958, M8959, M8960, M8955'S
434F                                            REQ     @7,0,0,0,0
 (1) 434F  CD  06  28     18.0                   CALL    REQST
 (1) 4352  00                                    .BYTE   0               ;DATA PATTERN NUMBER
 (1) 4353  00  00                                .WORD   0               ;SYSTEM "0" COUNT
 (1) 4355  00  00                                .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1) 4357  00                                    .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1) 4358  07                                    .BYTE   @7              ;REQUEST CODE
4359                                            RIN     R12L
 (1) 4359  DB  94         10.0                   IN      R12L            ;READ R12L INTO AC
 (1) 435B  7F              4.0                   MOV     A,A             ;RETRY LINK
435C  32  76  47          13.0          STA     UNITMP
                                        
435F  A7                  4.0           ANA     A
4360  C2  71  43         10.0           JNZ     UNITO
4363                                    ERR     TSTEND,DUMMY
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1) 4363  CD  09  28     18.0                   CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 4366  02                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 4367  00                                    .BYTE
 (1) 4368  CD  15  28     18.0           DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1) 436B  DA  18  28     10.0                   JC      TSTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>OPERATOR ERROR - NO TM78 UNITS SELECTED
                                        ;<FATAL ERROR - TEST ABORTED
436E  C3  18  28          10.0          JMP     TSTEND
```

```
1745
1746                                    ;
1747                                    ;TAPE UNITS PRESENT ROUTINE - DATA PATH ENABLE AND LOOP WRITE TO READ
1748                                    ;WILL BE SET FOR ALL UNITS PRESENT.
1749                                    ;
1750
1751  4371  CD  4A  47      18.0  UNIT0:  CALL    CLEAR
1752  4374  3A  76  47      13.0          LDA     UNITMP          ;GET THE UNIT MAP BYTE
1753  4377  E6  01           7.0          ANI     $01             ;UNIT 0 PRESENT?
1754  4379  CA  86  43      10.0          JZ      UNIT1           ;NO-CONTINUE
1755  437C  DB  E0          10.0          IN      INTSTA          ;
1756  437E  E6  80           7.0          ANI     BIT7            ;
1757  4380  D3  E0          10.0          OUT     MBSEL           ;
1758  4382  3E  10           7.0          MVI     A,P.WPEN
1759  4384  D3  4C          10.0          OUT     PENAB           ;CLEAR TAPE CMD WORD
1760
1761  4386  3A  76  47      13.0  UNIT1:  LDA     UNITMP          ;GET THE UNIT MAP BYTE
1762  4389  E6  02           7.0          ANI     $02             ;UNIT 1 PRESENT?
1763  438B  CA  9A  43      10.0          JZ      UNIT2           ;NO-CONTINUE
1764  438E  DB  E0          10.0          IN      INTSTA          ;
1765  4390  E6  80           7.0          ANI     BIT7            ;
1766  4392  F6  01           7.0          ORI     @01             ;
1767  4394  D3  E0          10.0          OUT     MBSEL           ;
1768  4396  3E  10           7.0          MVI     A,P.WPEN
1769  4398  D3  4C          10.0          OUT     PENAB           ;CLEAR TAPE CMD WORD
1770
1771  439A  3A  76  47      13.0  UNIT2:  LDA     UNITMP          ;GET THE UNIT MAP BYTE
1772  439D  E6  04           7.0          ANI     $04             ;UNIT 2 PRESENT?
1773  439F  CA  AE  43      10.0          JZ      UNIT3           ;NO-CONTINUE
1774  43A2  DB  E0          10.0          IN      INTSTA          ;
1775  43A4  E6  80           7.0          ANI     BIT7            ;
1776  43A6  F6  02           7.0          ORI     @02             ;
1777  43A8  D3  E0          10.0          OUT     MBSEL           ;
1778  43AA  3E  10           7.0          MVI     A,P.WPEN
1779  43AC  D3  4C          10.0          OUT     PENAB
1780
1781  43AE  3A  76  47      13.0  UNIT3:  LDA     UNITMP          ;GET THE UNIT MAP BYTE
1782  43B1  E6  08           7.0          ANI     $08             ;UNIT 3 PRESENT?
1783  43B3  CA  C2  43      10.0          JZ      GO              ;NO-START TEST
1784  43B6  DB  E0          10.0          IN      INTSTA          ;
1785  43B8  E6  80           7.0          ANI     BIT7            ;
1786  43BA  F6  03           7.0          ORI     @03             ;
1787  43BC  D3  E0          10.0          OUT     MBSEL           ;
1788  43BE  3E  10           7.0          MVI     A,P.WPEN
1789  43C0  D3  4C          10.0          OUT     PENAB
1790
1791
1792                                    ;
1793                                    ;ALL AVAILABLE PORTS ARE SET UP - START THE TEST SEQUENCE
1794                                    ;
1795
1796  43C2  3E  40           7.0  GO:     MVI     A,P.LCS         ;KILL MTA ON THIS PORT
1797  43C4  D3  48          10.0          OUT     PDIAG
1798  43C6  AF               4.0          XRA     A
```

```
1799  43C7  D3  40        10.0          OUT   TCMD              ;KILL MTA LINK
1800  43C9  3E  01         7.0          MVI   A,$1              ;LOAD A 1
1801  43CB  32  7A  47    13.0          STA   EGCRP             ;LOAD THE EXPECTED PARITY STATE
1802  43CE  32  78  47    13.0          STA   TEPMSK            ;LOAD THE INITIAL TRACK ENABLE MASK
1803  43D1  32  75  47    13.0          STA   PASSCNT           ;LOAD UP THE PASS COUNT
1804  43D4  3E  FF         7.0          MVI   A,$FF
1805                                                           ;SET THE DATA TRACK ENABLE
1806  43D6  32  77  47    13.0          STA   TEDMSK            ;MASK TO ALL ONES
1807  43D9  32  7B  47    13.0          STA   EGCRD             ;SET THE EXPECTED DATA STATE
1808  43DC  3E  01         7.0          MVI   A,W.RST           ;RESTART THE WMC/XMC
1809  43DE  D3  D3        10.0          OUT   WMCCTL            ;LOAD THE CONTROL
1810  43E0  3E  05         7.0          MVI   A,SSTEP           ;SELECT SINGLE STEP WRITE CLOCK
1811  43E2  D3  F0        10.0          OUT   CLKCTL            ;INITIALIZE CLOCK MODULE
1812  43E4  32  7C  47    13.0          STA   CCTLWD            ;SAVE SOFTWARE IMAGE
1813  43E7  AF             4.0          XRA   A
1814  43E8  D3  D3        10.0          OUT   WMCCTL            ;FINISH THE RESTART CYCLE
1815  43EA  AF             4.0          XRA   A                 ;CLEAR THE ACCUMULATOR
1816  43EB  D3  D2        10.0          OUT   TRKENA            ;CLEAR THE TRACK ENABLE REGISTER
1817  43ED  D3  D2        10.0          OUT   TRKENA
1818
1819  43EF  3E  78         7.0          MVI   A,X.ENAB!W.ONES.W.GCR!W.WRITE
1820  43F1  D3  D3        10.0          OUT   WMCCTL            ;START THE XMC
1821
1822  43F3  CD  35  47    18.0  ENLP0:  CALL  ADJDAT           ;ADJUST THE EXPECTED DATA
1823  43F6  3A  75  47    13.0          LDA   PASSCNT           ;GET THE PASS COUNT
1824  43F9                              ROUT  R05L              ;WRITE TO CAS
 (1)  43F9  D3  8A        10.0                OUT     R05L              ;WRITE AC INTO R05L
 (1)  43FB  7F             4.0                MOV     A,A               ;RETRY LINK
1825  43FC  AF             4.0          XRA   A                 ;CLEAR A
1826  43FD                              ROUT  R05H              ;WRITE TO CAS
 (1)  43FD  D3  8B        10.0                OUT     R05H              ;WRITE AC INTO R05H
 (1)  43FF  7F             4.0                MOV     A,A               ;RETRY LINK
1827  4400  3A  78  47    13.0          LDA   TEPMSK            ;GET THE TRACK ENABLE PARITY BIT
1828  4403  D3  D2        10.0          OUT   TRKENA            ;LOAD TO THE TRACK ENABLE BYTE
1829  4405  3A  77  47    13.0          LDA   TEDMSK            ;GET THE TRACK ENABLE DATA BITS
1830  4408  D3  D2        10.0          OUT   TRKENA            ;LOAD TO THE TRACK ENABLE BYTE
1831
1832  440A  3E  7F         7.0          MVI   A,@177            ;LOAD A TIMEOUT COUNT
1833  440C  32  79  47    13.0          STA   TOUT
1834
1835  440F  CD  18  47    18.0  1$:     CALL  CLKSYS            ;SINGLE STEP THE XMC
1836
1837  4412  3A  79  47    13.0          LDA   TOUT              ;GET THE TIMEOUT COUNT
1838  4415  3D             4.0          DCR   A                 ;DECREMENT COUNT
1839  4416  32  79  47    13.0          STA   TOUT              ;STORE THE TIMEOUT COUNT
1840  4419  DB  D0        10.0          IN    WMCSTA            ;GET THE WMC STATUS BYTE
1841  441B  E6  01         7.0          ANI   X.WCLK            ;XL WCLK SET?
1842  441D  C2  35  44    10.0          JNZ   TUNITO            ;YES TO CHECK DATA
1843  4420  3A  79  47    13.0          LDA   TOUT              ;GET THE TIMEOUT COUNT
1844  4423  A7             4.0          ANA   A                 ;SET CONDITION BITS
1845  4424  C2  0F  44    10.0          JNZ   1$
```

```
1847  4427                                    ERR     UNIT0,ENCN0
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4427  CD  09  28      18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0003                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  442A  03                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  442B  00                                          .BYTE
 (1)  442C  CD  15  28      18.0                ENCN0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  442F  DA  71  43      10.0                        JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
1848                                    ;>XMC TIMEOUT WHILE WAITING FOR "XL WCLK" TO SET
1849                                    ;>FATAL ERROR - TEST ABORTED
1850
1851  4432  C3  18  28      10.0                JMP     TSTEND
1852                                    ;
1853                                    ;TAPE UNIT PORT 0 TEST - DATA PHASE 0
1854                                    ;
1855  4435  CD  18  47      18.0        TUNIT0: CALL    CLKSYS          ;ISSUE ONE MORE CLOCK TO THE XMC
1856  4438  CD  18  47      18.0                CALL    CLKSYS
1857  443B  21  7B  47      10.0                LXI     H,EGCRD
1858  443E  3A  76  47      13.0                LDA     UNITMP          ;GET THE UNIT MAP BYTE
1859  4441  E6  01            7.0               ANI     $01             ;UNIT 0 PRESENT?
1860  4443  CA  89  44      10.0                JZ      TUNIT1          ;NO-CONTINUE
1861  4446  DB  E0            10.0              IN      INTSTA          ;
1862  4448  E6  80            7.0               ANI     BIT7            ;
1863  444A  D3  E0            10.0              OUT     MBSEL           ;
1864  444C  DB  40            10.0              IN      TSTS            ;YES-GET THE TU STATUS
1865  444E  32  7D  47      13.0                STA     TEMPD
1866  4451  BE                7.0               CMP     M               ;COMPARE WITH EXPECTED DATA
1867  4452  CA  61  44      10.0                JZ      TUOCN0          ;CONTINUE IF EQUAL
1868  4455                                    ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  4455  D3  94            10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4457  7F                4.0                     MOV     A,A             ;RETRY LINK
1869  4458  7E                7.0               MOV     A,M             ;GET EXPECTED DATA
1870  4459                                    ROUT    EDATA           ;STORE EXPECTED DATA
 (1)  4459  D3  95            10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  445B  7F                4.0                     MOV     A,A             ;RETRY LINK
1871  445C                                    ERRB    UNIT0,TUOCN0
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  445C  CD  12  28      18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0004                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  445F  04                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4460  00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4461  CD  15  28      18.0                TUOCN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4464  DA  71  43      10.0                        JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
1872                                    ;>FIRST GCR DATA STATE NOT AS EXPECTED
1873                                    ;>WHEN READ FROM TUO STATUS BYTE
1874
1875  4467  21  7A  47      10.0                LXI     H,EGCRP
1876  446A  DB  48            10.0              IN      PSTAT
1877  446C  0F                4.0               RRC
1878  446D  0F                4.0               RRC
1879  446E  E6  01            7.0               ANI     $01
1880  4470  32  7E  47      13.0                STA     TEMPP
1881  4473  BE                7.0               CMP     M
1882  4474  CA  83  44      10.0                JZ      TUOCN1
```

```
1883  4477                                    ROUT    ADATA
 (1)  4477  D3  94          10.0                      OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4479  7F               4.0                      MOV     A,A             ;RETRY LINK
1884  447A  7E               7.0              MOV     A,M
1885  447B                                    ROUT    EDATA
 (1)  447B  D3  95          10.0                      OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  447D  7F               4.0                      MOV     A,A             ;RETRY LINK
1886  447E                                    ERRB    UNIT0,TU0CN1
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  447E  CD  12  28      18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0005                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4481  05                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4482  00                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4483  CD  15  28      18.0              TU0CN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4486  DA  71  43      10.0                      JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
1887                                    ;>FIRST GCR PARITY STATE NOT AS EXPECTED
1888                                    ;>WHEN READ FROM "WCSP" BIT OF PORT 0 STATUS BYTE
1889
1890  4489                              S
 (1)                                    ; ***************************************************************************
1891                                    ;TAPE UNIT PORT 1 TEST - DATA PHASE 0
1892  4489                              S
 (1)                                    ; ***************************************************************************
1893
1894  4489  21  7B  47          10.0    TUNIT1: LXI     H,EGCRD
1895  448C  3A  76  47          13.0            LDA     UNITMP          ;GET THE UNIT MAP BYTE
1896  448F  E6  02               7.0            ANI     $02             ;UNIT 1 PRESENT?
1897  4491  CA  D9  44          10.0            JZ      TUNIT2          ;NO-CONTINUE
1898  4494  DB  E0              10.0            IN      INTSTA          ;
1899  4496  E6  80               7.0            ANI     BIT7            ;
1900  4498  F6  01               7.0            ORI     @01             ;
1901  449A  D3  E0              10.0            OUT     MBSEL           ;
1902  449C  DB  40              10.0            IN      TSTS            ;YES-GET THE TU STATUS
1903  449E  32  7D  47          13.0            STA     TEMPD
1904  44A1  BE                   7.0            CMP     M               ;COMPARE WITH EXPECTED DATA
1905  44A2  CA  B1  44          10.0            JZ      TU1CN0          ;CONTINUE IF EQUAL
1906  44A5                                    ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  44A5  D3  94          10.0                      OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44A7  7F               4.0                      MOV     A,A             ;RETRY LINK
1907  44A8  7E               7.0              MOV     A,M             ;GET EXPECTED DATA
1908  44A9                                    ROUT    EDATA           ;STORE EXPECTED DATA
 (1)  44A9  D3  95          10.0                      OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44AB  7F               4.0                      MOV     A,A             ;RETRY LINK
1909  44AC                                    ERRB    UNIT0,TU1CN0
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44AC  CD  12  28      18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0006                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44AF  06                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44B0  00                                      .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  44B1  CD  15  28      18.0              TU1CN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44B4  DA  71  43      10.0                      JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
1910                                    ;>FIRST GCR DATA STATE NOT AS EXPECTED
1911                                    ;>WHEN READ FROM TU1 STATUS BYTE
1912
```

```
 1913  44B7  21  7A  47   10.0          LXI     H,EGCRP
 1914  44BA  DB  48       10.0          IN      PSTAT
 1915  44BC  0F           4.0           RRC
 1916  44BD  0F           4.0           RRC
 1917  44BE  E6  01       7.0           ANI     $01
 1918  44C0  32  7E  47   13.0          STA     TEMPP
 1919  44C3  BE           7.0           CMP     M
 1920  44C4  CA  D3  44   10.0          JZ      TU1CN1
 1921  44C7                             ROUT    ADATA
  (1)  44C7  D3  94       10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  44C9  7F           4.0                   MOV     A,A             ;RETRY LINK
 1922  44CA  7E           7.0           MOV     A,M
 1923  44CB                             ROUT    EDATA
  (1)  44CB  D3  95       10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  44CD  7F           4.0                   MOV     A,A             ;RETRY LINK
 1924  44CE                             ERRB    UNIT0,TU1CN1
  (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  44CE  CD  12  28   18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)  0007                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  44D1  07                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  44D2  00                                 .BYTE                   ;PRINT ROUTINE NUMBER
  (1)  44D3  CD  15  28   18.0          TU1CN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  44D6  DA  71  43   10.0                  JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
 1925                          ;>FIRST GCR PARITY STATE NOT AS EXPECTED
 1926                          ;>WHEN READ FROM "WCSP" BIT OF PORT 1 STATUS BYTE
 1927
 1928  44D9                    S
  (1)                          ; ***************************************************************************
 1929                          ;TAPE UNIT PORT 2 TEST - DATA PHASE 0
 1930  44D9                    S
  (1)                          ; ***************************************************************************
 1931
 1932  44D9  21  7B  47   10.0  TUNIT2: LXI     H,FGCRD
 1933  44DC  3A  76  47   13.0          LDA     UNITMP          ;GET THE UNIT MAP BYTE
 1934  44DF  E6  04       7.0           ANI     $04             ;UNIT 2 PRESENT?
 1935  44E1  CA  29  45   10.0          JZ      TUNIT3          ;NO-CONTINUE
 1936  44E4  DB  E0       10.0          IN      INTSTA          ;
 1937  44E6  E6  80       7.0           ANI     BIT7            ;
 1938  44E8  F6  02       7.0           ORI     @02             ;
 1939  44EA  D3  E0       10.0          OUT     MBSEL           ;
 1940  44EC  DB  40       10.0          IN      TSTS            ;YES-GET THE TU STATUS
 1941  44EE  32  7D  47   13.0          STA     TEMPD
 1942  44F1  BE           7.0           CMP     M               ;COMPARE WITH EXPECTED DATA
 1943  44F2  CA  01  45   10.0          JZ      TU2CN0          ;CONTINUE IF EQUAL
 1944  44F5                             ROUT    ADATA           ;STORE ACTUAL DATA
  (1)  44F5  D3  94       10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  44F7  7F           4.0                   MOV     A,A             ;RETRY LINK
 1945  44F8  7E           7.0           MOV     A,M             ;GET EXPECTED DATA
 1946  44F9                             ROUT    EDATA           ;STORE EXPECTED DATA
  (1)  44F9  D3  95       10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  44FB  7F           4.0                   MOV     A,A             ;RETRY LINK
```

```
                                        ERRB    UNIT0,TU2CN0
                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44FC  CD  12  28      18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0008                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44FF  08                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4500  00                                  .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4501  CD  15  28      18.0        TU2CN0::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4504  DA  71  43      10.0                JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>FIRST GCR DATA STATE NOT AS EXPECTED
                                ;>WHEN READ FROM TU2 STATUS BYTE

 1952  4507  21  7A  47      10.0        LXI     H,EGCRP
 1953  450A  DB  48          10.0        IN      PSTAT
 1954  450C  0F               4.0        RRC
 1955  450D  0F               4.0        RRC
 1956  450E  E6  01           7.0        ANI     $01
 1957  4510  32  7E  47      13.0        STA     TEMPP
 1958  4513  BE               7.0        CMP     M
 1959  4514  CA  23  45      10.0        JZ      TU2CN1
 1960  4517                               ROUT    ADATA
 (1)  4517  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4519  7F               4.0                MOV     A,A             ;RETRY LINK
 1961  451A  7E               7.0        MOV     A,M
 1962  451B                               ROUT    EDATA
 (1)  451B  D3  95          10.0                OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  451D  7F               4.0                MOV     A,A             ;RETRY LINK
 1963  451E                               ERRB    UNIT0,TU2CN1
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  451E  CD  12  28      18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0009                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4521  09                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4522  00                                  .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4523  CD  15  28      18.0        TU2CN1::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4526  DA  71  43      10.0                JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
                                ;>FIRST GCR PARITY STATE NOT AS EXPECTED
                                ;>WHEN READ FROM "WCSP" BIT OF PORT 2 STATUS BYTE

 1967  4529                               S
 (1)                                    ; ***************************************************************************
                                ;TAPE UNIT PORT 3 TEST - DATA PHASE 0
 1969  4529                               S
 (1)                                    ; ***************************************************************************

 1971  4529  21  7B  47      10.0  TUNIT3: LXI     H,EGCRD
 1972  452C  3A  76  47      13.0        LDA     UNITMP          ;GET THE UNIT MAP BYTE
 1973  452F  E6  08           7.0        ANI     $08             ;UNIT 3 PRESENT?
 1974  4531  CA  79  45      10.0        JZ      PHASE2          ;NO-CONTINUE
 1975  4534  DB  E0          10.0        IN      INTSTA
 1976  4536  E6  80           7.0        ANI     BIT7
 1977  4538  F6  03           7.0        ORI     @03
 1978  453A  D3  E0          10.0        OUT     MBSEL
 1979  453C  DB  40          10.0        IN      TSTS            ;YES-GET THE TU STATUS
 1980  453E  32  7D  47      13.0        STA     TEMPD
 1981  4541  BE               7.0        CMP     M               ;COMPARE WITH EXPECTED DATA
```

```
1982  4542  CA  51  45   10.0          JZ      TU3CN0          ;CONTINUE IF EQUAL
1983  4545                             POUT    ADATA           ;STORE ACTUAL DATA
 (1)  4545  D3  94        10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4547  7F             4.0                 MOV     A,A             ;RETRY LINK
1984  4548  7E             7.0          MOV     A,M             ;GET EXPECTED DATA
1985  4549                             ROUT    EDATA           ;STORE EXPECTED DATA
 (1)  4549  D3  95        10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  454B  7F             4.0                 MOV     A,A             ;RETRY LINK
1986  454C                             ERRB    UNITO,TU3CN0
 (1)                                 ,FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  454C  CD  12  28    18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      000A                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  454F  0A                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4550  00                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4551  CD  15  28    18.0          TU3CN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4554  DA  71  43    10.0                 JC      UNITO           ;LOOP ADDRESS IF LOOP SPECIFIED
1987                         ;>FIRST GCR DATA STATE NOT AS EXPECTED
1988                         ;>WHEN READ FROM TU3 STATUS BYTE
1989  4557  21  7A  47    10.0          LXI     H,EGCRP
1990  455A  DB  48         10.0          IN      PSTAT
1991  455C  0F              4.0          RRC
1992  455D  0F              4.0          RRC
1993  455E  E6  01          7.0          ANI     $01
1994  4560  32  7E  47     13.0          STA     TEMPP
1995  4563  BE              7.0          CMP     M
1996  4564  CA  73  45     10.0          JZ      TU3CN1
1997  4567                             ROUT    ADATA
 (1)  4567  D3  94        10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4569  7F             4.0                 MOV     A,A             ;RETRY LINK
1998  456A  7E             7.0          MOV     A,M
1999  456B                             ROUT    EDATA
 (1)  456B  D3  95        10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  456D  7F             4.0                 MOV     A,A             ;RETRY LINK
2000  456E                             ERRB    UNITO,TU3CN1
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  456E  CD  12  28    18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      000B                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4571  0B                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4572  00                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4573  CD  15  28    18.0          TU3CN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4576  DA  71  43    10.0                 JC      UNITO           ;LOOP ADDRESS IF LOOP SPECIFIED
2001                         ;>FIRST GCR PARITY STATE NOT AS EXPECTED
2002                         ;>WHEN READ FROM "WCSP" BIT OF PORT 3 STATUS BYTE
2003
2004  4579  CD  24  47    18.0  PHASE2: CALL    COMP            ;COMPLEMENT THE EXPECTED DATA
2005  457C  CD  35  47    18.0          CALL    ADJDAT          ;ADJUST THE EXPECTED DATA
2006  457F  06  0E          7.0  2$:     MVI     B,14            ;SET UP THE CLOCK COUNT
2007  4581  CD  18  47     18.0  3$:     CALL    CLKSYS          ;CLOCK THE TRANSLATOR
2008  4584  05              4.0          DCR     B               ;DECREMENT THE CLOCK COUNT
2009  4585  C2  81  45     10.0          JNZ     3$              ;CONTINUE-TILL ZERO
2010  4588  DB  D0         10.0          IN      WMCSTA          ;GET THE WMC STATUS
2011  458A  E6  01          7.0          ANI     $01
2012  458C  C2  94  45     10.0          JNZ     PH2CN
```

```
2014  458F                                        ERR     UNIT0,PH2CN
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  458F  CD  09  28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000C                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4592  0C                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4593  00                                          .BYTE
 (1)  4594  CD  15  28        18.0               PH2CN:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4597  DA  71  43        10.0                       JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
2015                                    ;>XMC "XL WCLK" NOT SET 8 CLOCKS AFTER FIRST DATA STATE
2016
2017
2018                                    ;
2019                                    ;TAPE UNIT PORT 0 TEST - DATA PHASE 1
2020                                    ;
2021
2022  459A  CD  18  47        18.0      EUNIT0: CALL    CLKSYS          ;ISSUE ONE MORE CLOCK TO THE XMC
2023  459D  CD  18  47        18.0              CALL    CLKSYS
2024  45A0  21  7B  47        10.0              LXI     H,EGCRD
2025  45A3  3A  76  47        13.0              LDA     UNITMP          ;GET THE UNIT MAP BYTE
2026  45A6  E6  01             7.0              ANI     $01             ;UNIT 0 PRESENT?
2027  45A8  CA  EE  45        10.0              JZ      EUNIT1          ;NO-CONTINUE
2028  45AB  DB  E0            10.0              IN      INTSTA
2029  45AD  E6  80             7.0              ANI     BIT7
2030  45AF  D3  E0            10.0              OUT     MBSEL
2031  45B1  DB  40            10.0              IN      TSTS            ;YES-GET THE TU STATUS
2032  45B3  32  7D  47        13.0              STA     TEMPD
2033  45B6  BE                 7.0              CMP     M               ;COMPARE WITH EXPECTED DATA
2034  45B7  CA  C6  45        10.0              JZ      EU0CN0          ;CONTINUE IF EQUAL
2035  45BA                                    ROUT    ADATA           ;STORE ACEUAL DATA
 (1)  45BA  D3  94            10.0                      OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  45BC  7F                 4.0                      MOV     A,A             ;RETRY LINK
2036  45BD  7E                 7.0              MOV     A,M             ;GET EXPECTED DATA
2037  45BE                                    ROUT    EDATA           ;STORE EXPECTED DATA
 (1)  45BE  D3  95            10.0                      OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  45C0  7F                 4.0                      MOV     A,A             ;RETRY LINK
2038  45C1                                    ERRB    UNIT0,EU0CN0
 (1)                                            ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  45C1  CD  12  28        18.0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000D                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45C4  0D                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45C5  00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45C6  CD  15  28        18.0               EU0CN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45C9  DA  71  43        10.0                       JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
2039                                    ;>SECOND GCR DATA STATE NOT AS EXPECTED
2040                                    ;>WHEN READ FROM TU0 STATUS BYTE
2041
2042
2043  45CC  21  7A  47        10.0              LXI     H,EGCRP
2044  45CF  DB  48            10.0              IN      PSTAT
2045  45D1  0F                 4.0              RRC
2046  45D2  0F                 4.0              RRC
2047  45D3  E6  01             7.0              ANI     $01
2048  45D5  32  7E  47        13.0              STA     TEMPP
2049  45D8  BE                 7.0              CMP     M
```

```
2050  45D9  CA  E8  45  10.0          JZ      EUOCN1
2051  45DC                           ROUT    ADATA
 (1)  45DC  D3  94      10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  45DE  7F          4.0                   MOV     A,A             ;RETRY LINK
2052  45DF  7E          7.0           MOV     A,M
2053  45E0                           ROUT    EDATA
 (1)  45E0  D3  95      10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  45E2  7F          4.0                   MOV     A,A             ;RETRY LINK
2054  45E3                           ERRB    UNIT0,EUOCN1
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  45E3  CD  12  28  18.0                   CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      000E                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45E6  0E                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45E7  00                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  45E8  CD  15  28  18.0           EUOCN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45EB  DA  71  43  10.0                   JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
2055                           ;>SECOND GCR PARITY STATE NOT AS EXPECTED
2056                           ;>WHEN READ FROM "WCSP" BIT OF PORT 0 STATUS BYTE
2057
2058  45EE                     S
 (1)                           ; *****************************************************************************
2059                           ;TAPE UNIT PORT 1 TEST - DATA PHASE ONE
2060  45EE                     S
 (1)                           ; *****************************************************************************
2061
2062  45EE  21  7B  47  10.0   EUNIT1: LXI     H,EGCRD
2063  45F1  3A  76  47  13.0           LDA     UNITMP          ;GET THE UNIT MAP BYTE
2064  45F4  E6  02       7.0          ANI     $02             ;UNIT 1 PRESENT?
2065  45F6  CA  3E  46  10.0           JZ      EUNIT2          ;NO-CONTINUE
2066  45F9  DB  E0      10.0          IN      INTSTA
2067  45FB  E6  80       7.0          ANI     BIT7
2068  45FD  F6  01       7.0          ORI     @01
2069  45FF  D3  E0      10.0          OUT     MBSEL
2070  4601  DB  40      10.0          IN      TSTS            ;YES-GET THE TU STAEUS
2071  4603  32  7D  47  13.0          STA     TEMPD
2072  4606  BE          7.0          CMP     M               ;COMPARE WITH EXPECTED DATA
2073  4607  CA  16  46  10.0          JZ      EU1CN0          ;CONTINUE IF EQUAL
2074  460A                           ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  460A  D3  94      10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  460C  7F          4.0                   MOV     A,A             ;RETRY LINK
2075  460D  7E          7.0           MOV     A,M             ;GET EXPECTED DATA
2076  460E                           ROUT    EDATA           ;STORE EXPECTED DATA
 (1)  460E  D3  95      10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4610  7F          4.0                   MOV     A,A             ;RETRY LINK
2077  4611                           ERRB    UNIT0,EU1CN0
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4611  CD  12  28  18.0                   CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      000F                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4614  0F                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4615  00                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4616  CD  15  28  18.0           EU1CN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4619  DA  71  43  10.0                   JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
2078                           ;>SECOND GCR DATA STATE NOT AS EXPECTED
2079                           ;>WHEN READ FROM TU1 STATUS BYTE
```

```
2080
2081  461C  21  7A  47  10.0          LXI     H,EGCRP
2082  461F  DB  48      10.0          IN      PSTAT
2083  4621  0F          4.0          RRC
2084  4622  0F          4.0          RRC
2085  4623  E6  01      7.0          ANI     $01
2086  4625  32  7E  47  13.0          STA     TEMPP
2087  4628  BE          7.0          CMP     M
2088  4629  CA  38  46  10.0          JZ      EU1CN1
2089  462C                           ROUT    ADATA
 (1)  462C  D3  94      10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  462E  7F          4.0                   MOV     A,A             ;RETRY LINK
2090  462F  7E          7.0          MOV     A,M
2091  4630                           ROUT    EDATA
 (1)  4630  D3  95      10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4632  7F          4.0                   MOV     A,A             ;RETRY LINK
2092  4633                           ERRB    UNIT0,EU1CN1
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4633  CD  12  28  18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0010                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4636  10                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4637  00                                .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4638  CD  15  28  18.0          EU1CN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  463B  D.  71  43  10.0                  JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
2093                           ·>SECOND GCR PARITY STATE NOT AS EXPECTED
2094                           ;>WHEN READ FROM "WCSP" BIT OF PORT 1 STATUS BYTE
2095
2096  463E                     S
 (1)                           ; ***************************************************************************
2097                           ;TAPE UNIT PORT 2 TEST - DATA PHASE ONE
2098  463E                     S
 (1)                           ; ***************************************************************************
2099
2100  463E  21  7B  47  10.0  EUNIT2: LXI     H,EGCRD
2101  4641  3A  76  47  13.0          LDA     UNITMP          ;GET THE UNIT MAP BYTE
2102  4644  E6  04      7.0          ANI     $04             ;UNIT 2 PRESENT?
2103  4646  CA  8E  46  10.0          JZ      EUNIT3          ;NO-CONTINUE
2104  4649  DB  E0      10.0          IN      INTSTA
2105  464B  E6  80      7.0          ANI     BIT7
2106  464D  F6  02      7.0          ORI     @02
2107  464F  D3  E0      10.0          OUT     MBSEL
2108  4651  DB  40      10.0          IN      TSTS            ;YES-GET THE TU STAEUS
2109  4653  32  7D  47  13.0          STA     TEMPD
2110  4656  BE          7.0          CMP     M               ;COMPARE WITH EXPECTED DATA
2111  4657  CA  66  46  10.0          JZ      EU2CN0          ;CONTINUE IF EQUAL
2112  465A                           ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  465A  D3  94      10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  465C  7F          4.0                   MOV     A,A             ;RETRY LINK
2113  465D  7E          7.0          MOV     A,M             ;GET EXPECTED DATA
2114  465E                           ROUT    EDATA           ;STORE EXPECTED DATA
 (1)  465E  D3  95      10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4660  7F          4.0                   MOV     A,A             ;RETRY LINK
```

```
2116  4661                                          ERRB    UNIT0,EU2CN0
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4661   CD   12   28        18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0011                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4664   11                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4665   00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4666   CD   15   28        18.0               EU2CN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4669   DA   71   43        10.0                      JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
2117                                          ;>SECOND GCR DATA STATE NOT AS EXPECTED
2118                                          ;>WHEN READ FROM TU2 STATUS BYTE
2119
2120  466C   21   7A   47        10.0          LXI     H,EGCRP
2121  466F   DB   48             10.0          IN      PSTAT
2122  4671   0F                   4.0          RRC
2123  4672   0F                   4.0          RRC
2124  4673   E6   01              7.0          ANI     $01
2125  4675   32   7E   47        13.0          STA     TEMPP
2126  4678   BE                   7.0          CMP     M
2127  4679   CA   88   46        10.0          JZ      EU2CN1
2128  467C                                     ROUT    ADATA
 (1)  467C   D3   94             10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  467E   7F                   4.0                  MOV     A,A             ;RETRY LINK
2129  467F   7E                   7.0          MOV     A,M
2130  4680                                     ROUT    EDATA
 (1)  4680   D3   95             10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4682   7F                   4.0                  MOV     A,A             ;RETRY LINK
2131  4683                                     ERRB    UNIT0,EU2CN1
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4683   CD   12   28        18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0012                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4686   12                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4687   00                                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4688   CD   15   28        18.0               EU2CN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  468B   DA   71   43        10.0                      JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
2132                                          ;>SECOND GCR PARITY STATE NOT AS EXPECTED
2133                                          ;>WHEN READ FROM "WCSP" BIT OF PORT 2 STATUS BYTE
2134
2135  468E                                    S
 (1)                                          ; ***************************************************************************
2136                                          ;TAPE UNIT PORT 3 TEST - DATA PHASE ONE
2137  468E                                    S
 (1)                                          ; ***************************************************************************
2138
2139  468E   21   7B   47        10.0  EUNIT3: LXI     H,EGCRD
2140  4691   3A   76   47        13.0          LDA     UNITMP          ;GET THE UNIT MAP BYTE
2141  4694   E6   08             7.0          ANI     $08             ;UNIT 3 PRESENT?
2142  4696   CA   DE   46        10.0          JZ      PHASE3          ;NO-CONTINUE
2143  4699   DB   E0             10.0          IN      INTSTA
2144  469B   E6   80             7.0          ANI     BIT7
2145  469D   F6   03             7.0          ORI     @03
2146  469F   D3   E0             10.0          OUT     MBSEL
2147  46A1   DB   40             10.0          IN      TSTS            ;YES-GET THE TU STAEUS
2148  46A3   32   7D   47        13.0          STA     TEMPD
2149  46A6   BE                   7.0          CMP     M               ;COMPARE WITH EXPECTED DATA
```

```
2150  46A7  CA  B6  46      10.0          JZ      EU3CN0          ;CONTINUE IF EQUAL
2151  46AA                                ROUT    ADATA           ;STORE ACTUAL DATA
 (1)  46AA  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46AC  7F               4.0                  MOV     A,A             ;RETRY LINK
2152  46AD  7E               7.0          MOV     A,M             ;GET EXPECTED DATA
2153  46AE                                ROUT    EDATA           ;STORE EXPECTED DATA
 (1)  46AE  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  46B0  7F               4.0                  MOV     A,A             ;RETRY LINK
2154  46B1                                ERRB    UNIT0,EU3CN0
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  46B1  CD  12  28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0013                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46B4  13                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46B5  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  46B6  CD  15  28      18.0          EU3CN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46B9  DA  71  43      10.0                  JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
2155                            ;>SECOND GCR DATA STATE NOT AS EXPECTED
2156                            ;>WHEN READ FROM TU3 STATUS BYTE
2157
2158  46BC  21  7A  47      10.0          LXI     H,EGCRP
2159  46BF  DB  48          10.0          IN      PSTAT
2160  46C1  0F               4.0          RRC
2161  46C2  0F               4.0          RRC
2162  46C3  E6  01           7.0          ANI     $01
2163  46C5  32  7E  47      13.0          STA     TEMPP
2164  46C8  BE               7.0          CMP     M
2165  46C9  CA  D8  46      10.0          JZ      EU3CN1
2166  46CC                                ROUT    ADATA
 (1)  46CC  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46CE  7F               4.0                  MOV     A,A             ;RETRY LINK
2167  46CF  7E               7.0          MOV     A,M
2168  46D0                                ROUT    EDATA
 (1)  46D0  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  46D2  7F               4.0                  MOV     A,A             ;RETRY LINK
2169  46D3                                ERRB    UNIT0,EU3CN1
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  46D3  CD  12  28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0014                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46D6  14                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46D7  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  46D8  CD  15  28      18.0          EU3CN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46DB  DA  71  43      10.0                  JC      UNIT0           ;LOOP ADDRESS IF LOOP SPECIFIED
2170                            ;>SECOND GCR PARITY STATE NOT AS EXPECTED
2171                            ;>WHEN READ FROM ''WCSP'' BIT OF PORT 3 STATUS BYTE
2172
2173  46DE  CD  24  47      18.0  PHASE3: CALL    COMP            ;COMPLEMENT THE EXPECTED DATA
2174  46E1  3A  75  47      13.0          LDA     PASSCNT         ;GET THE PASS COUNTER
2175  46E4  3C               4.0          INR     A               ;UPDATE
2176  46E5  32  75  47      13.0          STA     PASSCNT         ;RESTORE IT
2177  46E8  3A  78  47      13.0          LDA     TEPMSK          ;GET THE TRACK ENABLE PARITY MASK
2178  46EB  47               4.0          MOV     B,A             ;BUFFER IT
2179  46EC  AF               4.0          XRA     A               ;CLEAR A
2180  46ED  32  78  47      13.0          STA     TEPMSK          ;CLEAR THE TRACK ENABLE PARITY MASK
2181  46F0  78               4.0          MOV     A,B             ;GET THE MASK BACK
```

```
2182  46F1  A7                    4.0          ANA     A               ;SET THE CONDITION CODE
2183  46F2  CA  F6  46            10.0         JZ      PH31            ;CONTINUE WITH C BIT CLEAR
2184  46F5  37                    4.0          STC                     ;ELSE-SET THE C BIT
2185  46F6  3A  77  47            13.0  PH31:  LDA     TEDMSK          ;GET THE TRACK ENABLE DATA MASK
2186  46F9  1F                    4.0          RAR                     .SHIFT 9 BIT MASK RIGHT
2187  46FA  32  77  47            13.0         STA     TEDMSK          .SAVE THE NEW TRACK ENABLE DATA MASK
2188  46FD  A7                    4.0          ANA     A               ;SET CONDITION BITS
2189  46FE  C2  F3  43            10.0         JNZ     ENLPO           ;CONTINUE UNTIL MASK = 0
2190
2191  4701                               EXIT:  ENDTST  UNITO
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4701                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4701  CD  06  28            18.0                        CALL    REQST
 (2)  4704  00                                                .BYTE                   ;DATA PATTERN NUMBER
 (2)  4705  00  00                                            .WORD                   ;SYSTEM "" COUNT
 (2)  4707  00  00                                            .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4709  00                                                .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  470A  07                                                .BYTE   7               ;REQUEST CODE
 (1)  470B  3A  9A  4F            13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1)  470E  3D                    4.0          DCR     A               ;DOWNCOUNT
 (1)  470F  32  9A  4F            13.0         STA     ITERA           ;SAVE COUNT
 (1)  4712  F2  71  43            10.0         JP      UNITO           ;DO TEST UNTIL TILL = 0
2192  4715  C3  18  28            10.0         JMP     TSTEND
2193
2194  4718                               $
 (1)                                   ; ***************************************************************************
2195                                   ;CLKSYS -- ROUTINE TO CLOCK THE SYSTEM 1 TIME BY WRITING CLOCK CONTROL
2196                                   ;       'CLK' BIT TO A '1' THEN A '0'.
2197  4718                             ;$
 (1)                                   ; ***************************************************************************
2198
2199  4718  3A  7C  47            13.0  CLKSYS: LDA     CCTLWD          ;GET THE SOFTWARE CLOCK CONTROL IMAGE
2200  471B  F6  40                 7.0          ORI     SSCLK           ;ADD IN 'CLK' BIT
2201  471D  D3  F0                 10.0         OUT     CLKCTL          ;LOAD CLOCK CONTROL
2202  471F  E6  BF                 7.0          ANI     @277            ;STRIP OFF CLOCK BIT
2203  4721  D3  F0                 10.0         OUT     CLKCTL          ;LOAD CLOCK CONTROL WORD
2204  4723  C9                    10.0         RET                     ;EXIT-CLOCK CYCLE
2205
2206
2207  4724  3A  7B  47            13.0  COMP:  LDA     EGCRD           ;COMPLEMENT THE DATA
2208  4727  2F                    4.0          CMA
2209  4728  32  7B  47            13.0         STA     EGCRD
2210  472B  3A  7A  47            13.0         LDA     EGCRP           ;COMPLEMENT THE PARITY
2211  472E  2F                    4.0          CMA
2212  472F  E6  01                 7.0          ANI     $01
2213  4731  32  7A  47            13.0         STA     EGCRP
2214  4734  C9                    10.0         RET
2215
2216  4735  21  78  47            10.0  ADJDAT: LXI     H,TEPMSK        ;GET THE ADDRESS OF THE PARITY TRACK ENABLE BIT
2217  4738  3A  7A  47            13.0         LDA     EGCRP           ;GET THE EXPECTED PARITY BIT
2218  473B  A6                    7.0          ANA     M               ;AND WITH THE TRACK ENABLE MASK
2219  473C  32  7A  47            13.0         STA     EGCRP           ;SAVE THE NEW EXPECTED PARITY BIT
2220  473F  21  77  47            10.0         LXI     H,TEDMSK        ;GET THE ADDRESS OF THE DATA TRACK ENABLE BITS
2221  4742  3A  7B  47            13.0         LDA     EGCRD           ;GET THE EXPECTER DATA BITS
```

```
2222  4745  A6                    7.0        ANA     M              ;AND WITH THE TRACK ENABLE MASK
2223  4746  32   7B   47          13.0        STA     EGCRD          ;SAVE THE NEW EXPECTED DATA BITS
2224  4749  C9                    10.0        RET
```

```
                                .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                SSUB
                                ; ************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*CLEAR ALL TU PORTS
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                ;*AND LOOP MODES.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                ;*    CLEAR PORT SELECT FOR TRANSPORT
                                ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                ;*    CLEAR PORT AMTIE WORD
                                ;*ENDSUB
                                S
                                ; ************************************************************************
474A F5              12.0 CLEAR:  PUSH   PSW             ;SAVE THE SELECTED PORT #
474B C5              12.0         PUSH   B
474C 06 00            7.0         MVI    B,0          ;START TO CLEAR AT PORT #0
474E DB E0           10.0 CLRLP:  IN     INTSTA       ;GET MB SELECT INFO
4750 E6 80            7.0         ANI    BIT7         ;SAVE ONLY THE MASSBUS SELECT BIT
4752 B0               4.0         ORA    B            ;ADD IN THE SELECTED PORT #
4753 D3 E0           10.0         OUT    MBSEL        ;RESET TO THIS PORT
4755 3E 80            7.0         MVI    A,@200       ;LOAD MTA REGISTER #0 SELECT CODE
4757 D3 40           10.0         OUT    TCMD         ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
4759 AF               4.0         XRA    A            ;CLEAR TU COMMAND A
475A D3 40           10.0         OUT    TCMD         ;
475C 3E 81            7.0         MVI    A,@201       ;LOAD MTA REGISTER #1 SELECT CODE
475E D3 40           10.0         OUT    TCMD         ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
4760 3E 00            7.0         MVI    A,SELCLR     ;LOAD TU "CLEAR SELECT" COMMAND
4762 D3 40           10.0         OUT    TCMD         ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
4764 AF               4.0         XRA    A
4765 D3 44           10.0         OUT    TAMT         ;CLEAR AMTIE WORD
4767 D3 48           10.0         OUT    PDIAG        ;CLEAR DIAG CONTROL WORD
4769 D3 4C           10.0         OUT    PENAB        ;CLEAR PORT ENABLE WORD
476B 04               4.0         INR    B            ;POINT TO THE NEXT PORT TO CLEAR
476C 78               4.0         MOV    A,B
476D FE 04            7.0         CPI    4            ;DONE?
476F C2 4E 47        10.0         JNZ    CLRLP        ;NO - CLEAR THIS PORT ALSO
4772 C1              10.0         POP    B            ;RESET B & C
4773 F1              10.0         POP    PSW          ;ALL DONE
4774 C9              10.0         RET                 ;EXIT
```

```
2270  4775   00                  PASSCNT: .BYTE 0          ;NUMBER OF PASSES THROUGH THE TEST
2271  4776   00                  UNITMP:  .BYTE   0        ;UNIT MAP
2272  4777   00                  TEDMSK:  .BYTE   0        ;TRACK ENABLE DATA BIT MASK
2273  4778   00                  TEPMSK:  .BYTE   0        ;TRACK ENABLE PARITY BIT MASK
2274  4779   00                  TOUT:    .BYTE   0        ;TIMEOUT COUNT
2275  477A   00                  EGCRP:   .BYTE   0        ;EXPECTED GCR PARITY STATE
2276  477B   00                  EGCRD:   .BYTE   0        ;EXPECTED GCR DATA STATE
2277  477C   00                  CCTLWD:  .BYTE   0        ;SOFTWARE CLOCK IMAGE
2278  477D   00                  TEMPD:   .BYTE   0        ;TEMP. DATA STORAGE
2279  477E   00                  TEMPP:   .BYTE   0        ;TEMP. PARITY STORAGE
2280        0000                           .END
```

```
A     =%0007          ADATA = 0094          ADJDAT  4735          AMTIEP= 0001
AMTIE7= 0002          ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91          B     =%0000          BADST = 0090          BIT0  = 0001
BIT1  = 0002          BIT15 = 8000          BIT2  = 0004          BIT3  = 0008
BIT4  = 0010          BIT5  = 0020          BIT6  = 0040          BIT7  = 0080
BIT8  = 0100          BIT9  = 0200          BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60          BRKXCT= 4F00          BSAVE   4F9C          BYTCNT= 00D4
BYTEH   4F24          BYTEL   4F23          C     =%0001          CASCT   4F21
CASCTL= 00A0          CASSTA= 00A0          CATTH = 0089          CATTL = 0088
CBUSST= 00A1          CBYTH = 008B          CBYTL = 008A          CCTLWD  477C
CDG1H = 0087          CDG1L = 0086          CDG2H = 0093          CDG2L = 0092
CDG3H = 0095          CDG3L = 0094          CDVTH = 008D          CDVTL = 008C
CHPTIE= 0028          CHOTIE= 0020          CH1TIE= 0021          CH2TIE= 0022
CH3TIE= 0023          CH4TIE= 0024          CH5TIE= 0025          CH6TIE= 0026
CH7TIE= 0027          CKLOP = 2815          CLEAR   474A          CLKCTL= 00F0
CLKSYS  4718          CLOCK   4F26          CLRLP   474E          CMC0H = 0099
CMC0L = 0098          CMC1H = 009B          CMC1L = 009A          CMC2H = 009D
CMC2L = 009C          CMC3H = 009F          CMC3L = 009E          CMINH = 0097
CMINL = 0096          CNTCTL= 00D7          COMP    4724          CRCWRD= 0018
CSAVE   4F9D          CSRLH = 0091          CSRLL = 0090          CTCH  = 0085
CTCL  = 0084          CTSTH = 008F          CTSTL = 008E          CXCTH = 0081
CXCTL = 0080          CXINH = 0083          CXINL = 0082          C.    = 0001
C.AVAI= 0080          C.DP  = 0008          C.DSE = 0010          C.DTU = 0003
C.DVA = 0008          C.FAIL= 00FC          C.FMT = 0070          C.FNCT= 003E
C.GO  = 0001          C.INTC= 00FE          C.MAIN= 0020          C.NSA = 0080
C.RCT = 00FC          C.SER = 0080          C.SHR = 0040          C.SKPC= 000F
C.TAPE= 0040          C.WCS = 0002          D     =%0002          DATACT= 00D0
DBUS    4F28          DBUSCT= 00C0          DBUSST= 00C0          DDRA  = 00D8
DDRAIN= 0010          DDRB  = 00D9          DDRBIN= 0002          DDRC  = 00DA
DDRCIN= 0001          DDRCO = 0088          DDRCTL= 00DB          DIAFLG  4F22
DIAGPG= 4300          DIAGRM= 4F90          DIARD = 000B          DONE1 = 0045
DONINT= 0010          DSAVE   4F9E          DSE   = 0006          DUMMY   4368  G
D.ATH0= 0001          D.ATH1= 0002          D.EOTD= 0010          D.LAGC= 0020
D.NOTW= 0040          D.NTHR= 0004          D.TACH= 0008          D.WR4 = 0080
E     =%0003          ECCBAD= 0042          ECCCOR= 0019          ECCOK = 0041
ECCSTA= 001A          ECCTST= 000E          EDATA = 0095          EGCRD   477B
EGCRP   477A          ENCNO   442C  G       ENLPO   43F3          EOTCLR= 0003
ERFLG   4F93          ERLP  = 2809          ERLPA = 280F          ERLPB = 2812
ERLPE = 280C          ERNUM   4F90          ERRCNT= 00D6          ESAVE   4F9F
EUNIT0  459A          EUNIT1  45EE          EUNIT2  463E          EUNIT3  468E
EU0CN0  45C6  G       EU0CN1  45E8  G       EU1CN0  4616  G       EU1CN1  4638  G
EU2CN0  4666  G       EU2CN1  4688  G       EU3CN0  46B6  G       EU3CN1  46D8  G
EXIT    4701          E.ACRC= 0010          E.AMT = 0020          E.CDP = 0080
E.CRC = 0080          E.PNTR= 0008          E.RPE = 0040          E.STEC= 0001
E.TTEC= 0002          E.UNC = 0004          FIFORD= 006A          FORMAT  4F25
FWDTST= 0061          GCRID = 0089          GCRSET= 0002          GO      43C2
GOODTM= 0092          H     =%0004          HLSAVE  4FA0          IE    = 0008
INTSTA= 00E0          ITEP*   4F9A          I.PWR = 0020          I.RMPE= 0040
I5.5  = 0010          I6    = 0020          I7.5  = 0040          KCALL = 005F
KCLR  = 007B          KI    = 003F          KENAB = 0078          KEXAM = 003E
KEYBRD= 00C8          KEY1  = 0078          KEY10 = 006D          KEY11 = 006E
KEY12 = 006F          KEY13 = 005C          KEY14 = 005D          KEY15 = 005E
KEY16 = 005F          KEY17 = 003C          KEY18 = 003D          KEY19 = 003E
KEY2  = 0079          KEY20 = 003F          KEY3  = 007A          KEY4  = 007B
```

```
KEY5  = 0074    KEY6  = 0075    KEY7  = 0076    KEY8  = 0077
KEY9  = 006C    KINTA = 006F    KLDAD = 003D    KN0   = 007C
KN1   = 005C    KN2   = 005D    KN3   = 005E    KN4   = 006C
KN5   = 006D    KN6   = 006E    KN7   = 0074    KN8   = 0075
KN9   = 0076    KU2   = 0079    KU3   = 007A    KU8   = 0077
L     =%0005    LBLANK= 000F    LCE   = 000B    LCH   = 000C
LCL   = 000D    LCP   = 000E    LC0   = 0000    LC1   = 0001
LC2   = 0002    LC3   = 0003    LC4   = 0004    LC5   = 0005
LC6   = 0006    LC7   = 0007    LC8   = 0008    LC9   = 0009
LDLEDA= 00CA    LDLEDB= 00CB    LDLEDC= 00CC    LDLEDD= 00CD
LDLEDE= 00CE    LDLEDF= 00CF    LKDIAG= 2800    LKKBD = 004C
LKKEY = 0049    LKLWMG= 0058    LKLWMP= 0055    LKLWPG= 0052
LKLWPP= 004F    LKMOD7= 0046    LKOPR = 0046    LPFLG   4F94
LPNUM   4F92    M     =%0006    MBSEL = 00E0    MB.A  = 0008
MB.B  = 0004    MEMTOP= 4FFF    MINUS = 000A    MM    = 8000
MSE   = 0008    MSGN  = 0014    MTACLR= 0000    MT.ARA= 0020
MT.CPE= 0080    MT.DSE= 0001    MT.FWD= 0040    MT.INH= 0008
MT.LWR= 0004    MT.MOT= 0002    MT.NWT= 0080    MT.PEC= 0040
MT.PSB= 0004    MT.PS0= 0001    MT.PS1= 0002    MT.REV= 0020
MT.WRT= 0010    MT.Z  = 0008    M.ATA = 0080    M.CAPE= 0020
M.CONT= 0080    M.DEM = 0020    M.EBL = 0010    M.EXC = 0008
M.FAIL= 0008    M.ILR = 0010    M.INIT= 0010    M.OCC = 0020
M.ONLI= 0001    M.PE  = 0040    M.PORT= 0080    M.RDEN= 0002
M.RDPE= 0008    M.RUN = 0004    M.SCLK= 0001    M.TRA = 0040
M.UNIT= 0007    M.WCLK= 0040    M.WCLN= 0080    M.WREN= 0080
M5.5  = 0001    M6.5  = 0002    M7.5  = 0004    NOTCAP= 0088
OKAY  = 00FF    OPRRAM= 4300    OPSTRT= 0058    OPVER = 0040
PADCNT= 00D5    PADCRC= 0080    PASSCN  4775    PDIAG = 0048
PEID  = 0C8A    PENAB = 004C    PESET = 0001    PHASE2  4579
PHASE3  46DE    PH2CN   4594  G PH31    46F6    PL    = 00B1
PRDD  = 004C    PRENF = 009C    PS    = 00B2    PSTAT = 0048
PSW   =%0009    P.AMTP= 0001    P.BCTC= 0040    P.CMDP= 0020
P.INTE= 0080    P.LCS = 0040    P.LWR = 0020    P.RDP = 0002
P.RPEN= 0004    P.RPST= 0002    P.RP0E= 0020    P.RP1E= 0010
P.RP2E= 0020    P.RP3E= 0010    P.SING= 0080    P.STAT= 0002
P.STPE= 0080    P.TACH= 0008    P.TUPR= 0010    P.WCSP= 0004
P.WDS = 0040    P.WFLP= 0001    P.WPEN= 0010    P.WP0E= 0008
P.WP1E= 0004    P.WP2E= 0008    P.WP3E= 0004    P.5VOK= 0002
QUE   = 281B    QUEM  = 281E    RAMT  = 0010    RARA  = 0006
RARAI = 0004    RCHBD0= 0048    RCHBD1= 0047    RCHOK = 0046
RCHTST= 000C    RCLRT = 000D    RCMD  = 000B    RCMLP = 0003
RCONT = 0080    RDATA = 0017    RDCLK = 0010    RDON  = 0011
READG = 0007    REND  = 0014    REQST = 2806    RESCHR= 00D1
REVTST= 0064    REWIND= 0004    RFIFOL= 0008    RGCLK = 0002
RGCRI = 0003    RIBG  = 0001    RILL  = 0012    RINST = 000C
RMCTST= 0008    RMK2  = 0013    RNOP  = 0000    RPATH = 0001
RPBAD = 0044    RPCHI = 0001    RPCLK = 0003    RPCTL = 0009
RPEI  = 0002    RPFAIL= 0000    RPF1  = 009D    RPF2  = 009E
RPOK  = 0043    RPOSTN= 0016    RPSTA = 0015    RRCMT = 000A
RSTAT = 0002    RTIEB = 000A    RTIER = 0030    RTM   = 0005
RUNKI = 0009    RUPTST= 005E    RWDUNL= 0005    R.AMT = 0001
R.BOP = 0008    R.DATA= 0040    R.DON = 0002    R.DRDY= 0010
R.END = 0010    R.ILL = 0004    R.JVOK= 0004    R.MK2 = 0008
R.PLOD= 0008    R.PLOO= 0010    R.PLO1= 0020    R.POST= 0020
```

```
R.STNM= 0002          R.STOP= 0004          R.STPC= 0001          R.TBJN= 0080
R.TSTD= 0040          R.VOK = 0080          R00H  = 0081          R00L  = 0080
R01H  = 0083          R01L  = 0082          R02H  = 0085          R02L  = 0084
R03H  = 0087          R03L  = 0086          R04H  = 0089          R04L  = 0088
R05H  = 008B          R05L  = 008A          R06H  = 008D          R06L  = 008C
R07H  = 008F          R07L  = 008E          R10H  = 0091          R10L  = 0090
R11H  = 0093          R11L  = 0092          R12H  = 0095          R12L  = 0094
R13H  = 0097          R13L  = 0096          R14H  = 0099          R14L  = 0098
R15H  = 009B          R15L  = 009A          R16H  = 009D          R16L  = 009C
R17H  = 009F          R17L  = 009E          R7.5  = 0010          SELCLR= 0000
SETATA= 00A1          SID   = 0080          SOD   = 0080          SOE   = 0040
SP    =%0008          SSCLK = 0040          SSTEP = 0005          STACK = 4FFF
STATRM= 4F20          STPCT   4F20          STRSP = 5000          TADR00= 0080
TADR01= 0081          TADR02= 0082          TADR03= 0083          TADR04= 0084
TADR05= 0085          TADR06= 0086          TADR07= 0087          TADR10= 0088
TADR11= 0089          TADR12= 008A          TADR13= 008B          TAMT  = 0044
TASEL = 0080          TCMD  = 0040          TC.FWD= 0040          TC.INH= 0008
TC.LWR= 0004          TC.REV= 0020          TC.WRT= 0010          TEDMSK  4777
TEMP    4F99          TEMPD   477D          TEMPP   477E          TEPMSK  4778
TEST1   4300          TEST2   434A          TMF   = 0099          TMRDY = 0040
TOUT    4779          TRKENA= 00D2          TSET  = 2803          TSTEND= 2818
TSTS  = 0040          TUNIT0  4435          TUNIT1  4489          TUNIT2  44D9
TUNIT3  4529          TUSEL0= 00D1          TUSEL1= 00D2          TU0CN0  4461  G
TU0CN1  4483  G       TU1CN0  44B1  G       TU1CN1  44D3  G       TU2CN0  4501  G
TU2CN1  4523  G       TU3CN0  4551  G       TU3CN1  4573  G       TU78  = 0010
T.ATH0= 0001          T.ATH1= 0002          T.BOT = 0004          T.EOT = 0002
T.FPT = 0001          T.NTHR= 0004          T.ONL = 0020          T.PES = 0008
T.PSBJ= 0020          T.PSOJ= 0008          T.PS1J= 0010          T.RDY = 0080
T.RDY0= 0040          T.RWD = 0010          T.SCLK= 0002          UIBG  = 00A1
UNITMP  4776          UNIT0   4371          UNIT1   4386          UNIT2   439A
UNIT3   43AE          VALFC   4F98          VALTB   4F95          VELTST= 005B
WDR.P = 0010          WMCCTL= 00D3          WMCERR= 00DA          WMCSTA= 00D0
WRTCLK= 0000          WRTDAT= 00D3          W.ACRC= 0004          W.CRC = 0008
W.DIAG= 0002          W.DONN= 0040          W.ECC = 0010          W.ENAB= 0080
W.ERR = 0020          W.FMT = 0070          W.GCR = 0010          W.LEFT= 0004
W.ONES= 0020          W.RESI= 0002          W.REV = 0004          W.ROME= 0010
W.RST = 0001          W.SKIP= 000F          W.WRIT= 0008          W.XFER= 0020
X     =%000A          XLCN1   4330  G       XLLP1   4312          X.DONN= 0080
X.ENAB= 0040          X.PEPE= 0002          X.ROME= 0001          X.WCLK= 0001
Y     =%000B          .     = 477F


 ERRORS DETECTED:  0

 *XMC1.A78/PTP,XMC1=NLIST,PARAM,MACRO,LIST,XMC1
 RUN-TIME: 4 6 0 SECONDS
 CORE USED:  10K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(25) 15-OCT-80 09:36
PARAM.M80 TABLE OF CONTENTS SEQ 0455

3	"MEMORY MAP" EXTENDED ADDRESS BIT
19	READ MICRO CONTROLLER CONTROL & STATUS WORDS
139	TRANSPORT/PORT COMMAND + STATUS WORDS
380	COMMON ADDRESS SPACE REGISTERS
444	MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485	MASSBUS CONTROL WORD DEFINITIONS
557	WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709	KEYBOARD/DISPLAY CONTROL + STATUS WORD
797	MISCELLANEOUS VARIABLE DEFINITIONS
801	RAM ADDRESS DEFINITIONS
819	LINKAGE TABLE ADDRESSES
842	BIT DEFINITIONS
856	RIM AND SIM INSTRUCTION BIT DEFINITIONS
877	INTERRUPTABLE INSTRUCTION MACROS
924	DIAGNOSTIC VARIABLES
945	COMMON VARIABLE STORAGE
959	MACRO CALL DEFINITIONS
1047	TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101	8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1333	TEST 1 - XMC - PE - PREAMBLE/POSTAMBLE TEST
1845	SUBROUTINE CLKSYS
1876	SUBROUTINE CLEAR ALL TU PORTS
1921	TABLE-EXPECTED XMC OUTPUT PC
2118	PROGRAM VARIABLES

```
961                 ;*************************************************************************
962                 ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
963                 ;ERROR MACRO CALLS
964                 ;
965                 ;       1  -    REQUEST HOST CPU TO PRINT:
966                 ;               "BYTE/SCLK COUNT NUMBER = LLL"
967                 ;               WHERE:
968                 ;               LLL = THE VALUE STORED IN CAS REGISTER 5
969                 ;                     (THE BYTE COUNT REGISTER 16 BITS).
970                 ;
971                 ;       2  -    REQUEST HOST CPU TO PRINT:
972                 ;               DATA FORMAT = MM
973                 ;               SKIP COUNT  = NN
974                 ;               WHERE:
975                 ;               MM = DATA FORMAT FROM CAS REGISTER 2
976                 ;               NN = SKIP COUNT FROM CAS REGISTER 2
977                 ;
978                 ;       3  -    REQUEST HOST CPU TO PRINT:
979                 ;               BYTE-SCLK COUNT = LLL
980                 ;               DATA FORMAT = MM
981                 ;               SKIP COUNT  = NN
982                 ;               WHERE:
983                 ;               LLL = AS ABOVE
984                 ;               MM  = AS ABOVE
985                 ;               NN  = AS ABOVE
986                 ;
987                 ;       4  -    REQUEST HOST TO PRINT:
988                 ;               TRANSITION COUNT = LLL
989                 ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
990                 ;
991                 ;       5  -    REQUEST HOST CPU TO PRINT:
992                 ;               EXPECTED 18 BITS =      E EEEEEE
993                 ;               ACTUAL 18 BITS =        A AAAAAA
994                 ;
995                 ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
996                 ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
997                 ;               OF CAS REG 15 LOW BYTE.
998                 ;
999                 ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
1000                ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
1001                ;               17 LOW BYTE SIGN BIT.
1002                ;
1003                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
1004                ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
1005                ;               AND/OR ACTUAL DATA.
1006                ;
1007                ;       7  -    REQUEST HOST CPU TO PRINT:
1008                ;               "SUBGROUP NUMBER = LLL"
1009                ;               WHERE:
1010                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
1011                ;                     (THE BYTE COUNT REGISTER 16 BITS).
1012                ;
1013                ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
1014                ;*************************************************************************
```

```
1016                    ; ************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ; ************************************************************************
1035                    ; ************************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ; ************************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  XMC2 - TRANSLATOR MICRO CONTROLLER PART #2
                                .SBTTL  TEST 1 - XMC - PE - PREAMBLE/POSTAMBLE TEST
                                ;ID     XMC2-TRANSLATOR MICRO CONTROLLER PART #2
4300                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - PE - PREAMBLE/POSTAMBLE TEST
4300                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;THIS TEST COMMANDS THE WRITE PATH TO WRITE A PE RECORD OF ONE BYTE
                                ;AND THEN SINGLE STEPS THE WRITE PATH (XMC, XMC) WHILE MONITORING THE
                                ;XMC "XL WCLK" BIT.  THE OUTPUT OF THE TRANSLATOR SHOULD INDICATE A PRE-
                                ;AMBLE, DATA, AND POSTAMBLE.  THIS OUTPUT IS CHECKED FOR SEQUENCE, CONTENT,
                                ;DATA STATES AND FLUX REVERSAL RATE.
4300                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  INITIALIZE THE BYTE, PAD AND ERROR COUNTERS
                                ;*  CLEAR THE ERROR COUNTER
                                ;*  CALL SUBROUTINE CLEAR
                                ;*  GET USER SELECTED TU PORT FOR TEST
                                ;*  ISSUE RESTART TO THE XMC CONTROL BYTE
                                ;*  SET THE SINGLE STEP CLOCK
                                ;*  CLEAR RESTART TO THE XMC CONTROL BYTE
                                ;*  SET WRITE PATH ENABLE TO SELECTED TU PORT
                                ;*  LOAD 300(8) IN THE RESIDUAL CHARACTER BYTE
                                ;*  SET FORMAT TO PDP-11 NORMAL
                                ;*  LOAD 1 IN THE BYTE COUNTER (BITS 7:0)
                                ;*  LOAD 0 IN THE BYTE COUNTER (BITS 15:8)
                                ;*  CLEAR THE TRACK ENABLE BYTES
                                ;*  SET THE DDR CONTROL TO "OUT"
                                ;*  LOAD DDRA WITH 0
                                ;*  LOAD DDRB WITH 374(8)
                                ;*  LOAD DDRC WITH 3
                                ;*  SET ALL TRACK ENABLE BITS
                                ;*  CLEAR THE POINTER COUNTER
                                ;*  INITIALIZE EXPECTED RESULTS POINTER
                                ;*  SET LOOP COUNT TO 377(8)
                                ;*  BGNDO
                                ;*  :  SET SUBROUTINE CLKSYS
                                ;*  :  DECREMENT THE LOOP COUNT
                                ;*  :  DO UNTIL THE LOOP COUNT=0
                                ;*  ENDDO
                                ;*  IF XMC ROM PARITY ERROR SET
                                ;*  :  THEN-ERROR-RESTART OF XMC NOT SUCCESSFUL
                                ;*  :  ELSE-CONTINUE
                                ;*  ENDIF
                                ;*  ISSUE XMC ENABLE, XMC ENABLE AND WRITE TO THE XMC CONTROL BYTE
                                ;*  SET THE LOOP COUNT TO 377(8)
```

```
1377                                    ;*  BGNDO
1378                                    ;*  :  BGNDO
1379                                    ;*  :  :  DO WHILE LOOP COUNT NOT=0 AND WRITE DATA STROBE=0
1380                                    ;*  :  :  DECREMENT THE LOOP COUNT
1381                                    ;*  :  ENDDO
1382                                    ;*  :  IF LOOP COUNT=0
1383                                    ;*  :  :  THEN-ERROR-TIMEOUT WAITING FOR WDS TO SET
1384                                    ;*  :  :  ELSE-CONTINUE
1385                                    ;*  :  ENDIF
1386                                    ;*  :  CALL SUBROUTINE CLKSYS
1387                                    ;*  :  CALL SUBROUTINE CLKSYS
1388                                    ;*  :  IF WRITE DATA STROBE=0
1389                                    ;*  :  :  THEN-CONTINUE
1390                                    ;*  :  :  ELSE-ERROR-WRITE DATA STROBE DID NOT CLEAR
1391                                    ;*  :  ENDIF
1392                                    ;*  :  CALL SUBROUTINE COMPPE
1393                                    ;*  :  INPUT THE ACTUAL TRANSLATOR OUTPUT FROM USER SELECTED TU PORT
1394                                    ;*  :  GET THE EXPECTED TRANSLATOR OUTPUT FROM THE TABLE
1395                                    ;*  :  IF ACTUAL DATA=EXPECTED DATA
1396                                    ;*  :  :  THEN-CONTINUE
1397                                    ;*  :  :  ELSE-ERROR-PE PARITY BIT CELL INCORRECT
1398                                    ;*  :  ENDIF
1399                                    ;*  :  IF WRITE MICRO STATUS=PE PARITY ERROR OR XMC ROM PARITY ERROR
1400                                    ;*  :  :  THEN-ERROR-PE PARITY ERROR OR ROM PARITY ERROR
1401                                    ;*  :  :  ELSE-CONTINUE
1402                                    ;*  :  ENDIF
1403                                    ;*  :  INCREMENT THE EXPECTED DATA TABLE POINTER
1404                                    ;*  :  INCREMENT THE POINTER COUNTER
1405                                    ;*  :  SET THE LOOP COUNT=56(8)
1406                                    ;*  :  BGNDO
1407                                    ;*  :  :  CALL SUBROUTINE CLKSYS
1408                                    ;*  :  :  DECREMENT THE LOOP COUNT
1409                                    ;*  :  :  DO UNTIL LOOP COUNT=0 OR WRITE DATA STROBE=1
1410                                    ;*  :  ENDDO
1411                                    ;*  :  IF LOOP COUNT=0
1412                                    ;*  :  :  THEN-IF POINTER COUNT NOT=80 OR 96
1413                                    ;*  :  :        :  THEN-ERROR-IMPROPER FLUX RATE
1414                                    ;*  :  :        :       CALL SUBROUTINE COMPPE
1415                                    ;*  :  :        :  ELSE-CONTINUE
1416                                    ;*  :  :       ENDIF
1417                                    ;*  :  :  ELSE-CONTINUE
1418                                    ;*  :  ENDIF
1419                                    ;*  :  CALL SUBROUTINE CLKSYS
1420                                    ;*  :  CALL SUBROUTINE CLKSYS
1421                                    ;*  :  CALL SUBROUTINE CLKSYS
1422                                    ;*  :  IF WRITE DATA STROBE=0
1423                                    ;*  :  :  THEN-CONTINUE
1424                                    ;*  :  :  ELSE-ERROR-WDS DID NOT RESET
1425                                    ;*  :  ENDIF
1426                                    ;*  :  INPUT THE ACTUAL TRANSLATOR OUTPUT AT SELECTED TU PORT
1427                                    ;*  :  GET THE EXPECTED TRANSLATOR OUTPUT FROM THE TABLE
1428                                    ;*  :  IF ACTUAL DATA=EXPECTED DATA
1429                                    ;*  :  :  THEN-CONTINUE
1430                                    ;*  :  :  ELSE-ERROR-PE BIT CELL INCORRECT
```

```
1431                                    ;*  :  ENDIF
1432                                    ;*  :  INPUT THE ACTUAL TRANSLATOR PARITY FROM SELECTED TU PORT
1433                                    ;*  :  GET THE EXPECTED TRANSLATOR PARITY FROM THE TABLE
1434                                    ;*  :  IF ACTUAL PARITY=EXPECTED PARITY
1435                                    ;*  :  :  THEN-CONTINUE
1436                                    ;*  :  :  ELSE-ERROR-PE PARITY BIT CELL INCORRECT
1437                                    ;*  :  ENDIF
1438                                    ;*  :  IF WRITE MICRO STATUS=PE PARITY ERROR OR XMC ROM PARITY ERROR
1439                                    ;*  :  :  THEN-ERROR-PE PARITY ERROR OR ROM PARITY ERROR
1440                                    ;*  :  :  ELSE-CONTINUE
1441                                    ;*  :  ENDIF
1442                                    ;*  :  IF THE POINTER COUNTER NOT=TABLE LENGTH
1443                                    ;*  :  :  THEN-INCREMENT THE POINTER COUNTER
1444                                    ;*  :  :       INCREMENT THE TABLE POINTER
1445                                    ;*  :  :       SET THE LOOP COUNT TO 46
1446                                    ;*  :  :  ELSE-CONTINUE
1447                                    ;*  :  ENDIF
1448                                    ;*  :  DO UNTIL THE POINTER COUNTER=TABLE LENGTH
1449                                    ;*  ENDDO
1450                                    ;*  IF TRANSLATOR NOT DONE=0
1451                                    ;*  :  THEN-ERROR-TRANSLATOR FINISHED BEFORE WRITE MICRO
1452                                    ;*  :  ELSE-IF WRITE MICRO NOT DONE=0
1453                                    ;*  :       :  THEN-CONTINUE
1454                                    ;*  :       :  ELSE-ERROR-WRITE MICRO NOT DONE
1455                                    ;*  :       ENDIF
1456                                    ;*  :       SET THE LOOP COUNT TO 20
1457                                    ;*  :       BGNDO
1458                                    ;*  :       :  CALL SUBROUTINE CLKSYS
1459                                    ;*  :       :  DECREMENT THE LOOP COUNT
1460                                    ;*  :       :  DO UNTIL LOOP COUNT=0 OR XMC DONE
1461                                    ;*  :       ENDDO
1462                                    ;*  :       IF LOOP COUNT=0
1463                                    ;*  :       :  THEN-ERROR-XMC DID NOT FINISH
1464                                    ;*  :       :  ELSE-CONTINUE
1465                                    ;*  :       ENDIF
1466                                    ;*  ENDIF
1467                                    ;*ENDTST
1468  4300                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1469                                    ;*XMC2 MICRO TEST 01
1470                                    ;*XMC2 MICRO ERROR 01
1471                                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1472                                    ;*M8958, M8955'S, M8959
1473                                    ;*OPERATOR ERROR NO TM78 UNITS SELECTED
1474                                    ;*FATAL ERROR - TEST ABORTED
1475                                    ;*
1476                                    ;*XMC2 MICRO TEST 01
1477                                    ;*XMC2 MICRO ERROR 02
1478                                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1479                                    ;*M8958, M8955'S, M8959
1480                                    ;*RESTART OF XMC DID NOT CLEAR XMC ROM ERROR BIT
1481                                    ;*ACTUAL = NNNN
```

```
1482                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1483                                    ;*
1484                                    ;*XMC2 MICRO TEST 01
1485                                    ;*XMC2 MICRO ERROR 03
1486                                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1487                                    ;*M8958, M8955'S, M8959
1488                                    ;*TIMEOUT WHILE WAITING FOR 'WDS' TO SET FOR 2ND HALF PE BIT CELL TIME
1489                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1490                                    ;*FATAL ERROR - TEST ABORTED
1491                                    ;*
1492                                    ;*XMC2 MICRO TEST 01
1493                                    ;*XMC2 MICRO ERROR 04
1494                                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1495                                    ;*M8958, M8955'S, M8959
1496                                    ;*2ND HALF PE BIT CELL 'WDS' FAILED TO CLEAR AFTER SET
1497                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1498                                    ;*
1499                                    ;*XMC2 MICRO TEST 01
1500                                    ;*XMC2 MICRO ERROR 05
1501                                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1502                                    ;*M8958, M8955'S, M8959
1503                                    ;*2ND HALF OF PE BIT CELL INCORRECT
1504                                    ;*ACTUAL = NNNN
1505                                    ;*EXPECTED = NNNN
1506                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1507                                    ;*
1508                                    ;*XMC2 MICRO TEST 01
1509                                    ;*XMC2 MICRO ERROR 06
1510                                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1511                                    ;*M8958, M8955'S, M8959
1512                                    ;*2ND HALF OF PE BIT CELL PARITY (WCSP) INCORRECT
1513                                    ;*ACTUAL = NNNN
1514                                    ;*EXPECTED = NNNN
1515                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1516                                    ;*
1517                                    ;*XMC2 MICRO TEST 01
1518                                    ;*XMC2 MICRO ERROR 07
1519                                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1520                                    ;*M8958, M8955'S, M8959
1521                                    ;*2ND HALF PE BIT CELL TIME - DETECTED PE PARITY ERROR OR
1522                                    ;*XMC ROM PARITY ERROR
1523                                    ;*ACTUAL = NNNN
1524                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1525                                    ;*
1526                                    ;*XMC2 MICRO TEST 01
1527                                    ;*XMC2 MICRO ERROR 10
1528                                    ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1529                                    ;*M8958, M8955'S, M8959
1530                                    ;*TIMEOUT WHILE WAITING FOR 'WDS' TO SET FOR 1ST HALF PE BIT CELL TIME
1531                                    ;*IMPROPER FLUX REVERSAL RATE WILL OCCUR ON TAPE
1532                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1533                                    ;*
1534                                    ;*XMC2 MICRO TEST 01
1535                                    ;*XMC2 MICRO ERROR 11
```

```
1536                            ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1537                            ;*M8958, M8955'S, M8959
1538                            ;*1ST HALF PE BIT CELL 'WDS' FAILED TO CLEAR AFTER SET
1539                            ;*BYTE/SCLK COUNT NUMBER = LLL
1540                            ;*
1541                            ;*XMC2 MICRO TEST 01
1542                            ;*XMC2 MICRO ERROR 12
1543                            ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1544                            ;*M8958, M8955'S, M8959
1545                            ;*1ST HALF OF PE BIT CELL INCORRECT
1546                            ;*ACTUAL = NNNN
1547                            ;*EXPECTED = NNNN
1548                            ;*BYTE/SCLK COUNT NUMBER = LLL
1549                            ;*
1550                            ;*XMC2 MICRO TEST 01
1551                            ;*XMC2 MICRO ERROR 13
1552                            ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1553                            ;*M8958, M8955'S, M8959
1554                            ;*1ST HALF OF PE BIT CELL PARITY (WCSP) INCORRECT
1555                            ;*ACTUAL = NNNN
1556                            ;*EXPECTED = NNNN
1557                            ;*BYTE/SCLK COUNT NUMBER = LLL
1558                            ;*
1559                            ;*XMC2 MICRO TEST 01
1560                            ;*XMC2 MICRO ERROR 14
1561                            ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1562                            ;*M8958, M8955'S, M8959
1563                            ;*1ST HALF PE BIT CELL TIME - DETECTED EITHER PE PARITY ERROR OR
1564                            ;*XMC ROM PARITY ERROR
1565                            ;*ACTUAL = NNNN
1566                            ;*BYTE/SCLK COUNT NUMBER = LLL
1567                            ;*
1568                            ;*XMC2 MICRO TEST 01
1569                            ;*XMC2 MICRO ERROR 15
1570                            ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1571                            ;*M8958, M8955'S, M8959
1572                            ;*'XMC NOT DONE' =1 AT END OF XMC PE POSTAMBLE
1573                            ;*XMC SHOULD HAVE FINISHED BEFORE START OF PE POSTAMBLE
1574                            ;*BYTE/SCLK COUNT NUMBER = LLL
1575                            ;*
1576                            ;*XMC2 MICRO TEST 01
1577                            ;*XMC2 MICRO ERROR 16
1578                            ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1579                            ;*M8958, M8955'S, M8959
1580                            ;*XMC FAILED TO FINISH WITHIN SEVERAL CLOCKS PAST END OF PE POSTAMBLE
1581                            ;*'XMC NOT DONE' =1
1582                            ;*BYTE/SCLK COUNT NUMBER = LLL
1583                            ;*
1584                            ;*XMC2 MICRO TEST 01
1585                            ;*XMC2 MICRO ERROR 17
1586                            ;*XMC2-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1587                            ;*M8958, M8955'S, M8959
1588                            ;*XMC FINISHED ('XMC NOT DONE' = 0) BEFORE THE XMC
1589                            ;*BYTE/SCLK COUNT NUMBER = LLL
```

```
1590  4300                                  S
 (1)                                        ; ************************************************************************
1591
1592  4300                                  TEST1:  TESTX   @1
 (1)  4300  3E  01              7.0                     MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28         18.0                     CALL    TSET            ;SETUP THE TEST
1593                                        ;%XMC2-PE-PREAMBLE/POSTAMBLE TEST-1 BYTE RECORD
1594                                        ;&M8958, M8955'S, M8959
1595  4305  AF                4.0                     XRA     A               ;CLEAR ALL TRACK ENABLE LINES
1596  4306  D3  D2            10.0                     OUT     TRKENA
1597  4308  D3  D2            10.0                     OUT     TRKENA
1598  430A                                  INICNT
 (1)  430A  3E  30              7.0                     MVI     A,@060          ;INIT THE
 (1)  430C  D3  D7            10.0                      OUT     CNTCTL          ;BYTE COUNTER
 (1)  430E  3E  70              7.0                     MVI     A,@160          ;INIT THE
 (1)  4310  D3  D7            10.0                      OUT     CNTCTL          ;PAD COUNTER
 (1)  4312  3E  B0              7.0                     MVI     A,@260          ;INIT THE
 (1)  4314  D3  D7            10.0                      OUT     CNTCTL          ;ERROR COUNTER
1599  4316                                  CLRECT
 (1)  4316  AF                  4.0                     XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  4317  D3  D6            10.0                      OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  4319  D3  D6            10.0                      OUT     ERRCNT          ;CLEAR BITS 15-8
1600  431B  CD  8B  45         18.0          CALL    CLEAR
1601  431E                                  REQ     @7,0,0,0
 (1)  431E  CD  06  28         18.0                     CALL    REQST
 (1)  4321  00                                         .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4322  00  00                                     .WORD   0               ;SYSTEM "0" COUNT
 (1)  4324  00  00                                     .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  4326  00                                         .BYTE                   ;DATA COMPARE FLAG IF -1
 (1)  4327  07                                         .BYTE   @7              ;REQUEST CODE
1602  4328                          RIN     R12L
 (1)  4328  DB  94              10.0                    IN      R12L            ;READ R12L INTO AC
 (1)  432A  7F                   4.0                    MOV     A,A             ;RETRY LINK
1603  432B  32  66  46         13.0          STA     UNITMP
1604  432E  06  00              7.0          MVI     B,0
1605  4330  E6  01              7.0          ANI     $01             ;UNIT 0 PRESENT?
1606  4332  C2  5E  43         10.0          JNZ     FOUND           ;YES-USE IT
1607  4335  04                   4.0          INR     B
1608  4336  3A  66  46         13.0          LDA     UNITMP          ;
1609  4339  E6  02              7.0          ANI     $02             ;UNIT 1 PRESENT?
1610  433B  C2  5E  43         10.0          JNZ     FOUND           ;YES-USE IT
1611  433E  04                   4.0          INR     B
1612  433F  3A  66  46         13.0          LDA     UNITMP          ;
1613  4342  E6  04              7.0          ANI     $04             ;UNIT 2 PRESENT?
1614  4344  C2  5E  43         10.0          JNZ     FOUND           ;YES-USE IT
1615  4347  04                   4.0          INR     B
1616  4348  3A  66  46         13.0          LDA     UNITMP
1617  434B  E6  08              7.0          ANI     $08             ;UNIT 3 PRESENT
1618  434D  C2  5E  43         10.0          JNZ     FOUND           ;YES-USE IT
1619                                                                ;ELSE - OPERATOR ERROR
```

```
1621  4350                                        ERR     EXIT,DUMMY
 (1)
 (1)  4350  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0001                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4353  01                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4354  00                                            .BYTE
 (1)  4355  CD  15  28      18.0                  DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4358  DA  6B  45      10.0                          JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1622                                    ;>OPERATOR ERROR NO TM78 UNITS SELECTED
1623                                    ;>FATAL ERROR - TEST ABORTED
1624  435B  C3  6B  45        10.0           JMP     EXIT
1625
1626  435E  DB  E0             10.0  FOUND:   IN      INTSTA          ;GET THE INTERRUPT STATUS
1627  4360  E6  80              7.0           ANI     BIT7            ;SAVE MB PORT SELECT BIT
1628  4362  B0                  4.0           ORA     B               ;OR IN TU PORT BITS
1629  4363  D3  E0             10.0           OUT     MBSEL           ;SELECT THE PORT
1630  4365  3E  40              7.0           MVI     A,P.LCS         ;KILL THE MTA BOARD
1631  4367  D3  48             10.0           OUT     PDIAG
1632  4369  AF                  4.0           XRA     A
1633  436A  D3  40             10.0           OUT     TCMD
1634
1635  436C  3E  01              7.0  T1LOOP:  MVI     A,W.RST         ;RESTART THE XMC
1636  436E  D3  D3             10.0           OUT     WMCCTL
1637  4370  3E  05              7.0           MVI     A,SSTEP         ;SET CLOCK TO SINGLE STEP
1638  4372  D3  F0             10.0           OUT     CLKCTL
1639  4374  AF                  4.0           XRA     A
1640  4375  D3  D3             10.0           OUT     WMCCTL          ;FINISH THE RESTART
1641
1642  4377  3E  10              7.0           MVI     A,P.WPEN        ;SET WRITE PATH ENABLE
1643  4379  D3  4C             10.0           OUT     PENAB
1644  437B  3E  C0              7.0           MVI     A,@300          ;SET FOR A 6 BYTE XFR (PE)
1645  437D  D3  D1             10.0           OUT     RESCHR          ;LOAD THE RESIDUAL CHAR.
1646  437F  3E  00              7.0           MVI     A,$0            ;SET FOR 11-NORMAL MODE
1647  4381  D3  D0             10.0           OUT     DATACTL         ;LOAD DATA CONTROL WORD
1648  4383  3E  01              7.0           MVI     A,1             ;LOAD THE BYTE COUNTER
1649  4385  D3  D4             10.0           OUT     BYTCNT
1650  4387  AF                  4.0           XRA     A
1651  4388  D3  D4             10.0           OUT     BYTCNT
1652  438A  D3  D2             10.0           OUT     TRKENA          ;INIT THE TRACKS
1653  438C  D3  D2             10.0           OUT     TRKENA
1654  438E  3E  88              7.0           MVI     A,@210          ;SET THE DDR CONTROL TO "OUT"
1655  4390  D3  DB             10.0           OUT     DDRCTL
1656  4392  AF                  4.0           XRA     A
1657  4393  D3  D8             10.0           OUT     DDRA            ;SET BITS 5-0,17-16 TO 0
1658  4395  3E  FC              7.0           MVI     A,@374
1659  4397  D3  D9             10.0           OUT     DDRB            ;SET BITS 13-8 TO 1'S & 7-6 TO 0
1660  4399  3E  03              7.0           MVI     A,3
1661  439B  D3  DA             10.0           OUT     DDRC            ;SET BITS 15-14 TO 1'S
1662                                    ;BYTE A IN 11-NORM = 000
1663                                    ;BYTE B IN 11-NORM = 377
1664  439D  3E  FF              7.0           MVI     A,@377
1665  439F  D3  D2             10.0           OUT     TRKENA          ;ENABLE ALL TRACKS
1666  43A1  D3  D2             10.0           OUT     TRKENA
1667
```

```
1668  43A3  21  B6  45  10.0          LXI   H,PXPTBL          ;LOAD ADDRESS OF EXPECTED
1669  43A6  22  67  46  16.0          SHLD  POINTER           ;RESULTS TABLE INTO POINTER
1670  43A9  3E  01       7.0          MVI   A,1
1671  43AB  32  69  46  13.0          STA   PNTCNT            ;SAVE THE COUNT
1672  43AE                           ROUT  R05L              ;START AT PE CHARACTER #1
 (1)  43AF  D3  8A      10.0                OUT   R05L                ;WRITE AC INTO R05L
 (1)  43B1  7F          4.0                 MOV   A,A                 ;RETRY LINK
1673  43B1  AF          4.0          XRA   A
1674  43B2                           ROUT  R05H
 (1)  43B2  D3  8B      10.0                OUT   R05H                ;WRITE AC INTO R05H
 (1)  43B4  7F          4.0                 MOV   A,A                 ;RETRY LINK
1675  43B5  32  6A  46  13.0          STA   PARBEXP           ;INIT THE EXPECTED PARITY BIT
1676  43B8  06  FF       7.0          MVI   B,$FF
1677
1678  43BA  CD  6E  45  18.0  RESTCK: CALL  CLKSYS
1679  43BD  05          4.0          DCR   B
1680  43BE  C2  BA  43  10.0          JNZ   RESTCK
1681  43C1  DB  E0      10.0          IN    INTSTA            ;CHECK TO SEE IF XMC PARITY ERROR SET
1682  43C3  E6  01       7.0          ANI   X.ROME            ;XMC ROM PARITY ERROR?
1683  43C5  CA  D0  43  10.0          JZ    RESTOK            ;JUMP IF RESTART WAS OK
1684  43C8                           ROUT  ADATA
 (1)  43C8  D3  94      10.0                OUT   ADATA               ;WRITE AC INTO ADATA
 (1)  43CA  7F          4.0                 MOV   A,A                 ;RETRY LINK
1685  43CB                           ERRA  T1LOOP,RESTOK,1
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' VALID
 (1)  43CB  CD  0F  28  18.0                      CALL  ERLPA               ;PROCESS ERROR - DO 2.3
 (1)       0002                               MSGN  =     MSGN+1                ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43CE  02                                 .BYTE MSGN              ;MESSAGE NUMBER ID
 (1)  43CF  01                                 .BYTE 1                 ;PRINT ROUTINE NUMBER
 (1)  43D0  CD  15  28  18.0              RESTOK::  CALL  CKLOP               ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43D3  DA  6C  43  10.0                       JC    T1LOOP              ;LOOP ADDRESS IF LOOP SPECIFIED
1686                               ;>RESTART OF XMC DID NOT CLEAR XMC ROM ERROR BIT
1687  43D6  3E  C8       7.0          MVI   A,W.ENAB!X.ENAB!W.WRITE
1688  43D8  D3  D3      10.0          OUT   WMCCTL            ;START THE OPERATION
1689
1690  43DA  06  FF       7.0  CLK1:   MVI   B,$FF             ;SET TIMEOUT COUNT TO 256
1691                                                          ;
1692  43DC  CD  6E  45  18.0  CLK:    CALL  CLKSYS            ;SINGLE STEP THE WRITE PATH
1693  43DF  DB  48      10.0          IN    PSTAT             ;GET THE WRITE DATA STROBE STATUS
1694  43E1  E6  40       7.0          ANI   P.WDS             ;P.WDS SET?
1695  43E3  C2  F8  43  10.0          JNZ   BGNPRE            ;YES-BEGINNING OF PREAMBLE
1696  43E6  05          4.0          DCR   B                 ;NO-DECREMENT CLOCK
1697  43E7  C2  DC  43  10.0          JNZ   CLK               ;CONTINUE UNTIL ZERO
1698  43EA                           ERR   T1LOOP,ABOCNO,1
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  43EA  CD  09  28  18.0                      CALL  ERLP                ;PROCESS ERROR - DO 2.3
 (1)       0003                               MSGN  =     MSGN+1                ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43ED  03                                 .BYTE MSGN              ;MESSAGE NUMBER ID
 (1)  43EE  01                                 .BYTE 1
 (1)  43EF  CD  15  28  18.0              ABOCNO::  CALL  CKLOP               ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43F2  DA  6C  43  10.0                       JC    T1LOOP              ;LOOP ADDRESS IF LOOP SPECIFIED
1699                               ;>TIMEOUT WHILE WAITING FOR 'WDS' TO SET FOR 2ND HALF PE BIT CELL TIME
1700                               ;>FATAL ERROR - TEST ABORTED
1701  43F5  C3  18  28  10.0          JMP   TSTEND
```

```
1702
1703  43F8  CD  6E  45   18.0  BGNPRE: CALL   CLKSYS          ;CLOCK THE BIT CLEAR
1704  43FB  CD  6E  45   18.0          CALL   CLKSYS
1705  43FE  DB  48        10.0          IN     PSTAT          ;GET WRT DATA STROBE STATUS WORD
1706  4400  E6  40         7.0          ANI    P.WDS          ;SEE IF BIT CLEARED AFTER CLOCK CYCLE
1707  4402  CA  0A  44   10.0          JZ     PREC1          ;JUMP IF OK
1708  4405                             ERR    T1LOOP,PREC1,1
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4405  CD  09  28   18.0                        CALL   ERLP           ;PROCESS ERROR - DO 2.3
 (1)       0004                                     MSGN   =      MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4408  04                                      .BYTE  MSGN           ;MESSAGE NUMBER ID
 (1)  4409  01                                      .BYTE  1
 (1)  440A  CD  15  28   18.0                PREC1:: CALL   CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  440D  DA  6C  43   10.0                        JC     T1LOOP         ;LOOP ADDRESS IF LOOP SPECIFIED
1709                          ;>2ND HALF PE BIT CELL 'WDS' FAILED TO CLEAR AFTER SET
1710  4410  CD  79  45   18.0          CALL   COMPPE          ;COMPLEMENT THE EXPECTED PARITY BIT
1711  4413  DB  40        10.0          IN     TSTS           ;GET THE DATA READ
1712
1713  4415  47           4.0          MOV    B,A            ;SAVE IT IN B=ADATA
1714  4416                             ROUT   ADATA          ;STORE ACTUAL DATA IN CAS
 (1)  4416  D3  94        10.0                 OUT    ADATA          ;WRITE AC INTO ADATA
 (1)  4418  7F            4.0                 MOV    A,A            ;RETRY LINK
1715                                          ;
1716  4419  2A  67  46   16.0          LHLD   POINTER        ;GET POINTER TO EXPECTED DATA
1717  441C  7E            7.0          MOV    A,M            ;GET EXPECTED DATA BYTE
1718  441D  B8            4.0          CMP    B              ;COMPARE WITH ACTUAL
1719  441E  CA  29  44   10.0          JZ     PREC1A         ;CONTINUE IF EQUAL
1720                                          ;ELSE-ERROR
1721  4421                             ROUT   EDATA          ;STORE EXPECTED DATA IN CAS
 (1)  4421  D3  95        10.0                 OUT    EDATA          ;WRITE AC INTO EDATA
 (1)  4423  7F            4.0                 MOV    A,A            ;RETRY LINK
1722  4424                             ERRB   T1LOOP,PREC1A,1
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4424  CD  12  28   18.0                        CALL   ERLPB          ;PROCESS ERROR - DO 2.3
 (1)       0005                                     MSGN   =      MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4427  05                                      .BYTE  MSGN           ;MESSAGE NUMBER ID
 (1)  4428  01                                      .BYTE  1              ;PRINT ROUTINE NUMBER
 (1)  4429  CD  15  28   18.0                PREC1A:: CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.2
 (1)  442C  DA  6C  43   10.0                        JC     T1LOOP         ;LOOP ADDRESS IF LOOP SPECIFIED
1723                          ;>2ND HALF OF PE BIT CELL INCORRECT
1724  442F  DB  48        10.0          IN     PSTAT          ;GET THE PORT PARITY BIT
1725  4431  E6  04         7.0          ANI    P.WCSP         ;SAVE ONLY THE PARITY BIT
1726  4433                             ROUT   ADATA          ;STORE ACTUAL PARITY IN CAS
 (1)  4433  D3  94        10.0                 OUT    ADATA          ;WRITE AC INTO ADATA
 (1)  4435  7F            4.0                 MOV    A,A            ;RETRY LINK
1727  4436  21  6A  46   10.0          LXI    H,PARBEXP      ;POINT TO EXPECTED PARITY BIT
1728  4439  BE            7.0          CMP    M              ;ACTUAL = EXPECTED PARITY BIT?
1729  443A  CA  48  44   10.0          JZ     PREC2          ;JUMP IF OK
1730  443D  3A  6A  46   13.0          LDA    PARBEXP        ;GET THE EXPECTED PARITY BIT
1731  4440                             ROUT   EDATA
 (1)  4440  D3  95        10.0                 OUT    EDATA          ;WRITE AC INTO EDATA
 (1)  4442  7F            4.0                 MOV    A,A            ;RETRY LINK
```

```
1733  4443                                        ERRB    T1LOOP,PREC2,1
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4443  CD  12  28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0006                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4446  06                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4447  01                                          .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4448  CD  15  28        18.0                PREC2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  444B  DA  6C  43        10.0                        JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1734                                    ;>2ND HALF OF PE BIT CELL PARITY (WCSP) INCORRECT
1735  444E  DB  E0              10.0          IN      INTSTA          ;SEE IF ERROR DETECTED
1736  4450  E6  03               7.0          ANI     X.PEPE!X.ROME
1737  4452  CA  5D  44          10.0          JZ      PREC2A          ;JUMP IF OK
1738  4455                                    ROUT    ADATA
 (1)  4455  D3  94              10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4457  7F                   4.0                  MOV     A,A             ;RETRY LINK
1739  4458                                    ERRA    T1LOOP,PREC2A,1
 (1)                                            ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4458  CD  0F  28        18.0                        CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)       0007                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  445B  07                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  445C  01                                          .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  445D  CD  15  28        18.0                PREC2A::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4460  DA  6C  43        10.0                        JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1740                                    ;>2ND HALF PE BIT CELL TIME - DETECTED PE PARITY ERROR OR
1741                                    ;>XMC ROM PARITY ERROR
1742  4463  2A  67  46          16.0          LHLD    POINTER         ;NO-UPDATE THE
1743  4466  23                   6.0          INX     H               ;TABLE POINTER
1744  4467  22  67  46          16.0          SHLD    POINTER         ;ADDRESS
1745  446A  3A  69  46          13.0          LDA     PNTCNT          ;UPDATE THE POINTER COUNT
1746  446D  3C                   4.0          INR     A
1747  446E  32  69  46          13.0          STA     PNTCNT          ;SAVE NEW COUNT
1748  4471                                    ROUT    RO5L            ;AND AGAIN
 (1)  4471  D3  8A              10.0                  OUT     RO5L            ;WRITE AC INTO RO5L
 (1)  4473  7F                   4.0                  MOV     A,A             ;RETRY LINK
1749
1750  4474  06  2E               7.0          MVI     B,46            ;LOAD THE CLOCK COUNT
1751  4476  CD  6E  45          18.0    1$:   CALL    CLKSYS          ;CLOCK THE WRITE PATH
1752  4479  DB  48              10.0          IN      PSTAT
1753  447B  E6  40               7.0          ANI     P.WDS           ;SEE IF WCLK SET
1754  447D  C2  96  44          10.0          JNZ     PREC3           ;JUMP IF ITS SET IN TIME
1755  4480  05                   4.0          DCR     B               ;DECREMENT THE COUNT
1756  4481  C2  76  44          10.0          JNZ     1$              ;CONTINUE UNTIL 23 CLOCKS
1757  4484  3A  69  46          13.0          LDA     PNTCNT          ;SEE IF SPECIAL CASE...END OF PREAMBLE
1758  4487  FE  50               7.0          CPI     80              ;40 PREAMBLE TICKS?
1759  4489  CA  9F  44          10.0          JZ      PREC3A          ;JUMP IF END OF PREAMBLE
1760  448C  FE  60               7.0          CPI     96              ;48 TICKS - AFTER ALL ONES ON POSTAMBLE
1761  448E  CA  9F  44          10.0          JZ      PREC3A          ;JUMP IF END OF 1'S TICK IN POSTAMBLE
```

```
1763  4491                                        ERR     T1LOOP,PREC3,1
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4491  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0008                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4494  08                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4495  01                                            .BYTE   1
 (1)  4496  CD  15  28      18.0                PREC3::   CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4499  DA  6C  43      10.0                          JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1764                                    ;>TIMEOUT WHILE WAITING FOR 'WDS' TO SET FOR 1ST HALF PE BIT CELL TIME
1765                                    ;>IMPROPER FLUX REVERSAL RATE WILL OCCUR ON TAPE
1766  449C  CD  79  45      18.0                CALL    COMPPE          ;COMPLIMENT THE EXPECTED PARITY BIT
1767  449F  CD  6E  45      18.0          PREC3A: CALL    CLKSYS          ;NOW CLOCK IT CLEAR
1768  44A2  CD  6E  45      18.0                CALL    CLKSYS
1769  44A5  CD  6E  45      18.0                CALL    CLKSYS          ;THIS IS A HACK CLOCK CAUSE OF XMC LOGIC
1770  44A8  DB  48          10.0                IN      PSTAT           ;GET STATUS WORD
1771  44AA  E6  40           7.0                ANI     P.WDS           ;SEE IF IT CLEARED
1772  44AC  CA  B4  44      10.0                JZ      PREC3B          ;JUMP IF OK
1773  44AF                                      ERR     T1LOOP,PREC3B,1
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44AF  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0009                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44B2  09                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44B3  01                                            .BYTE   1
 (1)  44B4  CD  15  28      18.0                PREC3B::          CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44B7  DA  6C  43      10.0                          JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1774                                    ;>1ST HALF PE BIT CELL 'WDS' FAILED TO CLEAR AFTER SET
1775  44BA  DB  40          10.0                IN      TSTS            ;GET THE DATA READ
1776  44BC  47               4.0                MOV     B,A             ;SAVE IT IN B=ADATA
1777  44BD                                      ROUT    ADATA           ;STORE ACTUAL DATA IN CAS
 (1)  44BD  D3  94          10.0                          OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44BF  7F               4.0                          MOV     A,A             ;RETRY LINK
1778                                                      ;
1779  44C0  2A  67  46      16.0                LHLD    POINTER         ;GET POINTER TO EXPECTED DATA
1780  44C3  7E               7.0                MOV     A,M             ;GET EXPECTED DATA BYTE
1781  44C4  B8               4.0                CMP     B               ;COMPARE WITH ACTUAL
1782  44C5  CA  D0  44      10.0                JZ      PRECN4          ;CONTINUE IF EQUAL
1783                                                      ;ELSE-ERROR
1784  44C8                                      ROUT    EDATA           ;STORE EXPECTED DATA IN CAS
 (1)  44C8  D3  95          10.0                          OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44CA  7F               4.0                          MOV     A,A             ;RETRY LINK
1785  44CB                                      ERRB    T1LOOP,PRECN4,1
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44CB  CD  12  28      18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000A                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44CE  0A                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44CF  01                                            .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  44D0  CD  15  28      18.0                PRECN4::          CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44D3  DA  6C  43      10.0                          JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1786                                    ;>1ST HALF OF PE BIT CELL INCORRECT
1787  44D6  DB  48          10.0                IN      PSTAT           ;GET THE PORT PARITY BIT
1788  44D8  E6  04           7.0                ANI     P.WCSP          ;SAVE ONLY THE PARITY BIT
1789  44DA                                      ROUT    ADATA           ;STORE ACTUAL PARITY IN CAS
 (1)  44DA  D3  94          10.0                          OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44DC  7F               4.0                          MOV     A,A             ;RETRY LINK
```

```
1790  44DD  21  6A  46  10.0          LXI   H,PARBEXP          ;SET POINTER
1791  44E0  BE               7.0      CMP   M                  ;ACTUAL = EXPECTED PARITY BIT?
1792  44E1  CA  EF  44  10.0          JZ    PREC5
1793  44E4  3A  6A  46  13.0          LDA   PARBEXP            ;GET THE EXPECTED PARITY BIT
1794  44E7                           ROUT  EDATA              ;SAVE FOR ERROR PRINTOUT
 (1)  44E7  D3  95        10.0                 OUT   EDATA            ;WRITE AC INTO EDATA
 (1)  44E9  7F             4.0                 MOV   A,A              ;RETRY LINK
1795  44EA                           ERRB  T1LOOP,PREC5,1
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44EA  CD  12  28  18.0                 CALL  ERLPB            ;PROCESS ERROR - DO 2.3
 (1)       000B                              MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44ED  0B                               .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  44EE  01                               .BYTE 1                ;PRINT ROUTINE NUMBER
 (1)  44EF  CD  15  28  18.0           PREC5:: CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44F2  DA  6C  43  10.0                 JC    T1LOOP           ;LOOP ADDRESS IF LOOP SPECIFIED
1796                           ;>1ST HALF OF PE BIT CELL PARITY (WCSP) INCORRECT
1797  44F5  DB  E0             10.0      IN    INTSTA             ;SEE IF ERROR DETECTED
1798  44F7  E6  03              7.0      ANI   X.PEPE!X.ROME
1799  44F9  CA  04  45  10.0          JZ    PREC5A             ;JUMP IF OK
1800  44FC                           ROUT  ADATA
 (1)  44FC  D3  94        10.0                 OUT   ADATA            ;WRITE AC INTO ADATA
 (1)  44FE  7F             4.0                 MOV   A,A              ;RETRY LINK
1801  44FF                           ERRA  T1LOOP,PREC5A,1
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  44FF  CD  0F  28  18.0                 CALL  ERLPA            ;PROCESS ERROR - DO 2.3
 (1)       000C                              MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4502  0C                               .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  4503  01                               .BYTE 1                ;PRINT ROUTINE NUMBER
 (1)  4504  CD  15  28  18.0           PREC5A::      CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4507  DA  6C  43  10.0                 JC    T1LOOP           ;LOOP ADDRESS IF LOOP SPECIFIED
1802                           ;>1ST HALF PE BIT CELL TIME - DETECTED EITHER PE PARITY ERROR OR
1803                           ;>XMC ROM PARITY ERROR
1804  450A  3A  69  46  13.0          LDA   PNTCNT             ;GET THE CURRENT PAIR COUNT
1805  450D  06  B0              7.0      MVI   B,<PTBLEN-PXPTBL> ;CALC END OF TEST...
1806  450F  B8                  4.0      CMP   B                  ;HAVE ALL POSTAMBLE YET?
1807  4510  CA  29  45  10.0          JZ    RECEND             ;YES - SEE IF XMC & XMC ARE FINISHED
1808  4513  2A  67  46  16.0          LHLD  POINTER            ;GET THE TABLE POINTER
1809  4516  23                6.0      INX   H                  ;UPDATE POINTER
1810  4517  22  67  46  16.0          SHLD  POINTER            ;SAVE IT
1811  451A  3A  69  46  13.0          LDA   PNTCNT             ;GET CURRENT COUNT
1812  451D  3C                  4.0      INR   A                  ;UPDATE IT
1813  451E  32  69  46  13.0          STA   PNTCNT             ;SAVE IT
1814  4521                           ROUT  R05L               ;SAVE AGAIN FOR ERROR PRINTING
 (1)  4521  D3  8A        10.0                 OUT   R05L             ;WRITE AC INTO R05L
 (1)  4523  7F             4.0                 MOV   A,A              ;RETRY LINK
1815
1816  4524  06  2E              7.0      MVI   B,46               ;RESET THE CLOCK COUNT FOR NEXT CYCLE
1817  4526  C3  DC  43  10.0          JMP   CLK                ;BACK TO LOOP
1818
1819  4529                     RECEND:
1820  4529  DB  D0             10.0      IN    WMCSTA             ;GET XMC STATUS
1821  452B  E6  80              7.0      ANI   X.DONN             ;XMC DONE
1822  452D  CA  60  45  10.0          JZ    SEQERR             ;YES-ERROR (OUT OF SEQUENCE)
1823  4530  DB  D0             10.0      IN    WMCSTA             ;INPUT THE XMC STATUS
```

```
1824  4532  E6  40        7.0        ANI   W.DONN          ;XMC DONE?
1825  4534  CA  3C  45   10.0        JZ    RECCN1          ;YES - SHOULD HAVE FINISHED AT POSTAMBLE
1826  4537                           ERR   T1LOOP,RECCN1,1
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4537  CD  09  28   18.0                 CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000D                    MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  453A  0D                     .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  453B  01                     .BYTE 1
 (1)  453C  CD  15  28   18.0  RECCN1::  CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  453F  DA  6C  43   10.0        JC    T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1827                          ;>'XMC NOT DONE' =1 AT END OF XMC PE POSTAMBLE
1828                          ;>XMC SHOULD HAVE FINISHED BEFORE START OF PE POSTAMBLE
1829  4542  06  14        7.0        MVI   B,20            ;LOAD THE TIMEOUT COUNT
1830  4544  CD  6E  45   18.0  3$:   CALL  CLKSYS          ;CLOCK THE SYSTEM
1831  4547  DB  D0        10.0        IN    WMCSTA          ;INPUT THE XMC STATUS
1832  4549  E6  80        7.0         ANI   X.DONN          ;XMC-DONE?
1833  454B  CA  57  45   10.0        JZ    RECCN3          ;YES-EXIT
1834  454E  05          4.0         DCR   B               ;NO-DECREMENT THE COUNT
1835  454F  C2  44  45   10.0        JNZ   3$              ;CONTINUE IF NOT ZERO
1836  4552                           ERR   T1LOOP,RECCN3,1
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4552  CD  09  28   18.0                 CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000E                    MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4555  0E                     .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  4556  01                     .BYTE 1
 (1)  4557  CD  15  28   18.0  RECCN3::  CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  455A  DA  6C  43   10.0        JC    T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1837                          ;>XMC FAILED TO FINISH WITHIN SEVERAL CLOCKS PAST END OF PE POSTAMBLE
1838                          ;>'XMC NOT DONE' =1
1839  455D  C3  6B  45   10.0        JMP   EXIT
1840
1841  4560                     SEQERR: ERR   T1LOOP,RECCN2,1
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4560  CD  09  28   18.0                 CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000F                    MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4563  0F                     .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  4564  01                     .BYTE 1
 (1)  4565  CD  15  28   18.0  RECCN2::  CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4568  DA  6C  43   10.0        JC    T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1842                          ;>XMC FINISHED ('XMC NOT DONE' = 0) BEFORE THE XMC
1843  456B  C3  18  28   10.0  EXIT:  JMP   TSTEND
```

```
                                        .SBTTL  SUBROUTINE CLKSYS
                                        SSUB
                                        ; ***************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLOCK THE SYSTEM
                                        SD
                                        ; ***************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE ISSUES ONE SINGLE STEP CLOCK
456E  3E  05            7.0  CLKSYS: MVI     A,SSTEP          ;GET THE SOFTWARE CLOCK CONTROL IMAGE
4570  F6  40            7.0          ORI     SSCLK            ;ADD IN 'CLK' BIT
4572  D3  F0           10.0          OUT     CLKCTL           ;LOAD CLOCK CONTROL
4574  E6  BF            7.0          ANI     @277             ;STRIP OFF CLOCK BIT
4576  D3  F0           10.0          OUT     CLKCTL           ;LOAD CLOCK CONTROL WORD
4578  C9               10.0          RET                      ;EXIT-CLOCK CYCLE

                                        SSUB
                                        ; ***************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*SUBROUTINE COMPARE
                                        SD
                                        ; ***************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*COMPPE -- ROUTINE TO COMPLIMENT THE EXPECTED PARITY BIT
                                        ;*      IN 'PARBEXP' STORAGE.
                                        S
                                        ; ***************************************************************************

4579  3A  6A  46       13.0  COMPPE: LDA     PARBEXP          ;GET THE CURRENT PARITY BIT INFO
457C  A7                4.0          ANA     A
457D  CA  85  45        10.0          JZ      PEBTON           ;SET IT ON

4580  AF                4.0  PEBTOFF: XRA    A
4581  32  6A  46       13.0          STA     PARBEXP
4584  C9               10.0          RET

4585  3E  04            7.0  PEBTON: MVI     A,4
4587  32  6A  46       13.0          STA     PARBEXP          ;SET IT ON
458A  C9               10.0          RET
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ;*********************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ;*********************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ;*********************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*    CLEAR PORT SELECT FOR TRANSPORT
                                        ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*    CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ;*********************************************************************
458B F5                  12.0  CLEAR:   PUSH    PSW             ;SAVE THE SELECTED PORT #
458C C5                  12.0           PUSH    B
458D 06 00                7.0           MVI     B,0             ;START TO CLEAR AT PORT #0
458F DB E0               10.0  CLRLP:   IN      INTSTA          ;GET MB SELECT INFO
4591 E6 80                7.0           ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
4593 B0                   4.0           ORA     B               ;ADD IN THE SELECTED PORT #
4594 D3 E0               10.0           OUT     MBSEL           ;RESET TO THIS PORT
4596 3E 80                7.0           MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
4598 D3 40               10.0           OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
459A AF                   4.0           XRA     A               ;CLEAR TU COMMAND A
459B D3 40               10.0           OUT     TCMD            ;
459D 3E 81                7.0           MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
459F D3 40               10.0           OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
45A1 3E 00                7.0           MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
45A3 D3 40               10.0           OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
45A5 AF                   4.0           XRA     A
45A6 D3 44               10.0           OUT     TAMT            ;CLEAR AMTIE WORD
45A8 D3 48               10.0           OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
45AA D3 4C               10.0           OUT     PENAB           ;CLEAR PORT ENABLE WORD
45AC 04                   4.0           INR     B               ;POINT TO THE NEXT PORT TO CLEAR
45AD 78                   4.0           MOV     A,B
45AE FE 04                7.0           CPI     4               ;DONE?
45B0 C2 8F 45            10.0           JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
45B3 C1                  10.0           POP     B               ;RESET B & C
45B4 F1                  10.0           POP     PSW             ;ALL DONE
45B5 C9                  10.0           RET                     ;EXIT
```

```
                                .SBTTL  TABLE-EXPECTED XMC OUTPUT PC
45B6                            PXPTBL:                         ;CHARACTER 1 - ALL ZEROS
45B6  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45B7  00                                .BYTE   $00             ;CHARACTER 2 - ALL ZEROS
45B8  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45B9  00                                .BYTE   $00             ;CHARACTER 3 - ALL ZEROS
45BA  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45BB  00                                .BYTE   $00             ;CHARACTER 4 - ALL ZEROS
45BC  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45BD  00                                .BYTE   $00             ;CHARACTER 5 - ALL ZEROS
45BE  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45BF  00                                .BYTE   $00             ;CHARACTER 6 - ALL ZEROS
45C0  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45C1  00                                .BYTE   $00             ;CHARACTER 7 - ALL ZEROS
45C2  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45C3  00                                .BYTE   $00             ;CHARACTER 8 - ALL ZEROS
45C4  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45C5  00                                .BYTE   $00             ;CHARACTER 9 - ALL ZEROS
45C6  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45C7  00                                .BYTE   $00             ;CHARACTER 10 - ALL ZEROS
45C8  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45C9  00                                .BYTE   $00             ;CHARACTER 11 - ALL ZEROS
45CA  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45CB  00                                .BYTE   $00             ;CHARACTER 12 - ALL ZEROS
45CC  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45CD  00                                .BYTE   $00             ;CHARACTER 13 - ALL ZEROS
45CE  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45CF  00                                .BYTE   $00             ;CHARACTER 14 - ALL ZEROS
45D0  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45D1  00                                .BYTE   $00             ;CHARACTER 15 - ALL ZEROS
45D2  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45D3  00                                .BYTE   $00             ;CHARACTER 16 - ALL ZEROS
45D4  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45D5  00                                .BYTE   $00             ;CHARACTER 17 - ALL ZEROS
45D6  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45D7  00                                .BYTE   $00             ;CHARACTER 18 - ALL ZEROS
45D8  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45D9  00                                .BYTE   $00             ;CHARACTER 19 - ALL ZEROS
45DA  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45DB  00                                .BYTE   $00             ;CHARACTER 20 - ALL ZEROS
45DC  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45DD  00                                .BYTE   $00             ;CHARACTER 21 - ALL ZEROS
45DE  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45DF  00                                .BYTE   $00             ;CHARACTER 22 - ALL ZEROS
45E0  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45E1  00                                .BYTE   $00             ;CHARACTER 23 - ALL ZEROS
45E2  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45E3  00                                .BYTE   $00             ;CHARACTER 24 - ALL ZEROS
45E4  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45E5  00                                .BYTE   $00             ;CHARACTER 25 - ALL ZEROS
45E6  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45E7  00                                .BYTE   $00             ;CHARACTER 26 - ALL ZEROS
45E8  FF                                .BYTE   $FF             ;2ND HALF PE BIT CELL TIME
45E9  00                                .BYTE   $00             ;CHARACTER 27 - ALL ZEROS
```

```
1975  45EA  FF                    .BYTE   $FF           ;2ND HALF PE BIT CELL TIME
1976  45EB  00                    .BYTE   $00           ;CHARACTER 28 - ALL ZEROS
1977  45EC  FF                    .BYTE   $FF           ;2ND HALF PE BIT CELL TIME
1978  45ED  00                    .BYTE   $00           ;CHARACTER 29 - ALL ZEROS
1979  45EE  FF                    .BYTE   $FF           ;2ND HALF PE BIT CELL TIME
1980  45EF  00                    .BYTE   $00           ;CHARACTER 30 - ALL ZEROS
1981  45F0  FF                    .BYTE   $FF           ;2ND HALF PE BIT CELL TIME
1982  45F1  00                    .BYTE   $00           ;CHARACTER 31 - ALL ZEROS
1983  45F2  FF                    .BYTE   $FF           ;2ND HALF PE BIT CELL TIME
1984  45F3  00                    .BYTE   $00           ;CHARACTER 32 - ALL ZEROS
1985  45F4  FF                    .BYTE   $FF           ;2ND HALF PE BIT CELL TIME
1986  45F5  00                    .BYTE   $00           ;CHARACTER 33 - ALL ZEROS
1987  45F6  FF                    .BYTE   $FF           ;2ND HALF PE BIT CELL TIME
1988  45F7  00                    .BYTE   $00           ;CHARACTER 34 - ALL ZEROS
1989  45F8  FF                    .BYTE   $FF           ;2ND HALF PE BIT CELL TIME
1990  45F9  00                    .BYTE   $00           ;CHARACTER 35 - ALL ZEROS
1991  45FA  FF                    .BYTE   $FF           ;2ND HALF PE BIT CELL TIME
1992  45FB  00                    .BYTE   $00           ;CHARACTER 36 - ALL ZEROS
1993  45FC  FF                    .BYTE   $FF           ;2ND HALF PE BIT CELL TIME
1994  45FD  00                    .BYTE   $00           ;CHARACTER 37 - ALL ZEROS
1995  45FE  FF                    .BYTE   $FF           ;2ND HALF PE BIT CELL TIME
1996  45FF  00                    .BYTE   $00           ;CHARACTER 38 - ALL ZEROS
1997  4600  FF                    .BYTE   $FF           ;2ND HALF PE BIT CELL TIME
1998  4601  00                    .BYTE   $00           ;CHARACTER 39 - ALL ZEROS
1999  4602  FF                    .BYTE   $FF           ;2ND HALF PE BIT CELL TIME
2000  4603  00                    .BYTE   $00           ;CHARACTER 40 - ALL ZEROS
2001  4604  FF                    .BYTE   $FF
2002
2003                        ;HERE IS THE END OF PREAMBLE...
2004
2005  4605  FF                    .BYTE   $FF           ;CHARACTER 41 - ALL ONES (NO WCLK)
2006  4606  00                    .BYTE   $00           ;2ND HALF PE BIT CELL TIME
2007                              .                     ;
2008                        ;6 BYTE DATA GROUP....DATA = 0'S & 1'S
2009                              .                     ;
2010  4607  00                    .BYTE   $00           ;DATA BYTE 1
2011  4608  FF                    .BYTE   $FF           ;BYTE A 11-NORMAL = 000
2012
2013  4609  FF                    .BYTE   $FF           ;DATA BYTE 2
2014  460A  00                    .BYTE   $00           ;BYTE B 11-NORMAL - 377
2015
2016  460B  00                    .BYTE   $00           ;DATA BYTE 3
2017  460C  FF                    .BYTE   $FF           ;DATA = 000
2018
2019  460D  FF                    .BYTE   $FF           ;DATA BYTE 4
2020  460E  00                    .BYTE   $00           ;DATA = 377
2021
2022  460F  00                    .BYTE   $00           ;DATA BYTE 5
2023  4610  FF                    .BYTE   $FF           ;DATA = 000
2024
2025  4611  FF                    .BYTE   $FF           ;DATA BYTE 6
2026  4612  00                    .BYTE   $00           ;DATA = 000 ... LAST DATA BYTE
2027
2028                        ;XMC SHOULD SEND 6 DATA BYTES THEN 2 PAD BYTES (000)
```

```
                                        ;PADS ARE IGNORED BY THE TRANSLATOR...
                                                                        ;
4613  FF                                        .BYTE   $FF             ;POSTAMBLE GROUP
4614  00                                        .BYTE   $00             ;CHARACTER 1 - ALL ONES
4615  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
4616  FF                                        .BYTE   $FF             ;CHARACTER 2 - ALL ZEROS
4617  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
4618  FF                                        .BYTE   $FF             ;CHARACTER 3 - ALL ZEROS
4619  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
461A  FF                                        .BYTE   $FF             ;CHARACTER 4 - ALL ZEROS
461B  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
461C  FF                                        .BYTE   $FF             ;CHARACTER 5 - ALL ZEROS
461D  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
461E  FF                                        .BYTE   $FF             ;CHARACTER 6 - ALL ZEROS
461F  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
4620  FF                                        .BYTE   $FF             ;CHARACTER 7 - ALL ZEROS
4621  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
4622  FF                                        .BYTE   $FF             ;CHARACTER 8 - ALL ZEROS
4623  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
4624  FF                                        .BYTE   $FF             ;CHARACTER 9 - ALL ZEROS
4625  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
4626  FF                                        .BYTE   $FF             ;CHARACTER 10 - ALL ZEROS
4627  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
4628  FF                                        .BYTE   $FF             ;CHARACTER 11 - ALL ZEROS
4629  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
462A  FF                                        .BYTE   $FF             ;CHARACTER 12 - ALL ZEROS
462B  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
462C  FF                                        .BYTE   $FF             ;CHARACTER 13 - ALL ZEROS
462D  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
462E  FF                                        .BYTE   $FF             ;CHARACTER 14 - ALL ZEROS
462F  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
4630  FF                                        .BYTE   $FF             ;CHARACTER 15 - ALL ZEROS
4631  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
4632  FF                                        .BYTE   $FF             ;CHARACTER 16 - ALL ZEROS
4633  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
4634  FF                                        .BYTE   $FF             ;CHARACTER 17 - ALL ZEROS
4635  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
4636  FF                                        .BYTE   $FF             ;CHARACTER 18 - ALL ZEROS
4637  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
4638  FF                                        .BYTE   $FF             ;CHARACTER 19 - ALL ZEROS
4639  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
463A  FF                                        .BYTE   $FF             ;CHARACTER 20 - ALL ZEROS
463B  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
463C  FF                                        .BYTE   $FF             ;CHARACTER 21 - ALL ZEROS
463D  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
463E  FF                                        .BYTE   $FF             ;CHARACTER 22 - ALL ZEROS
463F  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
4640  FF                                        .BYTE   $FF             ;CHARACTER 23 - ALL ZEROS
4641  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
4642  FF                                        .BYTE   $FF             ;CHARACTER 24 - ALL ZEROS
4643  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
4644  FF                                        .BYTE   $FF             ;CHARACTER 25 - ALL ZEROS
4645  00                                        .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
4646  FF                                        .BYTE   $FF             ;CHARACTER 26 - ALL ZEROS
```

```
2083  4647  00                .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2084  4648  FF                .BYTE   $FF             ;CHARACTER 27 - ALL ZEROS
2085  4649  00                .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2086  464A  FF                .BYTE   $FF             ;CHARACTER 28 - ALL ZEROS
2087  464B  00                .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2088  464C  FF                .BYTE   $FF             ;CHARACTER 29 - ALL ZEROS
2089  464D  00                .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2090  464E  FF                .BYTE   $FF             ;CHARACTER 30 - ALL ZEROS
2091  464F  00                .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2092  4650  FF                .BYTE   $FF             ;CHARACTER 31 - ALL ZEROS
2093  4651  00                .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2094  4652  FF                .BYTE   $FF             ;CHARACTER 32 - ALL ZEROS
2095  4653  00                .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2096  4654  FF                .BYTE   $FF             ;CHARACTER 33 - ALL ZEROS
2097  4655  00                .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2098  4656  FF                .BYTE   $FF             ;CHARACTER 34 - ALL ZEROS
2099  4657  00                .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2100  4658  FF                .BYTE   $FF             ;CHARACTER 35 - ALL ZEROS
2101  4659  00                .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2102  465A  FF                .BYTE   $FF             ;CHARACTER 36 - ALL ZEROS
2103  465B  00                .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2104  465C  FF                .BYTE   $FF             ;CHARACTER 37 - ALL ZEROS
2105  465D  00                .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2106  465E  FF                .BYTE   $FF             ;CHARACTER 38 - ALL ZEROS
2107  465F  00                .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2108  4660  FF                .BYTE   $FF             ;CHARACTER 39 - ALL ZEROS
2109  4661  00                .BYTE   $00             ;1ST HALF PE BIT CELL TIME - POSTAMBLE
2110  4662  FF                .BYTE   $FF             ;CHARACTER 40 - ALL ZEROS
2111  4663  00                .BYTE   $00             ;1ST HALF
2112  4664  FF                .BYTE   $FF             ;CHARACTER 41 - ALL ZEROS
2113  4665  00                .BYTE   $00             ;THIS IS THE LAST TICK CHECK
2114  4666                  PTBLEN:
2115
2116
2117
2118                        .SBTTL  PROGRAM VARIABLES
2119
2120  4666  00              UNITMP: .BYTE   0        ;UNIT MAP
2121  4667  00    00        POINTER: .WORD  0        ;POINTER TO EXPECTED DATA TABLE
2122  4669  00              PNTCNT: .BYTE   0        ;COUNT OF COMPARE POINTER
2123  466A  00              PARBEXP: .BYTE  0        ;STORAGE FOR EXPECTED PARITY BIT
2124        0000                      .END
```

```
A     =%0007          ABOCNO  43EF  G     ADATA = 0094          AMTIEP= 0001
AMTIE7= 0002          ARAIDF= 0098        ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91          B     =%0000        BADST = 0090          BGNPRE  43F8
BIT0  = 0001          BIT1  = 0002        BIT15 = 8000          BIT2  = 0004
BIT3  = 0008          BIT4  = 0010        BIT5  = 0020          BIT6  = 0040
BIT7  = 0080          BIT8  = 0100        BIT9  = 0200          BRKPBC= 4F0A
BRKRAM= 4F10          BRKSTR= 4E60        BRKXCT= 4F00          BSAVE   4F9C
BYTCNT= 00D4          BYTEH   4F24        BYTEL   4F23          C     =%0001
CASCT   4F21          CASCTL= 00A0        CASSTA= 00A0          CATTH = 0089
CATTL = 0088          CBUSST= 00A1        CBYTH = 008B          CBYTL = 008A
CDG1H = 0087          CDG1L = 0086        CDG2H = 0093          CDG2L = 0092
CDG3H = 0095          CDG3L = 0094        CDVTH = 008D          CDVTL = 008C
CHPTIE= 0028          CHOTIE= 0020        CH1TIE= 0021          CH2TIE= 0022
CH3TIE= 0023          CH4TIE= 0024        CH5TIE= 0025          CH6TIE= 0026
CH7TIE= 0027          CKLOP = 2815        CLEAR   458B          CLK     43DC
CLKCTL= 00F0          CLKSYS  456E        CLK1    43DA          CLOCK   4F26
CLRLP   458F          CMC0H = 0099        CMC0L = 0098          CMC1H = 009B
CMC1L = 009A          CMC2H = 009D        CMC2L = 009C          CMC3H = 009F
CMC3L = 009E          CMINH = 0097        CMINL = 0096          CNTCTL= 00D7
COMPPE  4579          CRCWRD= 0018        CSAVE   4F9D          CSRLH = 0091
CSRLL = 0090          CTCH  = 0085        CTCL  = 0084          CTSTH = 008F
CTSTL = 008E          CXCTH = 0081        CXCTL = 0080          CXINH = 0083
CXINL = 0082          C.    = 0001        C.AVAI= 0080          C.DP  = 0008
C.DSE = 0010          C.DTU = 0003        C.DVA = 0008          C.FAIL= 00FC
C.FMT = 0070          C.FNCT= 003E        C.GO  = 0001          C.INTC= 00FE
C.MAIN= 0020          C.NSA = 0080        C.RCT = 00FC          C.SER = 0080
C.SHR = 0040          C.SKPC= 000F        C.TAPE= 0040          C.WCS = 0002
D     =%0002          DATACT= 00D0        DBUS    4F28          DBUSCT= 00C0
DBUSST= 00C0          DDRA  = 00D8        DDRAIN= 0010          DDRB  = 00D9
DDRBIN= 0002          DDRC  = 00DA        DDRCIN= 0001          DDRCO = 0088
DDRCTL= 00DB          DIAFLG  4F22        DIAGPG= 4300          DIAGRM= 4F90
DIARD = 000B          DONE1 = 0045        DONINT= 0010          DSAVE   4F9E
DSE   = 0006          DUMMY   4355  G     D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010          D.LAGC= 0020        D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008          D.WR4 = 0080        E     =%0003          ECCBAD= 0042
ECCCOR= 0019          ECCOK = 0041        ECCSTA= 001A          ECCTST= 000E
EDATA = 0095          EOTCLR= 0003        ERFLG   4F93          ERLP  = 2809
ERLPA = 280F          ERLPB = 2812        ERLPE = 280C          ERNUM   4F90
ERRCNT= 00D6          ESAVE   4F9F        EXIT    456B          E.ACRC= 0010
E.AMT = 0020          E.CDP = 0080        E.CRC = 0080          E.PNTR= 0008
E.RPE = 0040          E.STEC= 0001        E.TTEC= 0002          E.UNC = 0004
FIFORD= 006A          FORMAT  4F25        FOUND   435E          FWDTST= 0061
GCRID = 0089          GCRSET= 0002        GOODTM= 0092          H     =%0004
HLSAVE  4FA0          IE    = 0008        INTSTA= 00E0          ITERA   4F9A
I.PWR = 0020          I.RMPE= 0040        I5.5  = 0010          I6.5  = 0020
I7.5  = 0040          KCALL = 005F        KCLR  = 007B          KDEP  = 003F
KENAB = 0078          KEXAM = 003E        KEYBRD= 00C8          KEY1  = 0078
KEY10 = 006D          KEY11 = 006E        KEY12 = 006F          KEY13 = 005C
KEY14 = 005D          KEY15 = 005E        KEY16 = 005F          KEY17 = 003C
KEY18 = 003D          KEY19 = 003E        KEY2  = 0079          KEY20 = 003F
KEY3  = 007A          KEY4  = 007B        KEY5  = 0074          KEY6  = 0075
KEY7  = 0076          KEY8  = 0077        KEY9  = 006C          KINTA = 006F
KLDAD = 003D          KN0   = 003C        KN1   = 005C          KN2   = 005D
KN3   = 005E          KN4   - 006C        KN5   = 006D          KN6   = 006E
```

```
KN7    = 0074          KN8    = 0075          KN9    = 0076          KU2    = 0079
KU3    = 007A          KU8    = 0077          L      =%0005          LBLANK= 000F
LCE    = 000B          LCH    = 000C          LCL    = 000D          LCP    = 000E
LC0    = 0000          LC1    = 0001          LC2    = 0002          LC3    = 0003
LC4    = 0004          LC5    = 0005          LC6    = 0006          LC7    = 0007
LC8    = 0008          LC9    = 0009          LDLEDA= 00CA           LDLEDB= 00CB
LDLEDC= 00CC           LDLEDD= 00CD           LDLEDE= 00CE           LDLEDF= 00CF
LKDIAG= 2800           LKKBD = 004C           LKKEY = 0049           LKLWMG= 0058
LKLWMP= 0055           LKLWPG= 0052           LKLWPP= 004F           LKMOD7= 0046
LKOPR = 0046           LPFLG   4F94           LPNUM   4F92           M      =%0006
MBSEL = 00E0           MB.A   = 0008          MB.B   = 0004          MEMTOP= 4FFF
MINUS = 000A           MM     = 8000          MSE    = 0008          MSGN   = 000F
MTACLR= 0000           MT.ARA= 0020           MT.CPE= 0080           MT.DSE= 0001
MT.FWD= 0040           MT.INH= 0008           MT.LWR= 0004           MT.MOT= 0002
MT.NWT= 0080           MT.PEC= 0040           MT.PSB= 0004           MT.PS0= 0001
MT.PS1= 0002           MT.REV= 0020           MT.WRT= 0010           MT.Z   = 0008
M.ATA = 0080           M.CAPE= 0020           M.CONT= 0080           M.DEM = 0020
M.EBL = 0010           M.EXC = 0008           M.FAIL= 0008           M.ILR = 0010
M.INIT= 0010           M.OCC = 0020           M.ONLI= 0001           M.PE   = 0040
M.PORT= 0080           M.RDEN= 0002           M.RDPE= 0008           M.RUN = 0004
M.SCLK= 0001           M.TRA = 0040           M.UNIT= 0007           M.WCLK= 0040
M.WCLN= 0080           M.WREN= 0080           M5.5   = 0001          M6.5   = 0002
M7.5   = 0004          NOTCAP= 0088           OKAY   = 00FF          OPRRAM= 4300
OPSTRT= 0058           OPVER = 0040           PADCNT= 00D5           PADCRC= 0080
PARBEX  466A           PDIAG = 0048           PEBTOF  4580           PEBTON  4585
PEID   = 008A          PENAB = 004C           PESET = 0001           PL     = 00B1
PNTCNT  4669           POINTE  4667           PRDD   = 004C          PRECN4  44D0  G
PREC1   440A  G        PREC1A  4429  G        PREC2   4448  G        PREC2A  445D  G
PREC3   4496  G        PREC3A  449F           PREC3B  44B4  G        PREC5   44EF  G
PREC5A  4504  G        PRENF = 009C           PS     = 00B2          PSTAT = 0048
PSW    =%0009          PTBLEN  4666           PXPTBL  45B6           P.AMTP= 0001
P.BCTC= 0040           P.CMDP= 0020           P.INTE= 0080           P.LCS = 0040
P.LWR = 0020           P.RDP = 0002           P.RPEN= 0004           P.RPST= 0002
P.RPOE= 0020           P.RP1E= 0010           P.RP2E= 0020           P.RP3E= 0010
P.SING= 0080           P.STAT= 0002           P.STPE= 0080           P.TACH= 0008
P.TUPR= 0010           P.WCSP= 0004           P.WDS  = 0040          P.WFLP= 0001
P.WPEN= 0010           P.WPOE= 0008           P.WP1E= 0004           P.WP2E= 0008
P.WP3E= 0004           P.5VOK= 0002           QUE    = 281B          QUEM   = 281E
RAMT   = 0010          RARA   = 0006          RARAI  = 0004          RCHBD0= 0048
RCHBD1= 0047           RCHOK  = 0046          RCHTST= 000C           RCLRT  = 000D
RCMD   = 000B          RCMLP  = 0003          RCONT  = 0080          RDATA  = 0017
RDCLK  = 0010          RDON   = 0011          READG  = 0007          RECCN1  453C  G
RECCN2  4565  G        RECCN3  4557  G        RECEND  4529           REND   = 0014
REQST  = 2806          RESCHR= 00D1           RESTCK  43BA           RESTOK  43D0  G
REVTST= 0064           REWIND= 0004           RFIFOL= 0008           RGCLK  = 0002
RGCRI  = 0003          RIBG   = 0001          RILL   = 0012          RINST  = 000C
RMCTST= 0008           RMK2   = 0013          RNOP   = 0000          RPATH  = 0001
RPBAD  = 0044          RPCHI  = 0001          RPCLK  = 0003          RPCTL  = 0009
RPEI   = 0002          RPFAIL= 0000           RPF1   = 009D          RPF2   = 009E
RPOK   = 0043          RPOSTN= 0016           RPSTA  = 0015          RRCMT  = 000A
RSTAT  = 0002          RTIEB  = 000A          RTIER  = 0030          RTM    = 0005
RUNKI  = 0009          RUPTST= 005E           RWDUNL= 0005           R.AMT  = 0001
R.BOP  = 0008          R.DATA= 0040           R.DON  = 0002          R.DRDY= 0010
R.END  = 0010          R.ILL  = 0004          R.JVOK= 0004           R.MK2  = 0008
```

```
R.PLOD= 0008    R.PLOO= 0010    R.PLO1= 0020    R.POST= 0020
R.STNM= 0002    R.STOP= 0004    R.STPC= 0001    R.TBJN= 0080
R.TSTD= 0040    R.VOK = 0080    R00H  = 0081    R00L  = 0080
R01H  = 0083    R01L  = 0082    R02H  = 0085    R02L  = 0084
R03H  = 0087    R03L  = 0086    R04H  = 0089    R04L  = 0088
R05H  = 008B    R05L  = 008A    R06H  = 008D    R06L  = 008C
R07H  = 008F    R07L  = 008E    R10H  = 0091    R10L  = 0090
R11H  = 0093    R11L  = 0092    R12H  = 0095    R12L  = 0094
R13H  = 0097    R13L  = 0096    R14H  = 0099    R14L  = 0098
R15H  = 009B    R15L  = 009A    R16H  = 009D    R16L  = 009C
R17H  = 009F    R17L  = 009E    R7.5  = 0010    SELCLR= 0000
SEQERR  4560    SETATA= 00A1    SID   = 0080    SOD   = 0080
SOE   = 0040    SP    =%0008    SSCLK = 0040    SSTEP = 0005
STACK = 4FFF    STATRM= 4F20    STPCT   4F20    STRSP = 5000
TADR00= 0080    TADR01= 0081    TADR02= 0082    TADR03= 0083
TADR04= 0084    TADR05= 0085    TADR06= 0086    TADR07= 0087
TADR10= 0088    TADR11= 0089    TADR12= 008A    TADR13= 008B
TAMT  = 0044    TASEL = 0080    TCMD  = 0040    TC.FWD= 0040
TC.INH= 0008    TC.LWR= 0004    TC.REV= 0020    TC.WRT= 0010
TEMP    4F99    TEST1   4300    TMF   = 0099    TMRDY = 0040
TRKENA= 00D2    TSET  = 2803    TSTEND= 2818    TSTS  = 0040
TUSEL0= 00D1    TUSEL1= 00D2    TU78  = 0010    T.ATH0= 0001
T.ATH1= 0002    T.BOT = 0004    T.EOT = 0002    T.FPT = 0001
T.NTHR= 0004    T.ONL = 0020    T.PES = 0008    T.PSBJ= 0020
T.PSOJ= 0008    T.PS1J= 0010    T.RDY = 0080    T.RDYO= 0040
T.RWD = 0010    T.SCLK= 0002    T1LOOP  436C    UIBG  = 00A1
UNITMP  4666    VALFC   4F98    VALTB   4F95    VELTST= 005B
WDR.P = 0010    WMCCTL= 00D3    WMCERR= 00DA    WMCSTA= 00D0
WRTCLK= 0000    WRTDAT= 00D3    W.ACRC= 0004    W.CRC = 0008
W.DIAG= 0002    W.DONN= 0040    W.ECC = 0010    W.ENAB= 0080
W.ERR = 0020    W.FMT = 0070    W.GCR = 0010    W.LEFT= 0004
W.ONES= 0020    W.RESI= 0002    W.REV = 0004    W.ROME= 0010
W.RSI = 0001    W.SKIP= 000F    W.WRIT= 0008    W.XFER= 0020
X     =%000A    X.DONN= 0080    X.ENAB= 0040    X.PEPE= 0002
X.ROME= 0001    X.WCLK= 0001    Y     =%000B    .     = 466B


 ERRORS DETECTED:  0

 *XMC2.A78/PTP,XMC2=NLIST,PARAM,MACRO,LIST,XMC2
 RUN-TIME: 4 5 0 SECONDS
 CORE USED:  10K
```

     3   "MEMORY MAP" EXTENDED ADDRESS BIT
    19   READ MICRO CONTROLLER CONTROL & STATUS WORDS
   139   TRANSPORT/PORT COMMAND + STATUS WORDS
   380   COMMON ADDRESS SPACE REGISTERS
   444   MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
   485   MASSBUS CONTROL WORD DEFINITIONS
   557   WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
   709   KEYBOARD/DISPLAY CONTROL + STATUS WORD
   797   MISCELLANEOUS VARIABLE DEFINITIONS
   801   RAM ADDRESS DEFINITIONS
   819   LINKAGE TABLE ADDRESSES
   842   BIT DEFINITIONS
   856   RIM AND SIM INSTRUCTION BIT DEFINITIONS
   877   INTERRUPTABLE INSTRUCTION MACROS
   924   DIAGNOSTIC VARIABLES
   945   COMMON VARIABLE STORAGE
   959   MACRO CALL DEFINITIONS
  1047   TM78 MICRO-DIAGNOSTIC MACRO CALLS
  1101   8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
  1333   TEST 1 - XMC - GCR PREAMBLE/POSTAMBLE TEST
  1795   SUBROUTINE CLKSYS
  1809   SUBROUTINE CLEAR ALL TU PORTS
  2310   TABLE EXPECTED PARITY
  2382   PROGRAM VARIABLES

```
                ;*************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;               "BYTE/SCLK COUNT NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               MM = DATA FORMAT FROM CAS REGISTER 2
                ;               NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;               BYTE-SCLK COUNT = LLL
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               LLL = AS ABOVE
                ;               MM  = AS ABOVE
                ;               NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;               TRANSITION COUNT = LLL
                ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;               EXPECTED 18 BITS =      E EEEEEE
                ;               ACTUAL 18 BITS =        A AAAAAA
                ;
                ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;               OF CAS REG 15 LOW BYTE.
                ;
                ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;               17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;               AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;               "SUBGROUP NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       10 -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;*************************************************************************
```

```
1016                    ; ***************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ; ***************************************************************************
1035                    ; ***************************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ; ***************************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  XMC3 - TRANSLATOR MICRO CONTROLLER PART #3
                                .SBTTL  TEST 1 - XMC - GCR PREAMBLE/POSTAMBLE TEST
                                ;ID     XMC3-TRANSLATOR MICRO CONTROLLER PART #3
4300                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*XMC - GCR PREAMBLE/POSTAMBLE TEST
4300                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;THIS TEST COMMANDS THE WRITE PATH TO WRITE A GCR RECORD OF SEVEN BYTES
                                ;AND THEN SINGLE STEPS THE WRITE PATH (WMC, XMC) WHILE MONITORING THE
                                ;XMC "XL WCLK" BIT.  THE OUTPUT OF THE TRANSLATOR SHOULD INDICATE A PRE-
                                ;AMBLE, END MARK, DATA GROUP, MARK-2, RESIDUAL GROUP, CRC GROUP, AND POSTAMBLE.
                                ;THIS OUTPUT IS CHECKED FOR SEQUENCE, CONTENT, DATA STATES AND FLUX REVERSAL RATE.
4300                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    INITIALIZE THE BYTE COUNTER, PAD COUNTER, AND ERROR COUNTER
                                ;*    CLEAR THE ERROR COUNTER
                                ;*    REQUEST A TU PORT FROM THE HOST
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    IF NO UNITS SELECTED
                                ;*    :    THEN-ERROR-ABORT TEST
                                ;*    :    ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    SET LOOP COMMAND/STATUS ON SELECTED PORT
                                ;*    CLEAR THE COMMAND BYTE FOR SELECTED PORT
                                ;*    SET WRITE PATH ENABLE
                                ;*    INITIALIZE THE EXPECTED RESULTS TABLE POINTER
                                ;*    SET WMC/XMC RESTART
                                ;*    SELECT THE SINGLE STEP CLOCK
                                ;*    CLEAR WMC/XMC RESTART
                                ;*    LOAD THE BYTE COUNT WITH 8 (BITS 7:0)
                                ;*    CLEAR THE BYTE COUNT (BITS 15:8)
                                ;*    SET DDR CONTROL TO "IN"
                                ;*    SELECT THE IMAGE MODE OF DATA FORMATTING
                                ;*    LOAD THE RESIDUAL CHARACTER WITH 6
                                ;*    LOAD THE PAD COUNT WITH 6 (BITS 7:0)
                                ;*    CLEAR THE PAD COUNT (BITS 15:8)
                                ;*    ENABLE ALL TRACKS LOAD TRACK ENABLE WITH ONES
                                ;*    SET THE CLOCK COUNT TO 180(10)
                                ;*    BGNDO
                                ;*    :    CALL SUBROUTINE CLKSYS
                                ;*    :    DECREMENT THE CLOCK COUNT
                                ;*    :    DO UNTIL CLOCK COUNT=0
                                ;*    ENDDO
                                ;*    SET WRITE MICRO TO WRITE MICRO ENABLE, TRANSLATOR ENABLE, GCR, AND WRITE
                                ;*    SET THE CLOCK COUNT TO 256(10)
                                ;*    BGNDO
```

```
1377                    ;*      :    CALL SUBROUTINE CLKSYS
1378                    ;*      :    DECREMENT THE CLOCK COUNT
1379                    ;*      :    DO UNTIL CLOCK COUNT=0 OR XL WRITE CLOCK=1
1380                    ;*    ENDDO
1381                    ;*    IF CLOCK COUNT=0
1382                    ;*      :    THEN-FATAL ERROR-ABORT TEST
1383                    ;*      :    ELSE-CONTINUE
1384                    ;*    ENDIF
1385                    ;*    BGNDO
1386                    ;*      :    CALL SUBROUTINE CLKSYS
1387                    ;*      :    CALL SUBROUTINE CLKSYS
1388                    ;*      :    INPUT THE TRANSLATOR OUTPUT DATA
1389                    ;*      :    IF THE ACTUAL TRANSLATOR OUTPUT=EXPECTED TRANSLATOR OUTPUT
1390                    ;*      :    :    THEN-CONTINUE
1391                    ;*      :    :    ELSE-ERROR
1392                    ;*      :    ENDIF
1393                    ;*      :    INPUT THE TRANSLATOR OUTPUT DATA PARITY
1394                    ;*      :    IF DATA OUTPUT DURING DATA GROUP
1395                    ;*      :    :    THEN-EXPECTED PARITY=ODD PARITY FOR OUTPUT DATA
1396                    ;*      :    :    ELSE-EXPECTED PARITY=1
1397                    ;*      :    ENDIF
1398                    ;*      :    IF INPUT PARITY=EXPECTED PARITY
1399                    ;*      :    :    THEN-CONTINUE
1400                    ;*      :    :    ELSE-ERROR
1401                    ;*      :    ENDIF
1402                    ;*      :    SET THE LOOP COUNT TO 14(8)
1403                    ;*      :    BGNDO
1404                    ;*      :    :    CALL SUBROUTINE CLKSYS
1405                    ;*      :    :    DECREMENT THE LOOP COUNT
1406                    ;*      :    :    DO UNTIL THE LOOP COUNT = 0
1407                    ;*      :    ENDDO
1408                    ;*      :    IF XL WRITE CLOCK = 1
1409                    ;*      :    :    THEN - CONTINUE
1410                    ;*      :    :    ELSE - ERROR - IMPROPER TRANSITION RATE FOR GCR
1411                    ;*      :    ENDIF
1412                    ;*      :    INCREMENT THE EXPECTED RESULTS TABLE POINTER
1413                    ;*      :    DO UNTIL THE EXPECTED RESULTS POINTER=END OF TABLE
1414                    ;*    ENDDO
1415                    ;*    SET THE CLOCK COUNT TO 256(10)
1416                    ;*    BGNDO
1417                    ;*      :    CALL SUBROUTINE CLKSYS
1418                    ;*      :    DECREMENT THE CLOCK COUNT
1419                    ;*      :    DO UNTIL CLOCK COUNT=0 OR TRANSLATOR OR WRITE MICRO DONE=0
1420                    ;*    ENDDO
1421                    ;*    IF CLOCK COUNT=0
1422                    ;*      :    THEN-ERROR-TIMEOUT WAITING FOR WRITE MICRO DONE
1423                    ;*      :    ELSE-CONTINUE
1424                    ;*    ENDIF
1425                    ;*    IF TRANSLATOR DONE
1426                    ;*      :    THEN-ERROR-TRANSLATOR DONE BEFORE WRITE MICRO
1427                    ;*      :    ELSE-SET THE CLOCK COUNT TO 256(10)
1428                    ;*      :    BGNDO
1429                    ;*      :    :    CALL SUBROUTINE CLKSYS
1430                    ;*      :    :    DECREMENT THE CLOCK COUNT
```

```
                                ;*     :     :    DO UNTIL TRANSLATOR DONE OR CLOCK COUNT=0
                                ;*     :     ENDDO
                                ;*     :     IF CLOCK COUNT=0
                                ;*     :     :    THEN-ERROR-TIMEOUT WAITING FOR TRANSLATOR DONE
                                ;*     :     :    ELSE-CONTINUE
                                ;*     :     ENDIF
                                ;*    ENDIF
                                ;*ENDTST
 4300                           SE
  (1)                           ; ************************************************************************
  (1)                           ;*ERRORS
  (1)                           ;*------
                                ;*XMC3 MICRO TEST 01
                                ;*XMC3 MICRO ERROR 01
                                ;*XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
                                ;*M8958, M8959, M8955'S
                                ;*OPERATOR ERROR NO TM78 UNITS SELECTED
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*XMC3 MICRO TEST 01
                                ;*XMC3 MICRO ERROR 02
                                ;*XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
                                ;*M8958, M8959, M8955'S
                                ;*WRITE PATH TIMEOUT WHILE WAITING FOR "XL CLK" TO SET
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*XMC3 MICRO TEST 01
                                ;*XMC3 MICRO ERROR 03
                                ;*XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
                                ;*M8958, M8959, M8955'S
                                ;*DATA STATE RECEIVED FROM THE XMC INCORRECT
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*TRANSITION COUNT = LLL
                                ;*
                                ;*XMC3 MICRO TEST 01
                                ;*XMC3 MICRO ERROR 04
                                ;*XMC3-GCR-PREAMBLE/POSTAMP _ TEST-1 DATA GROUP
                                ;*M8958, M8959, M8955'S
                                ;*PARITY STATE RECEIVED FROM THE XMC INCORRECT
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*TRANSITION COUNT = LLL
                                ;*
                                ;*XMC3 MICRO TEST 01
                                ;*XMC3 MICRO ERROR 05
                                ;*XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
                                ;*M8958, M8959, M8955'S
                                ;*TRANSLATOR "XL WCLK" NOT SET AFTER EIGHT CLOCK CYCLES
                                ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
                                ;*TRANSITION COUNT = LLL
                                ;*
                                ;*XMC3 MICRO TEST 01
                                ;*XMC3 MICRO ERROR 06
```

```
                                        ;*XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
                                        ;*M8958, M8959, M8955'S
                                        ;*''WMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
                                        ;*
                                        ;*XMC3 MICRO TEST 01
                                        ;*XMC3 MICRO ERROR 07
                                        ;*XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
                                        ;*M8958, M8959, M8955'S
                                        ;*''XMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
                                        ;*
                                        ;*XMC3 MICRO TEST 01
                                        ;*XMC3 MICRO ERROR 10
                                        ;*XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
                                        ;*M8958, M8959, M8955'S
                                        ;*''XMC NOT DONE'' RESET BEFORE ''WMC NOT DONE'' RESET
1497  4300                              S
 (1)                                    ; ************************************************************************
1498
1499  4300                              TEST1:  TESTX   @1
 (1)  4300  3E  01          7.0                        MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28     18.0                        CALL    TSET            ;SETUP THE TEST
1500                                    ;%XMC3-GCR-PREAMBLE/POSTAMBLE TEST-1 DATA GROUP
1501                                    ;&M8958, M8959, M8955'S
1502
1503  4305  AF                4.0     T1LOOP: XRA     A               ;CLEAR UNIT MAP STORAGE
1504  4306  32  91  46       13.0             STA     EXDLPC          ;CLEAR THE ERROR COUNTER
1505  4309  3E  30            7.0             MVI     A,@60           ;INIT THE BYTE COUNTER
1506  430B  D3  D7           10.0             OUT     CNTCTL          ;TO NORMAL MODE
1507  430D  3E  70            7.0             MVI     A,@160          ;INIT THE PAD COUNTER
1508  430F  D3  D7           10.0             OUT     CNTCTL          ;TO NORMAL MODE
1509  4311  3E  B0            7.0             MVI     A,@260          ;INIT THE ERROR COUNTER
1510  4313  D3  D7           10.0             OUT     CNTCTL          ;TO NORMAL MODE
1511  4315                              CLRECT
 (1)  4315  AF                4.0                             XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  4316  D3  D6           10.0                             OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  4318  D3  D6           10.0                             OUT     ERRCNT          ;CLEAR BITS 15-8
1512
1513                                            ;REQUEST THE HOST FOR THE PORTS TO BE USED IN THIS TEST...VALUE WILL
1514                                            ;BE IN CAS REG 12 LOW BYTE ON RETURN FROM HOST REQUEST.
1515
1516  431A                                      REQ     @7,0,0,0
 (1)  431A  CD  06  28     18.0                             CALL    REQST
 (1)  431D  00                                              .BYTE   0               ;DATA PATTERN NUMBER
 (1)  431E  00  00                                          .WORD   0               ;SYSTEM ''0'' COUNT
 (1)  4320  00  00                                          .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  4322  00                                              .BYTE                   ;DATA COMPARE FLAG IF  1
 (1)  4323  07                                              .BYTE   @7              ;REQUEST CODE
1517  4324                                      RIN     R12L
 (1)  4324  DB  94           10.0                        IN      R12L                    ;READ R12L INTO AC
 (1)  4326  7F                4.0                        MOV     A,A                     ;RETRY LINK
1518  4327  32  8D  46       13.0             STA     UNITMP          ;SAVE THE UNIT MAP INFO FROM HOST
1519  432A  AF                4.0             XRA     A
1520  432B  CD  6C  45       18.0             CALL    CLEAR
1521  432E  06  00            7.0             MVI     B,0
```

```
1522  4330  3A 8D 46   13.0  T1LPA:  LDA   UNITMP          ;GET SELECTED UNIT MAP
1523  4333  E6 01       7.0         ANI   $01             ;UNIT 0 PRESENT?
1524  4335  C2 61 43   10.0         JNZ   FOUND           ;YES-USE IT
1525  4338  04          4.0         INR   B
1526  4339  3A 8D 46   13.0         LDA   UNITMP          ;
1527  433C  E6 02       7.0         ANI   $02             ;UNIT 1 PRESENT?
1528  433E  C2 61 43   10.0         JNZ   FOUND           ;YES-USE IT
1529  4341  04          4.0         INR   B
1530  4342  3A 8D 46   13.0         LDA   UNITMP          ;
1531  4345  E6 04       7.0         ANI   $04             ;UNIT 2 PRESENT?
1532  4347  C2 61 43   10.0         JNZ   FOUND           ;YES-USE IT
1533  434A  04          4.0         INR   B
1534  434B  3A 8D 46   13.0         LDA   UNITMP
1535  434E  E6 08       7.0         ANI   $08             ;UNIT 3 PRESENT
1536  4350  C2 61 43   10.0         JNZ   FOUND           ;YES-USE IT
1537                                                      ;ELSE - OPERATOR ERROR
1538  4353                          ERR   EXIT,DUMMY
 (1)                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4353  CD 09 28   18.0                 CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0001                            MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4356  01                              .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  4357  00                              .BYTE
 (1)  4358  CD 15 28   18.0          DUMMY:: CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  435B  DA 5D 45   10.0                 JC    EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1539                          ;>OPERATOR ERROR NO TM78 UNITS SELECTED
1540                          ;>FATAL ERROR - TEST ABORTED
1541  435E  C3 5D 45   10.0         JMP   EXIT
1542
1543                          ;SET UP FOR THE DATA TRANSFER
1544  4361  DB E0       10.0  FOUND:  IN    INTSTA
1545  4363  E6 80       7.0         ANI   BIT7
1546  4365  B0          4.0         ORA   B
1547  4366  D3 E0      10.0         OUT   MBSEL
1548  4368  3E 40       7.0         MVI   A,P.LCS         ;KILL THE MTA BOARD
1549  436A  D3 48      10.0         OUT   PDIAG
1550  436C  AF          4.0         XRA   A
1551  436D  D3 40      10.0         OUT   TCMD
1552  436F  3E 10       7.0         MVI   A,P.WPEN        ;SET WRITE PATH ENABLE
1553  4371  D3 4C      10.0         OUT   PENAB
1554  4373  21 97 45   10.0         LXI   H,GXPTBL        ;LOAD ADDRESS OF EXPECTED
1555  4376  22 92 46   16.0         SHLD  POINTER         ;RESULTS TABLE INTO POINTER
1556  4379  3E 01       7.0         MVI   A,W.RST         ;LOAD THE WMC/XMC RESTART CODE
1557  437B  D3 D3      10.0         OUT   WMCCTL          ;ISSUE THE COMMAND
1558  437D  3E 05       7.0         MVI   A,SSTEP         ;SELECT SINGLE STEP WRITE CLOCK
1559  437F  D3 F0      10.0         OUT   CLKCTL          ;INITIALIZE THE CLOCK
1560  4381  32 96 46   13.0         STA   CCTLWD          ;SAVE SOFTWARE IMAGE
1561  4384  CD 60 45   18.0         CALL  CLKSYS          ;CLOCK THE SYSTEM TO ENABLE
1562  4387  CD 60 45   18.0         CALL  CLKSYS          ;RESTART TO TAKE HOLD
1563  438A  AF          4.0         XRA   A               ;CLEAR THE ACC.
1564  438B  D3 D3      10.0         OUT   WMCCTL          ;CLEAR THE WMC/XMC RESTART BIT
1565  438D  3E 07       7.0         MVI   A,7             ;GET THE BYTE COUNT - 1 DATA GROUP
1566  438F  3C          4.0         INR   A               ;CORRECT FOR INTEL 8253 CHIP
1567  4390  D3 D4      10.0         OUT   BYTCNT          ;LOAD BITS 7-0
1568  4392  AF          4.0         XRA   A
```

```
1569  4393  D3  D4        10.0          OUT     BYTCNT          ;CLEAR THE BYTE COUNTER BITS 15-8
1570  4395  32  94  46    13.0          STA     TRANSCNT        ;INITIALIZE THE TRANSITION COUNT
1571  4398  32  95  46    13.0          STA     TRANSCNT+1
1572  439B  32  8F  46    13.0          STA     DPCNT           ;CLEAR THE PARITY TEST COUNTER
1573  439E  3E  88         7.0          MVI     A,@210
1574  43A0  D3  DB        10.0          OUT     DDRCTL          ;SET THE DDR CONTROL TO 'IN'
1575  43A2  3E  50         7.0          MVI     A,$50           ;LOAD THE IMAGE MODE COMMAND
1576  43A4  D3  D0        10.0          OUT     DATACTL         ;SET THE DATA CONTROL MODE
1577  43A6  3E  06         7.0          MVI     A,6             ;LOAD THE RESIDUAL CHARACTER VALUE
1578  43A8  D3  D1        10.0          OUT     RESCHR          ;SET THE RESIDUAL CHARACTER
1579  43AA  3E  06         7.0          MVI     A,6             ;LOAD PAD COUNT VALUE
1580  43AC  3C             4.0          INR     A               ;CORRECT FOR INTEL 8253 CHIP
1581  43AD  D3  D5        10.0          OUT     PADCNT          ;WRITE TO WMC BITS 7-0
1582  43AF  AF             4.0          XRA     A               ;LOAD HI VALUE
1583  43B0  D3  D5        10.0          OUT     PADCNT          ;WRITE TO WMC
1584  43B2  D3  D2        10.0          OUT     TRKENA          ;1ST CLEAR THE TRACK ENABLE BITS
1585  43B4  D3  D2        10.0          OUT     TRKENA
1586  43B6  3E  FF         7.0          MVI     A,$FF           ;LOAD ACC WITH ALL ONES
1587  43B8  D3  D2        10.0          OUT     TRKENA          ;ENABLE PARITY TRACK
1588  43BA  D3  D2        10.0          OUT     TRKENA          ;ENABLE DATA TRACKS
1589  43BC  06  B4         7.0          MVI     B,180           ;LOAD THE CLOCK COUNT
1590
1591  43BE  CD  60  45     18.0  CLKRS:  CALL    CLKSYS          ;CLOCK IT
1592  43C1  05             4.0          DCR     B               ;
1593  43C2  C2  BE  43     10.0          JNZ     CLKRS           ;CONTINUE WITH RESTART CLOCKING
1594
1595  43C5  3E  D8         7.0          MVI     A,W.ENAB.X.ENAB!W.GCR!W.WRITE
1596  43C7  D3  D3        10.0          OUT     WMCCTL          ;START THE OPERATION
1597  43C9  06  FF         7.0  CLK1:   MVI     B,$FF           ;SET TIMEOUT COUNT TO 256
1598                                                           ;
1599  43CB  CD  60  45     18.0  CLK:    CALL    CLKSYS          ;SINGLE STEP THE WRITE PATH
1600                                                           ;
1601  43CE  DB  D0        10.0          IN      WMCSTA          ;GET THE WMC STATUS
1602  43D0  E6  01         7.0          ANI     X.WCLK          ;X.WCLK SET?
1603  43D2  C2  E7  43     10.0          JNZ     BGNPRE          ;YES-BEGINNING OF PREAMBLE
1604  43D5  05             4.0          DCR     B               ;NO-DECREMENT CLOCK
1605  43D6  C2  CB  43     10.0          JNZ     CLK             ;CONTINUE UNTIL ZERO
1606  43D9                               ERR     TSTEND,ABOCNO
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  43D9  CD  09  28     18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                                 MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43DC  02                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43DD  00                                  .BYTE
 (1)  43DE  CD  15  28     18.0          ABOCNO::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43E1  DA  18  28     10.0                  JC      TSTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
1607                                    ;>WRITE PATH TIMEOUT WHILE WAITING FOR "XL CLK" TO SET
1608                                    ;>FATAL ERROR - TEST ABORTED
1609
1610  43E4  C3  18  28     10.0          JMP     TSTEND
1611
1612  43E7  CD  60  45     18.0  BGNPRE: CALL    CLKSYS          ;CLOCK THE SYSTEM
1613  43EA  CD  60  45     18.0          CALL    CLKSYS          ;CLOCK AGAIN
1614  43ED  DB  40        10.0          IN      TSTS            ;GET THE DATA READ
1615  43EF  47             4.0          MOV     B,A             ;SAVE IT IN B=ADATA
```

```
1616  43F0                                    ROUT    ADATA           ;STORE ACTUAL DATA IN CAS
 (1)  43F0  D3  94          10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43F2  7F               4.0                     MOV     A,A             ;RETRY LINK
1617                                                ;
1618  43F3  2A  94  46       16.0            LHLD    TRANSCNT        ;GET THE TRANSITION COUNT
1619  43F6  23                6.0            INX     H               ;UPDATE
1620  43F7  22  94  46       16.0            SHLD    TRANSCNT        ;RESTORE THE TRANSITION COUNT
1621  43FA  7D                4.0            MOV     A,L             ;GET TRANSCOUNT LOW
1622  43FB                                    ROUT    R05L            ;WRITE TO CAS
 (1)  43FB  D3  8A          10.0                     OUT     R05L            ;WRITE AC INTO R05L
 (1)  43FD  7F               4.0                     MOV     A,A             ;RETRY LINK
1623  43FE  7C                4.0            MOV     A,H             ;GET TRANSCOUNT HIGH
1624  43FF                                    ROUT    R05H            ;WRITE TO CAS
 (1)  43FF  D3  8B          10.0                     OUT     R05H            ;WRITE AC INTO R05H
 (1)  4401  7F               4.0                     MOV     A,A             ;RETRY LINK
1625  4402  2A  92  46       16.0            LHLD    POINTER         ;GET POINTER TO EXPECTED DATA
1626  4405  7E                7.0            MOV     A,M             ;GET EXPECTED DATA BYTE
1627  4406  32  8E  46       13.0            STA     XDATA           ;SAVE FOR LATER
1628  4409  B8                4.0            CMP     B               ;COMPARE WITH ACTUAL
1629  440A  CA  22  44        10.0            JZ      PRECN1          ;CONTINUE IF EQUAL
1630                                                                ;ELSE-ERROR
1631  440D                                    ROUT    EDATA           ;STORE EXPECTED DATA IN CAS
 (1)  440D  D3  95          10.0                     OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  440F  7F               4.0                     MOV     A,A             ;RETRY LINK
1632  4410  3A  91  46       13.0    XDLPC1: LDA     EXDLPC          ;GET THE CURRENT COUNT AT ERROR
1633  4413  A7                4.0            ANA     A
1634  4414  C2  1D  44        10.0            JNZ     XDLPC2          ;STAY AT THIS COUNT IF NOT ZERO
1635  4417  3A  94  46       13.0            LDA     TRANSCNT        ;GET THE COUNT AT THE ERROR
1636  441A  32  91  46       13.0            STA     EXDLPC          ;SAVE IT
1637  441D                            XDLPC2: ERRB    XDLOOP,PRECN1,@4
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  441D  CD  12  28       18.0                     CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0003                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4420  03                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4421  04                                        .BYTE   @4              ;PRINT ROUTINE NUMBER
 (1)  4422  CD  15  28       18.0            PRECN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4425  DA  2B  44       10.0                     JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1638                            ;>DATA STATE RECEIVED FROM THE XMC INCORRECT
1639  442B  C3  46  44        10.0            JMP     XDLPC           ;SKIP OVER ERROR LOOP CONTROL
1640
1641
1642
1643                            ;HERE IS THE ERROR LOOP CONTROL
1644
1645  442B  3A  94  4F        13.0    XDLOOP: LDA     LPFLG           ;GET THE LOOP ON ERROR FLAG
1646  442E  A7                4.0            ANA     A
1647  442F  C2  38  44        10.0            JNZ     XDLP1           ;JUMP IF FLAG IS SET
1648  4432  32  91  46       13.0            STA     EXDLPC          ;ELSE - CLEAR THE COUNTER
1649  4435  C3  46  44        10.0            JMP     XDLPC           ;AND CONTINUE TESTING
1650
1651  4438  3A  94  46       13.0    XDLP1:  LDA     TRANSCNT        ;GET THE LOW BYTE TRANSITION COUNT
1652  443B  47                4.0            MOV     B,A             ;TEMP SAVE
1653  443C  3A  91  46       13.0            LDA     EXDLPC          ;GET THE TRANS. COUNT AT ERROR
1654  443F  B8                4.0            CMP     B               ;SAME?
```

```
1655  4440  CA  05  43   10.0          JZ    T1LOOP          ;YUP - LOOP NOW.
1656  4443  C3  46  44   10.0          JMP   XDLPC           ;NO - CONTINUE TILL A MATCH...
1657
1658  4446  DB  48       10.0  XDLPC:  IN    PSTAT           ;GET THE PORT PARITY BIT
1659  4448  0F            4.0          RRC                   ;RIGHT JUSTIFY
1660  4449  0F            4.0          RRC                   ;
1661  444A  E6  01        7.0          ANI   $01             ;REMOVE OTHER BITS
1662  444C  57            4.0          MOV   D,A             ;STORE ACTUAL PARITY IN REG D
1663
1664  444D                             ROUT  ADATA           ;STORE ACTUAL PARITY IN CAS
 (1)  444D  D3  94        10.0                OUT   ADATA           ;WRITE AC INTO ADATA
 (1)  444F  7F            4.0                 MOV   A,A             ;RETRY LINK
1665  4450  2A  92  46   16.0          LHLD  POINTER
1666  4453  01  EC  45   10.0          LXI   B,GXPDAT        ;LOAD BEGINNING ADDRESS OF EXPECTED DATA
1667  4456  78            4.0          MOV   A,B             ;MAKE THE TWOS COMPLEMENT OF REGISTER
1668  4457  2F            4.0          CMA                   ;PAIR B & C
1669  4458  47            4.0          MOV   B,A
1670  4459  79            4.0          MOV   A,C
1671  445A  2F            4.0          CMA
1672  445B  4F            4.0          MOV   C,A
1673  445C  03            6.0          INX   B
1674  445D  09           10.0          DAD   B               ;ADD POINTER AND BEGINNING OF DATA TABLE
1675  445E  D2  85  44   10.0          JNC   PPREAM          ;CHECK PREAMBLE PARITY TRACK
1676
1677  4461  2A  92  46   16.0          LHLD  POINTER         ;RESET H & L REGS
1678  4464  01  14  46   10.0          LXI   B,GDATEN        ;GET THE ENDING ADDRESS OF EXPECTED DATA
1679  4467  78            4.0          MOV   A,B             ;MAKE THE TWOS COMPLEMENT OF REGISTER
1680  4468  2F            4.0          CMA                   ;PAIR B & C
1681  4469  47            4.0          MOV   B,A
1682  446A  79            4.0          MOV   A,C
1683  446B  2F            4.0          CMA
1684  446C  4F            4.0          MOV   C,A
1685  446D  03            6.0          INX   B
1686  446E  09           10.0          DAD   B               ;ADD POINTER AND END OF DATA TABLE
1687  446F  DA  85  44   10.0          JC    PPOST           ;CHECK POSTAMBLE PARITY TRACK
1688
1689  4472  01  64  46   10.0  DPTST:  LXI   B,PXPTBL       ;POINT TO EXPECTED PARITY DATA TABLE
1690  4475  2A  8F  46   16.0          LHLD  DPCNT           ;POINT TO THE TABLE OFFSET NUMBER
1691  4478  23            6.0          INX   H               ;UPDATE IT
1692  4479  22  8F  46   16.0          SHLD  DPCNT           ;SAVE NEW COUNT
1693  447C  09           10.0          DAD   B               ;MAKE REAL POINTER
1694  447D  7E            7.0          MOV   A,M             ;GET EXPECTED PARITY STATE
1695  447E  A7            4.0          ANA   A               ;SET CONDITION BITS
1696  447F  FA  C8  44   10.0          JM    PTYC            ;JMP IF NO COMPARE ON THIS DATA
1697  4482  C3  8A  44   10.0          JMP   PTY             ;TEST THE PARITY BIT
1698
1699                       ;HERE TO CHECK THE PREAMBLE/POSTAMBLE PARITY TRACK BIT
1700
1701  4485                 PPOST:
1702  4485  3A  8E  46   13.0  PPREAM: LDA   XDATA           ;GET THE DATA EXPECTED
1703  4488  E6  01        7.0          ANI   1               ;PARITY TRACK IS SAME
1704
1705  448A  BA            4.0  PTY:    CMP   D               ;SEE IF EXPECTED-ACTUAL
1706  448B  CA  A7  44   10.0          JZ    PTSTOK          ;JMP IF OK
```

```
1707  448E                                    ROUT    EDATA           ;SAVE THE EXPECTED DATA
 (1)  448E  D3  95          10.0                     OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4490  7F               4.0                     MOV     A,A             ;RETRY LINK
1708  4491  7A               4.0             MOV     A,D             ;ERROR - GET THE ACTUAL DATA
1709  4492                                    ROUT    ADATA           ;SAVE FOR PRINTOUT
 (1)  4492  D3  94          10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4494  7F               4.0                     MOV     A,A             ;RETRY LINK
1710
1711  4495  3A  91  46       13.0  PTY1:     LDA     EXDLPC          ;GET THE COUNT AT THE ERROR
1712  4498  A7               4.0             ANA     A
1713  4499  C2  A2  44       10.0             JNZ     PTY2            ;JUMP IF AT AN ERROR NOW
1714  449C  3A  94  46       13.0             LDA     TRANSCNT        ;GET THE COUNT AT ERROR
1715  449F  32  91  46       13.0             STA     EXDLPC          ;SAVE IT
1716  44A2                            PTY2:   ERRB    XPLOOP,PTSTOK,@4
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44A2  CD  12  28       18.0                     CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0004                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44A5  04                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44A6  04                                       .BYTE   @4              ;PRINT ROUTINE NUMBER
 (1)  44A7  CD  15  28       18.0             PTSTOK::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44AA  DA  B0  44       10.0                     JC      XPLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1717                                    ;>PARITY STATE RECEIVED FROM THE XMC INCORRECT
1718  44AD  C3  C8  44       10.0             JMP     PTYC            ;JMP OVER CODE
1719
1720  44B0  3A  94  4F       13.0  XPLOOP: LDA     LPFLG           ;GET THE LOOP ON ERROR FLAG
1721  44B3  A7               4.0             ANA     A
1722  44B4  C2  BD  44       10.0             JNZ     XPLP1           ;JUMP IF FLAG IS SET
1723  44B7  32  91  46       13.0             STA     EXDLPC          ;ELSE - CLEAR THE COUNTER
1724  44BA  C3  C8  44       10.0             JMP     PTYC            ;AND CONTINUE TESTING
1725
1726  44BD  3A  94  46       13.0  XPLP1:  LDA     TRANSCNT        ;GET THE LOW BYTE TRANSITION COUNT
1727  44C0  47               4.0             MOV     B,A             ;TEMP SAVE
1728  44C1  3A  91  46       13.0             LDA     EXDLPC          ;GET THE TRANS. COUNT AT ERROR
1729  44C4  B8               4.0             CMP     B               ;SAME?
1730  44C5  CA  05  43       10.0             JZ      T1LOOP          ;YUP - LOOP NOW!
1731
1732  44C8  2A  92  46       16.0  PTYC:   LHLD    POINTER         ;NO-UPDATE THE
1733  44CB  23               6.0             INX     H               ;TABLE POINTER
1734  44CC  22  92  46       16.0             SHLD    POINTER         ;ADDRESS
1735
1736  44CF  06  0E            7.0             MVI     B,14            ;LOAD THE CLOCK COUNT
1737  44D1  CD  60  45       18.0  1$:     CALL    CLKSYS          ;CLOCK THE WRITE PATH
1738  44D4  05               4.0             DCR     B               ;DECREMENT THE COUNT
1739  44D5  C2  D1  44       10.0             JNZ     1$              ;CONTINUE UNTIL 8 CLOCKS
1740  44D8  DB  D0           10.0             IN      WMCSTA
1741  44DA  E6  01            7.0             ANI     X.WCLK
1742  44DC  C2  E4  44       10.0             JNZ     PRECN3
```

```
1744  44DF                                    ERR     T1LOOP,PRECN3,@4
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44DF  CD  09  28      18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0005                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44E2  05                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44E3  04                                        .BYTE   @4
 (1)  44E4  CD  15  28      18.0               PRECN3::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44E7  DA  05  43      10.0                       JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1745                                    ;>TRANSLATOR "XL WCLK" NOT SET AFTER EIGHT CLOCK CYCLES
1746                                    ;>IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
1747
1748  44EA  2A  92  46      16.0          LHLD    POINTER         ;GET THE TABLE POINTER
1749  44ED  01  63  46      10.0          LXI     B,GTBLEN        ;GET END OF TABLE ADDRESS
1750  44F0  78              4.0           MOV     A,B             ;FROM THE
1751  44F1  2F              4.0           CMA                     ;TWO'S COMPLIMENT
1752  44F2  47              4.0           MOV     B,A             ;OF THE TABLE END
1753  44F3  79              4.0           MOV     A,C             ;ADDRESS
1754  44F4  2F              4.0           CMA
1755  44F5  4F              4.0           MOV     C,A
1756  44F6  03              6.0           INX     B
1757  44F7  09              10.0          DAD     B               ;END OF TABLE?
1758  44F8  DA  FE  44      10.0          JC      RECE'.D         ;YES
1759  44FB  C3  E7  43      10.0          JMP     BGNPRE
1760
1761  44FE  06  FF           7.0  RECEND: MVI     B,$FF           ;LOAD THE TIMEOUT COUNT
1762  4500  CD  60  45      18.0  1$:     CALL    CLKSYS          ;CLOCK THE SYSTEM
1763  4503  DB  D0          10.0          IN      WMCSTA          ;INPUT THE WMC STATUS
1764  4505  E6  40           7.0          ANI     W.DONN          ;WMC DONE?
1765  4507  CA  1A  45      10.0          JZ      RECCN1          ;YES-EXIT
1766  450A  DB  D0          10.0          IN      WMCSTA          ;GET WMC STATUS
1767  450C  E6  80           7.0          ANI     X.DONN          ;XMC DONE
1768  450E  CA  3E  45      10.0          JZ      SEQERR          ;YES-ERROR
1769  4511  05              4.0          DCR     B               ;NO-DECREMENT THE COUNT
1770  4512  C2  00  45      10.0          JNZ     1$              ;CONTINUE IF NOT ZERO
1771  4515                                    ERR     RECEND,RECCN1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4515  CD  09  28      18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0006                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4518  06                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4519  00                                        .BYTE
 (1)  451A  CD  15  28      18.0               RECCN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  451D  DA  FE  44      10.0                       JC      RECEND          ;LOOP ADDRESS IF LOOP SPECIFIED
1772                                    ;>'WMC NOT DONE" TIMEOUT WAITING FOR IT TO GO TO ZERO
1773
1774  4520  06  FF           7.0          MVI     B,$FF           ;LOAD THE TIMEOUT COUNT
1775  4522  CD  60  45      18.0  3$:     CALL    CLKSYS          ;CLOCK THE SYSTEM
1776  4525  DB  D0          10.0          IN      WMCSTA          ;INPUT THE WMC STATUS
1777  4527  E6  80           7.0          ANI     X.DONN          ;XMC-DONE?
1778  4529  CA  35  45      10.0          JZ      RECCN3          ;YES-EXIT
1779  452C  05              4.0          DCR     B               ;NO-DECREMENT THE COUNT
1780  452D  C2  22  45      10.0          JNZ     3$              ;CONTINUE IF NOT ZERO
```

```
1782  4530                                    ERR     T1LOOP,RECCN3
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4530  CD  09  28      18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4533  07                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4534  00                                       .BYTE
 (1)  4535  CD  15  28      18.0              RECCN3::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4538  DA  05  43      10.0                      JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1783                                    ;>''XMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
1784
1785  453B  C3  49  45      10.0              JMP     XMEND
1786
1787  453E                              SEQERR: ERR     T1LOOP,RECCN2
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  453E  CD  09  28      18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0008                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4541  08                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4542  00                                       .BYTE
 (1)  4543  CD  15  28      18.0              RECCN2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4546  DA  05  43      10.0                      JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1788                                    ;>''XMC NOT DONE'' RESET BEFORE ''WMC NOT DONE'' RESET
1789
1790  4549                              XMEND:  ENDTST  T1LOOP
 (1)                                          ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4549                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4549  CD  06  28      18.0                      CALL    REQST
 (2)  454C  00                                       .BYTE                   ;DATA PATTERN NUMBER
 (2)  454D  00  00                                   .WORD                   ;SYSTEM '"' COUNT
 (2)  454F  00  00                                   .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4551  00                                       .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4552  07                                       .BYTE   7               ;REQUEST CODE
 (1)  4553  3A  9A  4F      13.0              LDA     ITERA           ;GET ITERATION COUNT
 (1)  4556  3D               4.0              DCR     A               ;DOWNCOUNT
 (1)  4557  32  9A  4F      13.0              STA     ITERA           ;SAVE COUNT
 (1)  455A  F2  05  43      10.0              JP      T1LOOP          ;DO TEST UNTIL TILL = 0
1791
1792  455D  C3  18  28      10.0  EXIT:       JMP     TSTEND
1793
```

```
                                        .SBTTL  SUBROUTINE CLKSYS

                                        SD
                                        ; ************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE WILL GENERATE ONE SINGLE STEP USING THE CONTENTS OF
                                        ;*CCTLWD AS THE CLOCK CONTROL BYTE.
                                        S
                                        ; ************************************************************************

4560 3A 96 46          13.0  CLKSYS: LDA     CCTLWD          ;GET THE SOFTWARE CLOCK CONTROL IMAGE
4563 F6 40              7.0          ORI     SSCLK           ;ADD IN 'CLK' BIT
4565 D3 F0             10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL
4567 E6 [illegible]     7.0          ANI     @277            ;STRIP OFF CLOCK BIT
4569 D3 [illegible]    10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL WORD
456B C9                10.0          RET                     ;EXIT-CLOCK CYCLE
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ; ************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ; ************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ; ************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*    CLEAR PORT SELECT FOR TRANSPORT
                                        ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*    CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ; ************************************************************************
456C F5                  12.0  CLEAR:   PUSH    PSW             ;SAVE THE SELECTED PORT #
456D C5                  12.0           PUSH    B
456E 06 00                7.0           MVI     B,0             ;START TO CLEAR AT PORT #0
4570 DB E0               10.0  CLRLP:   IN      INTSTA          ;GET MB SELECT INFO
4572 E6 80                7.0           ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
4574 B0                   4.0           ORA     B               ;ADD IN THE SELECTED PORT #
4575 D3 E0               10.0           OUT     MBSEL           ;RESET TO THIS PORT
4577 3E 80                7.0           MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
4579 D3 40               10.0           OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
457B AF                   4.0           XRA     A               ;CLEAR TU COMMAND A
457C D3 40               10.0           OUT     TCMD            ;
457E 3E 81                7.0           MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
4580 D3 40               10.0           OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
4582 3E 00                7.0           MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
4584 D3 40               10.0           OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
4586 AF                   4.0           XRA     A
4587 D3 44               10.0           OUT     TAMT            ;CLEAR AMTIE WORD
4589 D3 48               10.0           OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
458B D3 4C               10.0           OUT     PENAB           ;CLEAR PORT ENABLE WORD
458D 04                   4.0           INR     B               ;POINT TO THE NEXT PORT TO CLEAR
458E 78                   4.0           MOV     A,B
458F FE 04                7.0           CPI     4               ;DONE?
4591 C2 70 45            10.0           JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
4594 C1                  10.0           POP     B               ;RESET B & C
4595 F1                  10.0           POP     PSW             ;ALL DONE
4596 C9                  10.0           RET                     ;EXIT
```

```
1853  4597  FF                GXPTBL: .BYTE   $FF             ;PREAMBLE TERM GROUP (10101)
1854                                                          ;CHARACTER 1 = ALL ONES
1855  4598  FF                        .BYTE   $FF             ;PREAMBLE TERM GROUP
1856                                                          ;CHARACTER 2 = ALL ZEROS
1857  4599  00                        .BYTE   $00             ;PREAMBLE TERM GROUP
1858                                                          ;CHARACTER 3 = ALL ONES
1859  459A  00                        .BYTE   $00             ;PREAMBLE TERM GROUP
1860                                                          ;CHARACTER 4 = ALL ZEROS
1861  459B  FF                        .BYTE   $FF             ;PREAMBLE TERM GROUP
1862                                                          ;CHARACTER 5 = ALL ONES
1863
1864  459C  FF                        .BYTE   $FF             ;PREAMBLE SECOND GROUP (01111)
1865                                                          ;CHARACTER 1 = ALL ZEROS
1866  459D  00                        .BYTE   $00             ;PREAMBLE SECOND GROUP
1867                                                          ;CHARACTER 2 = ALL ONES
1868  459E  FF                        .BYTE   $FF             ;PREAMBLE SECOND GROUP
1869                                                          ;CHARACTER 3 = ALL ONES
1870  459F  00                        .BYTE   $00             ;PREAMBLE SECOND GROUP
1871                                                          ;CHARACTER 4 = ALL ONES
1872  45A0  FF                        .BYTE   $FF             ;PREAMBLE SECOND GROUP
1873                                                          ;CHARACTER 5 = ALL ONES
1874                                                          ;
1875  45A1  00                        .BYTE   $00             ;PREAMBLE SYNC GROUP 1 (14 GRPS OF 11111)
1876                                                          ;CHARACTER 1 = ALL ONES
1877  45A2  FF                        .BYTE   $FF             ;PREAMBLE SYNC GROUP 1
1878                                                          ;CHARACTER 2 = ALL ONES
1879  45A3  00                        .BYTE   $00             ;PREAMBLE SYNC GROUP 1
1880                                                          ;CHARACTER 3 = ALL ONES
1881  45A4  FF                        .BYTE   $FF             ;PREAMBLE SYNC GROUP 1
1882                                                          ;CHARACTER 4 - ALL ONES
1883  45A5  00                        .BYTE   $00             ;PREAMBLE SYNC GROUP 1
1884                                                          ;CHARACTER 5 = ALL ONES
1885                                                          ;
1886  45A6  FF                  .BYTE   $FF                   ;PREAMBLE SYNC GROUP 2
1887                                                          ;CHARACTER 1 = ALL ONES
1888  45A7  00                  .BYTE   $00                   ;PREAMBLE SYNC GROUP 2
1889                                                          ;CHARACTER 2 = ALL ONES
1890  45A8  FF                  .BYTE   $FF                   ;PREAMBLE SYNC GROUP 2
1891                                                          ;CHARACTER 3 = ALL ONES
1892  45A9  00                  .BYTE   $00                   ;PREAMBLE SYNC GROUP 2
1893                                                          ;CHARACTER 4 = ALL ONES
1894  45AA  FF                  .BYTE   $FF                   ;PREAMBLE SYNC GROUP 2
1895                                                          ;CHARACTER 5 = ALL ONES
1896                                                          ;
1897  45AB  00                  .BYTE   $00                   ;PREAMBLE SYNC GROUP 3
1898                                                          ;CHARACTER 1 = ALL ONES
1899  45AC  FF                  .BYTE   $FF                   ;PREAMBLE SYNC GROUP 3
1900                                                          ;CHARACTER 2 = ALL ONES
1901  45AD  00                  .BYTE   $00                   ;PREAMBLE SYNC GROUP 3
1902                                                          ;CHARACTER 3 = ALL ONES
1903  45AE  FF                  .BYTE   $FF                   ;PREAMBLE SYNC GROUP 3
1904                                                          ;CHARACTER 4 - ALL ONES
1905  45AF  00                  .BYTE   $00                   ;PREAMBLE SYNC GROUP 3
1906                                                          ;CHARACTER 5 = ALL ONES
```

```
1907
1908  45B0  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 4
1909                                    ;CHARACTER 1 = ALL ONES
1910  45B1  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 4
1911                                    ;CHARACTER 2 = ALL ONES
1912  45B2  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 4
1913                                    ;CHARACTER 3 = ALL ONES
1914  45B3  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 4
1915                                    ;CHARACTER 4 = ALL ONES
1916  45B4  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 4
1917                                    ;CHARACTER 5 = ALL ONES
1918                                    ;
1919  45B5  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 5
1920                                    ;CHARACTER 1 = ALL ONES
1921  45B6  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 5
1922                                    ;CHARACTER 2 = ALL ONES
1923  45B7  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 5
1924                                    ;CHARACTER 3 = ALL ONES
1925  45B8  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 5
1926                                    ;CHARACTER 4 - ALL ONES
1927  45B9  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 5
1928                                    ;CHARACTER 5 = ALL ONES
1929                                    ;
1930  45BA  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 6
1931                                    ;CHARACTER 1 = ALL ONES
1932  45BB  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 6
1933                                    ;CHARACTER 2 = ALL ONES
1934  45BC  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 6
1935                                    ;CHARACTER 3 = ALL ONES
1936  45BD  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 6
1937                                    ;CHARACTER 4 = ALL ONES
1938  45BE  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 6
1939                                    ;CHARACTER 5 = ALL ONES
1940                                    ;
1941  45BF  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 7
1942                                    ;CHARACTER 1 = ALL ONES
1943  45C0  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 7
1944                                    ;CHARACTER 2 = ALL ONES
1945  45C1  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 7
1946                                    ;CHARACTER 3   ALL ONES
1947  45C2  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 7
1948                                    ;CHARACTER 4 - ALL ONES
1949  45C3  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 7
1950                                    ;CHARACTER 5 = ALL ONES
1951                                    ;
1952  45C4  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 8
1953                                    ;CHARACTER 1 = ALL ONES
1954  45C5  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 8
1955                                    ;CHARACTER 2 = ALL ONES
1956  45C6  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 8
1957                                    ;CHARACTER 3 = ALL ONES
1958  45C7  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 8
1959                                    ;CHARACTER 4 = ALL ONES
1960  45C8  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 8
```

```
1961                                                    ;CHARACTER 5 = ALL ONES
1962                                                    ;
1963  45C9  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 9
1964                                                    ;CHARACTER 1 = ALL ONES
1965  45CA  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 9
1966                                                    ;CHARACTER 2 = ALL ONES
1967  45CB  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 9
1968                                                    ;CHARACTER 3 = ALL ONES
1969  45CC  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 9
1970                                                    ;CHARACTER 4 - ALL ONES
1971  45CD  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 9
1972                                                    ;CHARACTER 5 = ALL ONES
1973                                                    ;
1974  45CE  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 10
1975                                                    ;CHARACTER 1 = ALL ONES
1976  45CF  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 10
1977                                                    ;CHARACTER 2 = ALL ONES
1978  45D0  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 10
1979                                                    ;CHARACTER 3 = ALL ONES
1980  45D1  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 10
1981                                                    ;CHARACTER 4 = ALL ONES
1982  45D2  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 10
1983                                                    ;CHARACTER 5 = ALL ONES
1984                                                    ;
1985  45D3  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 11
1986                                                    ;CHARACTER 1 = ALL ONES
1987  45D4  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 11
1988                                                    ;CHARACTER 2 = ALL ONES
1989  45D5  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 11
1990                                                    ;CHARACTER 3 = ALL ONES
1991  45D6  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 11
1992                                                    ;CHARACTER 4 - ALL ONES
1993  45D7  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 11
1994                                                    ;CHARACTER 5 = ALL ONES
1995                                                    ;
1996  45D8  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 12
1997                                                    ;CHARACTER 1 = ALL ONES
1998  45D9  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 12
1999                                                    ;CHARACTER 2 = ALL ONES
2000  45DA  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 12
2001                                                    ;CHARACTER 3 = ALL ONES
2002  45DB  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 12
2003                                                    ;CHARACTER 4 = ALL ONES
2004  45DC  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 12
2005                                                    ;CHARACTER 5 = ALL ONES
2006                                                    ;
2007  45DD  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 13
2008                                                    ;CHARACTER 1 = ALL ONES
2009  45DE  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 13
2010                                                    ;CHARACTER 2 = ALL ONES
2011  45DF  00                  .BYTE   $00            ;PREAMBLE SYNC GROUP 13
2012                                                    ;CHARACTER 3 = ALL ONES
2013  45E0  FF                  .BYTE   $FF            ;PREAMBLE SYNC GROUP 13
2014                                                    ;CHARACTER 4 - ALL ONES
```

```
2015  45E1  00                        .BYTE   $00            ;PREAMBLE SYNC GROUP 13
2016                                                        ;CHARACTER 5 = ALL ONES
2017                                                        ;
2018  45E2  FF                        .BYTE   $FF            ;PREAMBLE SYNC GROUP 14
2019                                                        ;CHARACTER 1 = ALL ONES
2020  45E3  00                        .BYTE   $00            ;PREAMBLE SYNC GROUP 14
2021                                                        ;CHARACTER 2 = ALL ONES
2022  45E4  FF                        .BYTE   $FF            ;PREAMBLE SYNC GROUP 14
2023                                                        ;CHARACTER 3 = ALL ONES
2024  45E5  00                        .BYTE   $00            ;PREAMBLE SYNC GROUP 14
2025                                                        ;CHARACTER 4 = ALL ONES
2026  45E6  FF                        .BYTE   $FF            ;PREAMBLE SYNC GROUP 14
2027                                                        ;CHARACTER 5 = ALL ONES
2028                                                        ;
2029  45E7  FF                        .BYTE   $FF            ;MARK ONE GROUP
2030                                                        ;CHARACTER 1 = ALL ZEROS
2031  45E8  FF                        .BYTE   $FF            ;MARK ONE GROUP
2032                                                        ;CHARACTER 2 = ALL ZEROS
2033  45E9  00                        .BYTE   $00            ;MARK ONE GROUP
2034                                                        ;CHARACTER 3 = ALL ONES
2035  45EA  FF                        .BYTE   $FF            ;MARK ONE GROUP
2036                                                        ;CHARACTER 4 = ALL ONES
2037  45EB  00                        .BYTE   $00            ;MARK ONE GROUP
2038                                                        ;CHARACTER 5 = ALL ONES
2039                                                        ;
2040                          ;HERE IS THE DATA GROUP - 7 BYTES OF ALL 0'S (BYTE #86.)
2041                          ;ECC FOR DATA GRP = 0.
2042
2043  45EC  FF                GXPDAT: .BYTE   $FF            ;1ST CHAR IN 1ST HALF OF DATA GROUP - P - 0
2044  45ED  00                        .BYTE   0              ;2ND CHAR - P = 1
2045  45EE  00                        .BYTE   0              ;3RD CHAR - P = 1
2046  45EF  00                        .BYTE   0              ;4TH CHAR - P - 1
2047  45F0  FF                        .BYTE   $FF            ;5TH CHAR - P = 1
2048
2049  45F1  00                        .BYTE   0              ;1ST CHAR IN 2ND HALF OF DATA GROUP - P   0
2050  45F2  FF                        .BYTE   $FF            ;2ND CHAR - P - 1
2051  45F3  FF                        .BYTE   $FF            ;3RD CHAR - P = 1
2052  45F4  FF                        .BYTE   $FF            ;4TH CHAR - P = 1
2053  45F5  00                        .BYTE   0              ;5TH CHAR (ECC CHAR =0) - P - 1
2054
2055                          ;END MARK = 11111 (ALL 1'S)
2056
2057  45F6  FF                        .BYTE   $FF            ;END MARK GROUP
2058                                                        ;CHARACTER 1 = ALL ONES
2059  45F7  00                        .BYTE   $00            ;END MARK GROUP
2060                                                        ;CHARACTER 2 = ALL ONES
2061  45F8  FF                        .BYTE   $FF            ;END MARK GROUP
2062                                                        ;CHARACTER 3 = ALL ONES
2063  45F9  00                        .BYTE   $00            ;END MARK GROUP
2064                                                        ;CHARACTER 4 = ALL ONES
2065  45FA  FF                        .BYTE   $FF            ;END MARK GROUP
2066                                                        ;CHARACTER 5 = ALL ONES
2067                                                        ;
2068
```

```
                                ;4 PAD BYTES
45FB  00                              .BYTE   @0              ;RESIDUAL GROUP - AFTER TRANSLATION - 1ST HALF
                                                              ;CHARACTER 1
45FC  FF                              .BYTE   $FF             ;RESIDUAL GROUP - AFTER TRANSLATION
                                                              ;CHARACTER 2
45FD  FF                              .BYTE   $FF             ;RESIDUAL GROUP - AFTER TRANSLATION
                                                              ;CHARACTER 3
45FE  FF                              .BYTE   $FF             ;RESIDUAL GROUP - AFTER TRANSLATION
                                                              ;CHARACTER 4
45FF  00                              .BYTE   @0              ;RESIDUAL GROUP - AFTER TRANSLATION
                                                              ;CHARACTER 5
                                ;2 PAD BYTES
                                ;ACRC = 323
                                ;ECC = 225
4600  FF                              .BYTE   @377            ;RESIDUAL GROUP - AFTER TRANSLATION - 2ND HALF
                                                              ;CHARACTER 1
4601  D3                              .BYTE   @323            ;RESIDUAL GROUP - AFTER TRANSLATION
                                                              ;CHARACTER 2
4602  D3                              .BYTE   @323            ;RESIDUAL GROUP - AFTER TRANSLATION
                                                              ;CHARACTER 3
4603  04                              .BYTE   @004            ;RESIDUAL GROUP - AFTER TRANSLATION
                                                              ;CHARACTER 4
4604  B9                              .BYTE   @271            ;RESIDUAL GROUP - AFTER TRANSLATION
                                                              ;CHARACTER 5
                                ;4 CRC BYTES OF 136
4605  18                              .BYTE   @030            ;CRC GROUP - AFTER TRANSLATION - 1ST HALF
                                                              ;CHARACTER 1
4606  E7                              .BYTE   @347            ;CRC GROUP - AFTER TRANSLATION
                                                              ;CHARACTER 2
4607  B9                              .BYTE   @271            ;CRC GROUP - AFTER TRANSLATION
                                                              ;CHARACTER 3
4608  E7                              .BYTE   @347            ;CRC GROUP - AFTER TRANSLATION
                                                              ;CHARACTER 4
4609  18                              .BYTE   @030            ;CRC GROUP - AFTER TRANSLATION
                                                              ;CHARACTER 5
                                ;2 CRC OF 136
                                ;RESID OF 006
                                ;ECC OF 352
460A  A9                              .BYTE   @251            ;CRC GROUP - AFTER TRANSLATION - 2ND HALF
                                                              ;CHARACTER 6
460B  56                              .BYTE   @126            ;CRC GROUP - AFTER TRANSLATION
                                                              ;CHARACTER 7
460C  08                              .BYTE   @010            ;CRC GROUP - AFTER TRANSLATION
                                                              ;CHARACTER 8
460D  BE                              .BYTE   @276            ;CRC GROUP - AFTER TRANSLATION
                                                              ;CHARACTER 9
460E  55                              .BYTE   @125            ;CRC GROUP - AFTER TRANSLATION
                                                              ;CHARACTER 10
                                                              ;


460F  AA                              .BYTE   @252            ;MARK TWO
```

```
2123
2124  4610  55                          .BYTE   @125        ;CHARACTER 1 = ALL ONES
2125                                                        ;MARK TWO
2126  4611  AA                          .BYTE   @252        ;CHARACTER 2 = ALL ONES
2127                                                        ;MARK TWO
2128  4612  AA                          .BYTE   @252        ;CHARACTER 3 = ALL ONES
2129                                                        ;MARK TWO
2130  4613  AA                          .BYTE   @252        ;CHARACTER 4 = ALL ZEROS
2131                                                        ;MARK TWO
2132                                                        ;CHARACTER 5 = ALL ZEROS
2133  4614  55                GDATEN:   .BYTE   @125        ;
2134                                                        ;POSTAMBLE SYNC GROUP 1
2135  4615  AA                          .BYTE   @252        ;CHARACTER 1 = ALL ONES
2136                                                        ;POSTAMBLE SYNC GROUP 1
2137  4616  55                          .BYTE   @125        ;CHARACTER 2 - ALL ONES
2138                                                        ;POSTAMBLE SYNC GROUP 1
2139  4617  AA                          .BYTE   @252        ;CHARACTER 3 = ALL ONES
2140                                                        ;POSTAMBLE SYNC GROUP 1
2141  4618  55                          .BYTE   @125        ;CHARACTER 4 = ALL ONES
2142                                                        ;POSTAMBLE SYNC GROUP 1
2143                                                        ;CHARACTER 5 = ALL ONES
2144  4619  AA                          .BYTE   @252        ;
2145                                                        ;POSTAMBLE SYNC GROUP 2
2146  461A  55                          .BYTE   @125        ;CHARACTER 1 = ALL ONES
2147                                                        ;POSTAMBLE SYNC GROUP 2
2148  461B  AA                          .BYTE   @252        ;CHARACTER 2 - ALL ONES
2149                                                        ;POSTAMBLE SYNC GROUP 2
2150  461C  55                          .BYTE   @125        ;CHARACTER 3 = ALL ONES
2151                                                        ;POSTAMBLE SYNC GROUP 2
2152  461D  AA                          .BYTE   @252        ;CHARACTER 4 = ALL ONES
2153                                                        ;POSTAMBLE SYNC GROUP 2
2154                                                        ;CHARACTER 5 = ALL ONES
2155  461E  55                          .BYTE   @125        ;
2156                                                        ;POSTAMBLE SYNC GROUP 3
2157  461F  AA                          .BYTE   @252        ;CHARACTER 1 = ALL ONES
2158                                                        ;POSTAMBLE SYNC GROUP 3
2159  4620  55                          .BYTE   @125        ;CHARACTER 2 - ALL ONES
2160                                                        ;POSTAMBLE SYNC GROUP 3
2161  4621  AA                          .BYTE   @252        ;CHARACTER 3 = ALL ONES
2162                                                        ;POSTAMBLE SYNC GROUP 3
2163  4622  55                          .BYTE   @125        ;CHARACTER 4 = ALL ONES
2164                                                        ;POSTAMBLE SYNC GROUP 3
2165                                                        ;CHARACTER 5 = ALL ONES
2166  4623  AA                          .BYTE   @252        ;
2167                                                        ;POSTAMBLE SYNC GROUP 4
2168  4624  55                          .BYTE   @125        ;CHARACTER 1 = ALL ONES
2169                                                        ;POSTAMBLE SYNC GROUP 4
2170  4625  AA                          .BYTE   @252        ;CHARACTR 2 - ALL ONES
2171                                                        ;POSTAMBLE SYNC GROUP 4
2172  4626  55                          .BYTE   @125        ;CHARACTER 3 = ALL ONES
2173                                                        ;POSTAMBLE SYNC GROUP 4
2174  4627  AA                          .BYTE   @252        ;CHARACTER 4 = ALL ONES
2175                                                        ;POSTAMBLE SYNC GROUP 4
2176                                                        ;CHARACTER 5 = ALL ONES
                                                            ;
```

```
2177  4628  55          .BYTE  @125      ;POSTAMBLE SYNC GROUP 5
2178                                     ;CHARACTER 1 = ALL ONES
2179  4629  AA          .BYTE  @252      ;POSTAMBLE SYNC GROUP 5
2180                                     ;CHARACTER 2 - ALL ONES
2181  462A  55          .BYTE  @125      ;POSTAMBLE SYNC GROUP 5
2182                                     ;CHARACTER 3 = ALL ONES
2183  462B  AA          .BYTE  @252      ;POSTAMBLE SYNC GROUP 5
2184                                     ;CHARACTER 4 = ALL ONES
2185  462C  55          .BYTE  @125      ;POSTAMBLE SYNC GROUP 5
2186                                     ;CHARACTER 5 = ALL ONES
2187                                     ;
2188  462D  AA          .BYTE  @252      ;POSTAMBLE SYNC GROUP 6
2189                                     ;CHARACTER 1 = ALL ONES
2190  462E  55          .BYTE  @125      ;POSTAMBLE SYNC GROUP 6
2191                                     ;CHARACTER 2 - ALL ONES
2192  462F  AA          .BYTE  @252      ;POSTAMBLE SYNC GROUP 6
2193                                     ;CHARACTER 3 = ALL ONES
2194  4630  55          .BYTE  @125      ;POSTAMBLE SYNC GROUP 6
2195                                     ;CHARACTER 4 = ALL ONES
2196  4631  AA          .BYTE  @252      ;POSTAMBLE SYNC GROUP 6
2197                                     ;CHARACTER 5 = ALL ONES
2198                                     ;
2199  4632  55          .BYTE  @125      ;POSTAMBLE SYNC GROUP 7
2200                                     ;CHARACTER 1 = ALL ONES
2201  4633  AA          .BYTE  @252      ;POSTAMBLE SYNC GROUP 7
2202                                     ;CHARACTER 2 - ALL ONES
2203  4634  55          .BYTE  @125      ;POSTAMBLE SYNC GROUP 7
2204                                     ;CHARACTER 3 = ALL ONES
2205  4635  AA          .BYTE  @252      ;POSTAMBLE SYNC GROUP 7
2206                                     ;CHARACTER 4 = ALL ONES
2207  4636  55          .BYTE  @125      ;POSTAMBLE SYNC GROUP 7
2208                                     ;CHARACTER 5 = ALL ONES
2209                                     ;
2210  4637  AA          .BYTE  @252      ;POSTAMBLE SYNC GROUP 8
2211                                     ;CHARACTER 1 = ALL ONES
2212  4638  55          .BYTE  @125      ;POSTAMBLE SYNC GROUP 8
2213                                     ;CHARACTER 2 - ALL ONES
2214  4639  AA          .BYTE  @252      ;POSTAMBLE SYNC GROUP 8
2215                                     ;CHARACTER 3 = ALL ONES
2216  463A  55          .BYTE  @125      ;POSTAMBLE SYNC GROUP 8
2217                                     ;CHARACTER 4 = ALL ONES
2218  463B  AA          .BYTE  @252      ;POSTAMBLE SYNC GROUP 8
2219                                     ;CHARACTER 5 = ALL ONES
2220                                     ;
2221  463C  55          .BYTE  @125      ;POSTAMBLE SYNC GROUP 9
2222                                     ;CHARACTER 1 = ALL ONES
2223  463D  AA          .BYTE  @252      ;POSTAMBLE SYNC GROUP 9
2224                                     ;CHARACTER 2 - ALL ONES
2225  463E  55          .BYTE  @125      ;POSTAMBLE SYNC GROUP 9
2226                                     ;CHARACTER 3 = ALL ONES
2227  463F  AA          .BYTE  @252      ;POSTAMBLE SYNC GROUP 9
2228                                     ;CHARACTER 4 = ALL ONES
2229  4640  55          .BYTE  @125      ;POSTAMBLE SYNC GROUP 9
2230                                     ;CHARACTER 5 = ALL ONES
```

```
2231                                                    ;
2232  4641  AA                  .BYTE   @252            ;POSTAMBLE SYNC GROUP 10
2233                                                    ;CHARACTER 1 = ALL ONES
2234  4642  55                  .BYTE   @125            ;POSTAMBLE SYNC GROUP 10
2235                                                    ;CHARACTER 2 - ALL ONES
2236  4643  AA                  .BYTE   @252            ;POSTAMBLE SYNC GROUP 10
2237                                                    ;CHARACTER 3 = ALL ONES
2238  4644  55                  .BYTE   @125            ;POSTAMBLE SYNC GROUP 10
2239                                                    ;CHARACTER 4 = ALL ONES
2240  4645  AA                  .BYTE   @252            ;POSTAMBLE SYNC GROUP 10
2241                                                    ;CHARACTER 5 = ALL ONES
2242                                                    ;
2243  4646  55                  .BYTE   @125            ;POSTAMBLE SYNC GROUP 11
2244                                                    ;CHARACTER 1 = ALL ONES
2245  4647  AA                  .BYTE   @252            ;POSTAMBLE SYNC GROUP 11
2246                                                    ;CHARACTER 2 - ALL ONES
2247  4648  55                  .BYTE   @125            ;POSTAMBLE SYNC GROUP 11
2248                                                    ;CHARACTER 3 = ALL ONES
2249  4649  AA                  .BYTE   @252            ;POSTAMBLE SYNC GROUP 11
2250                                                    ;CHARACTER 4 = ALL ONES
2251  464A  55                  .BYTE   @125            ;POSTAMBLE SYNC GROUP 11
2252                                                    ;CHARACTER 5 = ALL ONES
2253                                                    ;
2254  464B  AA                  .BYTE   @252            ;POSTAMBLE SYNC GROUP 12
2255                                                    ;CHARACTER 1 = ALL ONES
2256  464C  55                  .BYTE   @125            ;POSTAMBLE SYNC GROUP 12
2257                                                    ;CHARACTER 2 - ALL ONES
2258  464D  AA                  .BYTE   @252            ;POSTAMBLE SYNC GROUP 12
2259                                                    ;CHARACTER 3 = ALL ONES
2260  464E  55                  .BYTE   @125            ;POSTAMBLE SYNC GROUP 12
2261                                                    ;CHARACTER 4 = ALL ONES
2262  464F  AA                  .BYTE   @252            ;POSTAMBLE SYNC GROUP 12
2263                                                    ;CHARACTER 5 = ALL ONES
2264                                                    ;
2265  4650  55                  .BYTE   @125            ;POSTAMBLE SYNC GROUP 13
2266                                                    ;CHARACTER 1 = ALL ONES
2267  4651  AA                  .BYTE   @252            ;POSTAMBLE SYNC GROUP 13
2268                                                    ;CHARACTER 2 - ALL ONES
2269  4652  55                  .BYTE   @125            ;POSTAMBLE SYNC GROUP 13
2270                                                    ;CHARACTER 3 = ALL ONES
2271  4653  AA                  .BYTE   @252            ;POSTAMBLE SYNC GROUP 13
2272                                                    ;CHARACTER 4 = ALL ONES
2273  4654  55                  .BYTE   @125            ;POSTAMBLE SYNC GROUP 13
2274                                                    ;CHARACTER 5 = ALL ONES
2275                                                    ;
2276  4655  AA                  .BYTE   @252            ;POSTAMBLE SYNC GROUP 14
2277                                                    ;CHARACTER 1 = ALL ONES
2278  4656  55                  .BYTE   @125            ;POSTAMBLE SYNC GROUP 14
2279                                                    ;CHARACTER 2 - ALL ONES
2280  4657  AA                  .BYTE   @252            ;POSTAMBLE SYNC GROUP 14
2281                                                    ;CHARACTER 3 = ALL ONES
2282  4658  55                  .BYTE   @125            ;POSTAMBLE SYNC GROUP 14
2283                                                    ;CHARACTER 4 = ALL ONES
2284  4659  AA                  .BYTE   @252            ;POSTAMBLE SYNC GROUP 14
```

```
2285                                                        ;CHARACTER 5 = ALL ONES
2286                                                        ;
2287  465A  55                          .BYTE   @125       ;POSTAMBLE SECOND GROUP
2288                                                        ;CHARACTER 1 = ALL ONES
2289  465B  AA                          .BYTE   @252       ;POSTAMBLE SECOND GROUP
2290                                                        ;CHARACTER 2 = ALL ONES
2291  465C  55                          .BYTE   @125       ;POSTAMBLE SECOND GROUP
2292                                                        ;CHARACTER 3 = ALL ONES
2293  465D  AA                          .BYTE   @252       ;POSTAMBLE SECOND GROUP
2294                                                        ;CHARACTER 4 = ALL ONES
2295  465E  AA                          .BYTE   @252       ;POSTAMBLE SECOND GROUP
2296                                                        ;CHARACTER 5 = ALL ZEROS
2297                                                        ;
2298  465F  55                          .BYTE   @125       ;POSTAMBLE TERM GROUP
2299                                                        ;CHARACTER 1 = ALL ONES
2300  4660  55                          .BYTE   @125       ;POSTAMBLE TERM GROUP
2301                                                        ;CHARACTER 2 = ALL ZEROS
2302  4661  AA                          .BYTE   @252       ;POSTAMBLE TERM GROUP
2303                                                        ;CHARACTER 3 = ALL ONES
2304  4662  AA                          .BYTE   @252       ;POSTAMBLE TERM GROUP
2305                                                        ;CHARACTER 4 = ALL ZEROS
2306  4663  AA                GTBLEN:   .BYTE   @252       ;POSTAMBLE TERM GROUP
2307                                                        ;CHARACTER 5 = 0'S TO RESET TRACKS
```

```
                                .SBTTL  TABLE EXPECTED PARITY

                                ;HERE IS THE TABLE OF EXPECTED PARITY STATES FOR DATA & RESIDUAL GROUPS
                                ;ALL BYTES IN DATA GRP=0 (EVEN ECC) SO 'P' TRACK 4 X 5 IS 0001 = 01111

4664  00                        PXPTBL: .BYTE   0               ;1ST ENTRY IS A DUMMY
4665  00                                .BYTE   0               ;'P' TRANSLATION ACTS SAME AS OTHER
4666  01                                .BYTE   1               ;TRACKS
4667  00                                .BYTE   0
4668  01                                .BYTE   1
4669  00                                .BYTE   0
466A  00                                .BYTE   0
466B  01                                .BYTE   1
466C  00                                .BYTE   0
466D  01                                .BYTE   1
466E  00                                .BYTE   0

                                ;SKIP OVER END MARK GRP

466F  FF                                .BYTE   @377
4670  FF                                .BYTE   @377
4671  FF                                .BYTE   @377
4672  FF                                .BYTE   @377
4673  FF                                .BYTE   @377

                                ;RESIDUAL GRP
                                ;1ST GRP IS 4 PADS SO 4 X 5 IS 0000 = 11001

4674  01                                .BYTE   1
4675  00                                .BYTE   0
4676  01                                .BYTE   1
4677  00                                .BYTE   0
4678  01                                .BYTE   1

                                ;2ND GRP IS 2 PAD BYTES + ACRC OF 323 + ECC OF 225
                                ;'P' 4 X 5 IS 1101 = 01101

4679  01                                .BYTE   1
467A  00                                .BYTE   0
467B  01                                .BYTE   1
467C  01                                .BYTE   1
467D  00                                .BYTE   0

                                ;CRC GRP 1ST HALF IS 4 CRC BYTES OF 136
                                ;'P' 4 X 5 IS 0000 = 11001

467E  01                                .BYTE   1
467F  00                                .BYTE   0
4680  00                                .BYTE   0
4681  00                                .BYTE   0
4682  01                                .BYTE   1

                                ;2ND HALF OF CRC GRP IS 2 CRC OF 136 + RESID BYTE OF 006 + ECC BYTE OF 352
                                ;'P' 4 X 5 IS 0010 = 10010
```

```
2364
2365  4683  00                          .BYTE   0
2366  4684  00                          .BYTE   0
2367  4685  00                          .BYTE   0
2368  4686  01                          .BYTE   1
2369  4687  01                          .BYTE   1
2370
2371                              ;HERE IS THE MARK-2 PARITY TABLE
2372
2373  4688  00                          .BYTE   0               ;MARK 2 IS 11100
2374  4689  01                          .BYTE   1
2375  468A  00                          .BYTE   0
2376  468B  00                          .BYTE   0
2377  468C  00                          .BYTE   0
2378
2379
2380
2381
2382                              .SBTTL  PROGRAM VARIABLES
2383
2384  468D  00                    UNITMP: .BYTE   0               ;UNIT MAP
2385  468E  00                    XDATA:  .BYTE   0               ;COPY OF EXPECTED DATA
2386  468F  00    00              DPCNT:  .WORD   0               ;PARITY TEST COUNTER
2387  4691  00                    EXDLPC: .BYTE   0               ;TRANSITION COUNT AT ERROR
2388  4692  00    00              POINTER: .WORD  0               ;POINTER TO EXPECTED DATA TABLE
2389  4694  00    00              TRANSCNT: .WORD 0               ;TRANSITION COUNT
2390  4696  00                    CCTLWD: .BYTE   0               ;CLOCK CONTROL WORD
2391
2392        0000                          .END
```

```
A      =%0007          ABOCNO  43DE  G       ADATA  = 0094          AMTIEP= 0001
AMTIE7= 0002           ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000         BADST  = 0090          BGNPRE  43E7
BIT0   = 0001          BIT1   = 0002         BIT15  = 8000          BIT2   = 0004
BIT3   = 0008          BIT4   = 0010         BIT5   = 0020          BIT6   = 0040
BIT7   = 0080          BIT8   = 0100         BIT9   = 0200          BRKPBC= 4F0A
BRKRAM= 4F10           BRKSTR= 4E60          BRKXCT= 4F00           BSAVE   4F9C
BYTCNT= 00D4           BYTEH   4F24          BYTEL   4F23           C      =%0001
CASCT   4F21           CASCTL= 00A0          CASSTA= 00A0           CATTH  = 0089
CATTL  = 0088          CBUSST= 00A1          CBYTH  = 008B          CBYTL  = 008A
CCTLWD  4696           CDG1H  = 0087         CDG1L  = 0086          CDG2H  = 0093
CDG2L  = 0092          CDG3H  = 0095         CDG3L  = 0094          CDVTH  = 008D
CDVTL  = 008C          CHPTIE= 0028          CHOTIE= 0020           CH1TIE= 0021
CH2TIE= 0022           CH3TIE= 0023          CH4TIE= 0024           CH5TIE= 0025
CH6TIE= 0026           CH7TIE= 0027          CKLOP  = 2815          CLEAR   456C
CLK     43CB           CLKCTL= 00F0          CLKRS   43BE          CLKSYS  4560
CLK1    43C9           CLOCK   4F26          CLRLP   4570          CMC0H  = 0099
CMC0L  = 0098          CMC1H  = 009B         CMC1L  = 009A          CMC2H  = 009D
CMC2L  = 009C          CMC3H  = 009F         CMC3L  = 009E          CMINH  = 0097
CMINL  = 0096          CNTCTL= 00D7          CRCWRD= 0018           CSAVE   4F9D
CSRLH  = 0091          CSRLL  = 0090         CTCH   = 0085          CTCL   = 0084
CTSTH  = 008F          CTSTL  = 008E         CXCTH  = 0081          CXCTL  = 0080
CXINH  = 0083          CXINL  = 0082         C.     = 0001          C.AVAI= 0080
C.DP   = 0008          C.DSE  = 0010         C.DTU  = 0003          C.DVA  = 0008
C.FAIL= 00FC           C.FMT  = 0070         C.FNCT= 003E           C.GO   = 0001
C.INTC= 00FE           C.MAIN= 0020          C.NSA  = 0080          C.RCT  = 00FC
C.SER  = 0080          C.SHR  = 0040         C.SKPC= 000F           C.TAPE= 0040
C.WCS  = 0002          D      =%0002         DATACT= 00D0           DBUS    4F28
DBUSCT= 00C0           DBUSST= 00C0          DDRA   = 00D8          DDRAIN= 0010
DDRB   = 00D9          DDRBIN= 0002          DDRC   = 00DA          DDRCIN= 0001
DDRCO  = 0088          DDRCTL= 00DB          DIAFLG  4F22          DIAGPG= 4300
DIAGRM= 4F90           DIARD  = 000B         DONE1  = 0045          DONINT= 0010
DPCNT   468F           DPTST   4472          DSAVE   4F9E          DSE    = 0006
DUMMY   4358  G        D.ATH0= 0001          D.ATH1= 0002           D.EOTD= 0010
D.LAGC= 0020           D.NOTW= 0040          D.NTHR= 0004           D.TACH= 0008
D.WR4  = 0080          E      =%0003         ECCBAD= 0042           ECCCOR= 0019
ECCOK  = 0041          ECCSTA= 001A          ECCTST= 000E           EDATA  = 0095
EOTCLR= 0003           ERFLG   4F93          ERLP   = 2809          ERLPA  = 280F
ERLPB  = 2812          ERLPE  = 280C         ERNUM   4F90          ERRCNT= 00D6
ESAVE   4F9F           EXDLPC  4691          EXIT    455D          E.ACRC= 0010
E.AMT  = 0020          E.CDP  = 0080         E.CRC  = 0080          E.PNTR= 0008
E.RPE  = 0040          E.STEC= 0001          E.TTEC= 0002           E.UNC  = 0004
FIFORD= 006A           FORMAT  4F25          FOUND   4361          FWDTST= 0061
GCRID  = 0089          GCRSET= 0002          GDATEN  4614          GOODTM= 0092
GTBLEN  4663           GXPDAT  45EC          GXPTBL  4597          H      =%0004
HLSAVE  4FA0           IE     = 0008         INTSTA= 00E0           ITERA   4F9A
I.PWR  = 0020          I.RMPE= 0040          I5.5   = 0010          I6.5   = 0020
I7.5   = 0040          KCALL  = 005F         KCLR   = 007B          KDEP   = 003F
KENAB  = 0078          KEXAM  = 003E         KEYBRD= 00C8           KEY1   = 0078
KEY10  = 006D          KEY11  = 006E         KEY12  = 006F          KEY13  = 005C
KEY14  = 005D          KEY15  = 005E         KEY16  = 005F          KEY17  = 003C
KEY18  = 003D          KEY19  = 003E         KEY2   = 0079          KEY20  = 003F
KEY3   = 007A          KEY4   = 007B         KEY5   = 0074          KEY6   = 0075
KEY7   = 0076          KEY8   = 0077         KEY9   = 006C          KINTA  = 006F
```

```
KLDAD = 003D        KN0   = 003C        KN1   = 005C        KN2   = 005D
KN3   = 005E        KN4   = 006C        KN5   = 006D        KN6   = 006E
KN7   = 0074        KN8   = 0075        KN9   = 0076        KU2   = 0079
KU3   = 007A        KU8   = 0077        L     =%0005        LBLANK= 000F
LCE   = 000B        LCH   = 000C        LCL   = 000D        LCP   = 000E
LC0   = 0000        LC1   = 0001        LC2   = 0002        LC3   = 0003
LC4   = 0004        LC5   = 0005        LC6   = 0006        LC7   = 0007
LC8   = 0008        LC9   = 0009        LDLEDA= 00CA        LDLEDB= 00CB
LDLEDC= 00CC        LDLEDD= 00CD        LDLEDE= 00CE        LDLEDF= 00CF
LKDIAG= 2800        LKKBD = 004C        LKKEY = 0049        LKLWMG= 0058
LKLWMP= 0055        LKLWPG= 0052        LKLWPP= 004F        LKMOD7= 0046
LKOPR = 0046        LPFLG   4F94        LPNUM   4F92        M     =%0006
MBSEL = 00E0        MB.A  = 0008        MB.B  = 0004        MEMTOP= 4FFF
MINUS = 000A        MM    = 8000        MSE   = 0008        MSGN  = 0008
MTACLR= 0000        MT.ARA= 0020        MT.CPE= 0080        MT.DSE= 0001
MT.FWD= 0040        MT.INH= 0008        MT.LWR= 0004        MT.MOT= 0002
MT.NWT= 0080        MT.PEC= 0040        MT.PSB= 0004        MT.PS0= 0001
MT.PS1= 0002        MT.REV= 0020        MT.WRT= 0010        MT.Z  = 0008
M.ATA = 0080        M.CAPE= 0020        M.CONT= 0080        M.DEM = 0020
M.EBL = 0010        M.EXC = 0008        M.FAIL= 0008        M.ILR = 0010
M.INIT= 0010        M.OCC = 0020        M.ONLI= 0001        M.PE  = 0040
M.PORT= 0080        M.RDEN= 0002        M.RDPE= 0008        M.RUN = 0004
M.SCLK= 0001        M.TRA = 0040        M.UNIT= 0007        M.WCLK= 0040
M.WCLN= 0080        M.WREN= 0080        M5.5  = 0001        M6.5  = 0002
M7.5  = 0004        NOTCAP= 0088        OKAY  = 00FF        OPRRAM= 4300
OPSTRT= 0058        OPVER = 0040        PADCNT= 00D5        PADCRC= 0080
PDIAG = 0048        PEID  = 008A        PENAB = 004C        PESET = 0001
PL    = 00B1        POINTE  4692        PPOST   4485        PPREAM  4485
PRDD  = 004C        PRECN1  4422  G     PRECN3  44E4  G     PRENF = 009C
PS    = 00B2        PSTAT = 0048        PSW   =%0009        PTSTOK  44A7  G
PTY     448A        PTYC    44C8        PTY1    4495        PTY2    44A2
PXPTBL  4664        P.AMTP= 0001        P.BLTC= 0040        P.CMDP= 0020
P.INIE= 0080        P.LCS = 0040        P.LWR = 0020        P.RDP = 0002
P.RPEN= 0004        P.RPST= 0002        P.RP0E= 0020        P.RP1E= 0010
P.RP2E= 0020        P.RP3E= 0010        P.SING= 0080        P.STAT= 0002
P.STPE= 0080        P.TACH= 0008        P.TUPR= 0010        P.WCSP= 0004
P.WDS = 0040        P.WFLP= 0001        P.WPEN= 0010        P.WP0E= 0008
P.WP1E= 0004        P.WP2E= 0008        P.WP3E= 0004        P.5VOK= 0002
QUE   = 281B        QUEM  = 281E        RAMT  = 0010        RARA  = 0006
RARAI = 0004        RCHBD0= 0048        RCHBD1= 0047        RCHOK = 0046
RCHTST= 000C        RCLRT = 000D        RCMD  = 000B        RCMLP = 0003
RCONT = 0080        RDATA = 0017        RDCLK = 0010        RDON  = 0011
READG = 0007        RECCN1  451A  G     RECCN2  4543  G     RECCN3  4535  G
RECEND  44FE        REND  = 0014        REGST = 2806        RESCHR= 00D1
REVTST= 0064        REWIND= 0004        RFIFOL= 0008        RGCLK = 0002
RGCRI = 0003        RIBG  = 0001        RILL  = 0012        RINST = 000C
RMCTST= 0008        RMK2  = 0013        RNOP  = 0000        RPATH = 0001
RPBAD = 0044        RPCHI = 0001        RPCLK = 0003        RPCTL = 0009
RPEI  = 0002        RPFAIL= 0000        RPF1  = 009D        RPF2  = 009E
RPOK  = 0043        RPOSTN= 0016        RPSTA = 0015        RRCMT = 000A
RSTAT = 0002        RTIEB = 000A        RTIER = 0030        RTM   = 0005
RUNKI = 0009        RUPTST= 005E        RWDUNL= 0005        R.AMT = 0001
R.BOP = 0008        R.DATA= 0040        R.DON = 0002        R.DRDY= 0010
R.END = 0010        R.ILL = 0004        R.JVOK= 0004        R.MK2 = 0008
```

```
R.PLOD= 0008    R.PLOO= 0010    R.PLO1= 0020    R.POST= 0020
R.STNM= 0002    R.STOP= 0004    R.STPC= 0001    R.TBJN= 0080
R.TSTD= 0040    R.VOK = 0080    R00H  = 0081    R00L  = 0080
R01H  = 0083    R01L  = 0082    R02H  = 0085    R02L  = 0084
R03   = 0087    R03L  = 0086    R04H  = 0089    R04L  = 0088
R05H  = 008B    R05L  = 008A    R06H  = 008D    R06L  = 008C
R07H  = 008F    R07L  = 008E    R10H  = 0091    R10L  = 0090
R11H  = 0093    R11L  = 0092    R12H  = 0095    R12L  = 0094
R13H  = 0097    R13L  = 0096    R14H  = 0099    R14L  = 0098
R15H  = 009B    R15L  = 009A    R16H  = 009D    R16L  = 009C
R17H  = 009F    R17L  = 009E    R7.5  = 0010    SELCLR= 0000
SEQERR  453E    SETATA= 00A1    SID   = 0080    SOD   = 0080
SOE   = 0040    SP    =%0008    SSCLK = 0040    SSTEP = 0005
STACK = 4FFF    STATRM= 4F20    STPCT   4F20    STRSP = 5000
TADR00= 0080    TADR01= 0081    TADR02= 0082    TADR03= 0083
TADR04= 0084    TADR05= 0085    TADR06= 0086    TADR07= 0087
TADR10= 0088    TADR11= 0089    TADR12= 008A    TADR13= 008B
TAMT  = 0044    TASEL = 0080    TCMD  = 0040    TC.FWD= 0040
TC.INH= 0008    TC.LWR= 0004    TC.REV= 0020    TC.WRT= 0010
TEMP    4F99    TEST1   4300    TMF   = 0099    TMRDY = 0040
TRANSC  4694    TRKENA= 00D2    TSET  = 2803    TSTEND= 2818
TSTS  = 0040    TUSEL0= 00D1    TUSEL1= 00D2    TU78  = 0010
T.ATH0= 0001    T.ATH1= 0002    T.BOT = 0004    T.EOT = 0002
T.FPT = 0001    T.NTHR= 0004    T.ONL = 0020    T.PES = 0008
T.PSBJ= 0020    T.PS0J= 0008    T.PS1J= 0010    T.RDY = 0080
T.RDY0= 0040    T.RWD = 0010    T.SCLK= 0002    T1LOOP  4305
T1LPA   4330    UIBG  = 00A1    UNITMP  468D    VALFC   4F98
VALTB   4F95    VELTST= 005B    WDR.P = 0010    WMCCTL= 00D3
WMCERR= 00DA    WMCSTA= 00D0    WRTCLK= 0000    WRTDAT= 00D3
W.ACRC= 0004    W.CRC = 0008    W.DIAG= 0002    W.DONN= 0040
W.ECC = 0010    W.ENAB= 0080    W.ERR = 0020    W.FMT = 0070
W.GCR = 0010    W.LEFT= 0004    W.ONES= 0020    W.RESI= 0002
W.REV = 0004    W.ROME= 0010    W.RST = 0001    W.SKIP= 000F
W.WRIT= 0008    W.XFER= 0020    X     =%000A    XDATA   468E
XDLOOP  442B    XDLPC   4446    XDLPC1  4410    XDLPC2  441D
XDLP1   4438    XMEND   4549    XPLOOP  44B0    XPLP1   44BD
X.DONN= 0080    X.ENAB= 0040    X.PEPE= 0002    X.ROME= 0001
X.WCLK= 0001    Y     =%000B    .     = 4697


 ERRORS DETECTED:  0

 *XMC3.A78/PTP,XMC3=NLIST,PARAM,MACRO,LIST,XMC3
 RUN-TIME: 4 6 0 SECONDS
 CORE USED:  10K
```

3 "MEMORY MAP" EXTENDED ADDRESS BIT
19 READ MICRO CONTROLLER CONTROL & STATUS WORDS
139 TRANSPORT/PORT COMMAND + STATUS WORDS
380 COMMON ADDRESS SPACE REGISTERS
444 MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485 MASSBUS CONTROL WORD DEFINITIONS
557 WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709 KEYBOARD/DISPLAY CONTROL + STATUS WORD
797 MISCELLANEOUS VARIABLE DEFINITIONS
801 RAM ADDRESS DEFINITIONS
819 LINKAGE TABLE ADDRESSES
842 BIT DEFINITIONS
856 RIM AND SIM INSTRUCTION BIT DEFINITIONS
877 INTERRUPTABLE INSTRUCTION MACROS
924 DIAGNOSTIC VARIABLES
945 COMMON VARIABLE STORAGE
959 MACRO CALL DEFINITIONS
1047 TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101 8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1333 TEST 1 - XMC - GCR - PREAMBLE/POSTAMBLE TEST NO DATA GROUP
1808 SUBROUTINE CLEAR ALL TU PORTS
2311 TABLE EXPECTED PARITY
2363 PROGRAM VARIABLES

```
                ; ************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;               "BYTE/SCLK COUNT NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               MM = DATA FORMAT FROM CAS REGISTER 2
                ;               NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;               BYTE-SCLK COUNT = LLL
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               LLL = AS ABOVE
                ;               MM  = AS ABOVE
                ;               NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;               TRANSITION COUNT = LLL
                ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;               EXPECTED 18 BITS =      E EEEEEE
                ;               ACTUAL 18 BITS =        A AAAAAA
                ;
                ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;               OF CAS REG 15 LOW BYTE.
                ;
                ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;               17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;               AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;               "SUBGROUP NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       10 -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ; ************************************************************************
```

```
1016                            ; ************************************************************************
1017                            ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                            ;TO CAUSE SOME HOST CPU ACTION.
1019                            ;
1020                            ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                            ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                            ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                            ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                            ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                            ;               - HOST RESPONSE CODE 31 OR 33
1026                            ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                            ;               - HOST RESPONSE CODE 31 OR 33
1028                            ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                            ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                            ;                       BIT0 = 1 TEST PORT 0
1031                            ;                       BIT1 = 1 TEST PORT 1
1032                            ;                       BIT2 = 1 TEST PORT 2
1033                            ;                       BIT3 = 1 TEST PORT 3
1034                            ; ************************************************************************
1035                            ; ************************************************************************
1036                            ;DATA PATTERN CODES
1037                            ;
1038                            ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                            ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                            ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                            ;               18 BITS OF ALL 1'S
1042                            ;               18 BITS OF ALL 0'S
1043                            ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                            ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                            ; ************************************************************************
1329    4300                            .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  XMC4 - TRANSLATOR MICRO CONTROLLER PART #4
                                .SBTTL  TEST 1 - XMC - GCR - PREAMBLE/POSTAMBLE TEST NO DATA GROUP
                                ;ID     XMC4-TRANSLATOR MICRO CONTROLLER PART #4
4300                            ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR - PREAMBLE/POSTAMBLE TEST - NO DATA GROUP
4300                            SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;THIS TEST COMMANDS THE WRITE PATH TO WRITE A GCR RECORD OF ONE BYTE
                                ;AND THEN SINGLE STEPS THE WRITE PATH (WMC, XMC) WHILE MONITORING THE
                                ;XMC "XL WCLK" BIT.  THE OUTPUT OF THE TPANSLATOR SHOULD INDICATE A PRE-
                                ;AMBLE, END MARK, RESIDUAL GROUP, CRC GROUP, AND POSTAMBLE.
                                ;THIS OUTPUT IS CHECKED FOR SEQUENCE, CONTENT, DATA STATES AND FLUX REVERSAL RATE.
4300                            SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      INITIALIZE THE BYTE COUNTER, PAD COUNTER, AND ERROR COUNTER
                                ;*      CLEAR THE ERROR COUNTER
                                ;*      REQUEST A TU PORT FROM THE HOST
                                ;*      CALL SUBROUTINE CLEAR
                                ;*      IF NO UNITS SELECTED
                                ;*      :       THEN-ERROR-ABORT TEST
                                ;*      :       ELSE-CONTINUE
                                ;*      ENDIF
                                ;*      SET LOOP COMMAND/STATUS ON SELECTED PORT
                                ;*      CLEAR THE COMMAND BYIE FOR SELECTED PORT
                                ;*      SET WRITE PATH ENABLE
                                ;*      INITIALIZE THE EXPECTED RESULTS TABLE POINTER
                                ;*      SET WMC/XMC RESTART
                                ;*      SELECT THE SINGLE STEP CLOCK
                                ;*      CLEAR WMC/XMC RESTART
                                ;*      LOAD THE BYTE COUNT WITH 1 (BITS 7:0)
                                ;*      CLEAR THE BYTE COUNT (BITS 15:8)
                                ;*      SET DDR CONTROL TO "IN"
                                ;*      SELECT THE IMAGE MODE OF DATA FORMATTING
                                ;*      LOAD THE RESIDUAL CHARACTER WITH 40(8)
                                ;*      LOAD THE PAD COUNT WITH 6 (BITS 7:0)
                                ;*      CLEAR THE PAD COUNT (BITS 15:8)
                                ;*      ENABLE ALL TRACKS LOAD TRACK ENABLE WITH ONES
                                ;*      SET THE CLOCK COUNT TO 180(10)
                                ;*      BGNDO
                                ;*      :       CALL SUBROUTINE CLKSYS
                                ;*      :       DECREMENT THE CLOCK COUNT
                                ;*      :       DO UNTIL CLOCK COUNT=0
                                ;*      ENDDO
                                ;*      SET WRITE MICRO TO WRITE MICRO ENABLE, TRANSLATOR ENABLE, GCR, AND WRITE
                                ;*      SET THE CLOCK COUNT TO 256(10)
                                ;*      BGNDO
```

```
1377                    ;*      :   CALL SUBROUTINE CLKSYS
1378                    ;*      :   DECREMENT THE CLOCK COUNT
1379                    ;*      :   DO UNTIL CLOCK COUNT=0 OR XL WRITE CLOCK=1
1380                    ;*      ENDDO
1381                    ;*      IF CLOCK COUNT=0
1382                    ;*      :   THEN-FATAL ERROR-ABORT TEST
1383                    ;*      :   ELSE-CONTINUE
1384                    ;*      ENDIF
1385                    ;*      BGNDO
1386                    ;*      :   CALL SUBROUTINE CLKSYS
1387                    ;*      :   CALL SUBROUTINE CLKSYS
1388                    ;*      :   INPUT THE TRANSLATOR OUTPUT DATA
1389                    ;*      :   IF THE ACTUAL TRANSLATOR OUTPUT=EXPECTED TRANSLATOR OUTPUT
1390                    ;*      :   :   THEN-CONTINUE
1391                    ;*      :   :   ELSE-ERROR
1392                    ;*      :   ENDIF
1393                    ;*      :   INPUT THE TRANSLATOR OUTPUT DATA PARITY
1394                    ;*      :   IF DATA OUTPUT DURING DATA GROUP
1395                    ;*      :   :   THEN-EXPECTED PARITY=ODD PARITY FOR OUTPUT DATA
1396                    ;*      :   :   ELSE-EXPECTED PARITY=1
1397                    ;*      :   ENDIF
1398                    ;*      :   IF INPUT PARITY=EXPECTED PARITY
1399                    ;*      :   :   THEN-CONTINUE
1400                    ;*      :   :   ELSE-ERROR
1401                    ;*      :   ENDIF
1402                    ;*      :   SET THE LOOP COUNT TO 14(8)
1403                    ;*      :   BGNDO
1404                    ;*      :   :   CALL SUBROUTINE CLKSYS
1405                    ;*      :   :   DECREMENT THE LOOP COUNT
1406                    ;*      :   :   DO UNTIL THE LOOP COUNT = 0
1407                    ;*      :   ENDDO
1408                    ;*      :   IF XL WRITE CLOCK = 1
1409                    ;*      :   :   THEN - CONTINUE
1410                    ;*      :   :   ELSE - ERROR - IMPROPER TRANSITION RATE FOR GCR
1411                    ;*      :   ENDIF
1412                    ;*      :   INCREMENT THE EXPECTED RESULTS TABLE POINTER
1413                    ;*      :   DO UNTIL THE EXPECTED RESULTS POINTER=END OF TABLE
1414                    ;*      ENDDO
1415                    ;*      SET THE CLOCK COUNT TO 256(10)
1416                    ;*      BGNDO
1417                    ;*      :   CALL SUBROUTINE CLKSYS
1418                    ;*      :   DECREMENT THE CLOCK COUNT
1419                    ;*      :   DO UNTIL CLOCK COUNT=0 OR TRANSLATOR OR WRITE MICRO DONE=0
1420                    ;*      ENDDO
1421                    ;*      IF CLOCK COUNT=0
1422                    ;*      :   THEN-ERROR-TIMEOUT WAITING FOR WRITE MICRO DONE
1423                    ;*      :   ELSE-CONTINUE
1424                    ;*      ENDIF
1425                    ;*      IF TRANSLATOR DONE
1426                    ;*      :   THEN-ERROR-TRANSLATOR DONE BEFORE WRITE MICRO
1427                    ;*      :   ELSE-SET THE CLOCK COUNT TO 256(10)
1428                    ;*      :   BGNDO
1429                    ;*      :   :   CALL SUBROUTINE CLKSYS
1430                    ;*      :   :   DECREMENT THE CLOCK COUNT
```

```
                                ;*    :    :    DO UNTIL TRANSLATOR DONE OR CLOCK COUNT=0
                                ;*    :    ENDDO
                                ;*    :    IF CLOCK COUNT=0
                                ;*    :    :    THEN-ERROR-TIMEOUT WAITING FOR TRANSLATOR DONE
                                ;*    :    :    ELSE-CONTINUE
                                ;*    :    ENDIF
                                ;*    ENDIF
                                ;*ENDTST
 1439  4300                     SE
  (1)                           ; ************************************************************************
  (1)                           ;*ERRORS
  (1)                           ;*------
 1440                           ;*XMC4 MICRO TEST 01
 1441                           ;*XMC4 MICRO ERROR 01
 1442                           ;*XMC4-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
 1443                           ;*M8958, M8959, M8955'S
 1444                           ;*OPERATOR ERROR NO TM78 UNITS SELECTED
 1445                           ;*FATAL ERROR - TEST ABORTED
 1446                           ;*
 1447                           ;*XMC4 MICRO TEST 01
 1448                           ;*XMC4 MICRO ERROR 02
 1449                           ;*XMC4-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
 1450                           ;*M8958, M8959, M8955'S
 1451                           ;*WRITE PATH TIMEOUT WHILE WAITING FOR "XL CLK" TO SET
 1452                           ;*FATAL ERROR - TEST ABORTED
 1453                           ;*
 1454                           ;*XMC4 MICRO TEST 01
 1455                           ;*XMC4 MICRO ERROR 03
 1456                           ;*XMC4-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
 1457                           ;*M8958, M8959, M8955'S
 1458                           ;*DATA STATE RECEIVED FROM THE XMC INCORRECT
 1459                           ;*ACTUAL = NNNN
 1460                           ;*EXPECTED = NNNN
 1461                           ;*TRANSITION COUNT = LLL
 1462                           ;*
 1463                           ;*XMC4 MICRO TEST 01
 1464                           ;*XMC4 MICRO ERROR 04
 1465                           ;*XMC4-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
 1466                           ;*M8958, M8959, M8955'S
 1467                           ;*PARITY STATE RECEIVED FROM THE XMC INCORRECT
 1468                           ;*ACTUAL = NNNN
 1469                           ;*EXPECTED = NNNN
 1470                           ;*TRANSITION COUNT = LLL
 1471                           ;*
 1472                           ;*XMC4 MICRO TEST 01
 1473                           ;*XMC4 MICRO ERROR 05
 1474                           ;*XMC4-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
 1475                           ;*M8958, M8959, M8955'S
 1476                           ;*TRANSLATOR "XL WCLK" NOT SET AFTER EIGHT CLOCK CYCLES
 1477                           ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
 1478                           ;*TRANSITION COUNT = LLL
 1479                           ;*
 1480                           ;*XMC4 MICRO TEST 01
 1481                           ;*XMC4 MICRO ERROR 06
```

```
1482                                    ;*XMC4-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
1483                                    ;*M8958, M8959, M8955'S
1484                                    ;*''WMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
1485                                    ;*
1486                                    ;*XMC4 MICRO TEST 01
1487                                    ;*XMC4 MICRO ERROR 07
1488                                    ;*XMC4-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
1489                                    ;*M8958, M8959, M8955'S
1490                                    ;*''XMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
1491                                    ;*
1492                                    ;*XMC4 MICRO TEST 01
1493                                    ;*XMC4 MICRO ERROR 10
1494                                    ;*XMC4-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECORD
1495                                    ;*M8958, M8959, M8955'S
1496                                    ;*''XMC NOT DONE'' RESET BEFORE ''WMC NOT DONE'' RESET
1497  4300                              S
 (1)                                    ; ************************************************************************************
1498
1499  4300                              TEST1:  TESTX   @1
 (1)  4300  3E  01          7.0                          MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28     18.0                          CALL    TSET            ;SETUP THE TEST
1500                                    ;%XMC4-GCR-PREAMBLE/POSTAMBLE TEST-NO DATA GROUP-1 BYTE RECURD
1501                                    ;&M8958, M8959, M8955'S
1502
1503  4305  AF                4.0             XRA     A               ;CLEAR ALL TRACK ENABLE LINES
1504  4306  D3  D2            10.0             OUT     TRKENA
1505  4308  D3  D2            10.0             OUT     TRKENA
1506  430A  AF                 4.0             XRA     A               ;CLEAR UNIT MAP STORAGE
1507  430B  32  7E  46         13.0             STA     UNITMP          ;1ST TIME HERE
1508  430E  32  82  46         13.0             STA     EXDLPC          ;CLEAR THE ERROR COUNTER
1509
1510  4311  3E  30              7.0  T1LOOP:    MVI     A,@60           ;INIT THE BYTE COUNTER
1511  4313  D3  D7             10.0             OUT     CNTCTL          ;TO NORMAL MODE
1512  4315  3E  70              7.0             MVI     A,@160          ;INIT THE PAD COUNTER
1513  4317  D3  D7             10.0             OUT     CNTCTL          ;TO NORMAL MODE
1514  4319  3E  B0              7.0             MVI     A,@260          ;INIT THE ERROR COUNTER
1515  431B  D3  D7             10.0             OUT     CNTCTL          ;TO NORMAL MODE
1516  431D                                    CLRECT
 (1)  431D  AF                 4.0                             XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  431E  D3  D6            10.0                             OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  4320  D3  D6            10.0                             OUT     ERRCNT          ;CLEAR BITS 15-8
1517
1518                                    ;REQUEST THE HOST FOR THE PORTS TO BE USED IN THIS TEST...VALUE WILL
1519                                    ;BE IN CAS REG 12 LOW BYTE ON RETURN FROM HOST REQUEST.
1520
1521  4322  3A  7E  46         13.0             LDA     UNITMP          ;GET THE UNIT MAP STORAGE
1522  4325  A7                 4.0             ANA     A
1523  4326  C2  3F  43         10.0             JNZ     T1LPA           ;JUMP OVER IF ALREADY HAVE IT
1524  4329                                    REQ     @7,0,0,0
 (1)  4329  CD  06  28        18.0                             CALL    REQST
 (1)  432C  00                                                .BYTE   0               ;DATA PATTERN NUMBER
 (1)  432D  00  00                                            .WORD   0               ;SYSTEM ''0'' COUNT
 (1)  432F  00  00                                            .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  4331  00                                                .BYTE                   ;DATA COMPARE FLAG IF  1
```

```
 (1)  4332  07                                        .BYTE   @7              ;REQUEST CODE
1525  4333                                  RIN     R12L
 (1)  4333  DB  94          10.0                     IN      R12L            ;READ R12L INTO AC
 (1)  4335  7F               4.0                     MOV     A,A             ;RETRY LINK
1526  4336  32  7E  46       13.0          STA       UNITMP          ;SAVE THE UNIT MAP INFO FROM HOST
1527  4339  AF               4.0          XRA       A
1528  433A  CD  76  45       18.0          CALL      CLEAR
1529  433D  06  00            7.0          MVI       B,0
1530  433F  3A  7E  46       13.0  T1LPA:  LDA       UNITMP          ;GET SELECTED UNIT MAP
1531  4342  E6  01            7.0          ANI       $01             ;UNIT 0 PRESENT?
1532  4344  C2  70  43       10.0          JNZ       FOUND           ;YES-USE IT
1533  4347  04               4.0          INR       B
1534  4348  3A  7E  46       13.0          LDA       UNITMP          ;
1535  434B  E6  02            7.0          ANI       $02             ;UNIT 1 PRESENT?
1536  434D  C2  70  43       10.0          JNZ       FOUND           ;YES-USE IT
1537  4350  04               4.0          INR       B
1538  4351  3A  7E  46       13.0          LDA       UNITMP          ;
1539  4354  E6  04            7.0          ANI       $04             ;UNIT 2 PRESENT?
1540  4356  C2  70  43       10.0          JNZ       FOUND           ;YES-USE IT
1541  4359  04               4.0          INR       B
1542  435A  3A  7E  46       13.0          LDA       UNITMP
1543  435D  E6  08            7.0          ANI       $08             ;UNIT 3 PRESENT
1544  435F  C2  70  43       10.0          JNZ       FOUND           ;YES-USE IT
1545                                                                  ;ELSE - OPERATOR ERROR
1546  4362                                  ERR     EXIT,DUMMY
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4362  CD  09  28       18.0                     CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4365  01                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4366  00                                        .BYTE
 (1)  4367  CD  15  28       18.0          DUMMY:: CALL      CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  436A  DA  67  45       10.0                  JC        EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1547                                    ;>OPERATOR ERROR NO TM78 UNITS SELECTED
1548                                    ;>FATAL ERROR - TEST ABORTED
1549  436D  C3  67  45       10.0          JMP       EXIT
1550
1551                                    ;SET UP FOR THE DATA TRANSFER
1552  4370  DB  E0            10.0  FOUND:  IN        INTSTA
1553  4372  E6  80             7.0          ANI       BIT7
1554  4374  B0               4.0          ORA       B
1555  4375  D3  E0            10.0          OUT       MBSEL
1556  4377  3E  40             7.0          MVI       A,P.LCS         ;KILL THE MTA BOARD
1557  4379  D3  48            10.0          OUT       PDIAG
1558  437B  AF               4.0          XRA       A
1559  437C  D3  40            10.0          OUT       TCMD
1560
1561  437E  3E  10             7.0          MVI       A,P.WPEN
1562  4380  D3  4C            10.0          OUT       PENAB           ;ENABLE THE WRITE PATH
1563  4382  21  A1  45        10.0          LXI       H,GXPTBL        ;LOAD ADDRESS OF EXPECTED
1564  4385  22  83  46        16.0          SHLD      POINTER         ;RESULTS TABLE INTO POINTER
1565  4388  3E  01             7.0          MVI       A,W.RST         ;LOAD THE WMC/XMC RESTART CODE
1566  438A  D3  D3            10.0          OUT       WMCCTL          ;ISSUE THE COMMAND
1567  438C  3E  05             7.0          MVI       A,SSTFP         ;SELECT SINGLE STEP WRITE CLOCK
1568  438E  D3  F0            10.0          OUT       CLKCTL          ;INITIALIZE THE CLOCK
```

```
1569  4390  32  87  46    13.0          STA     CCTLWD          ;SAVE SOFTWARE IMAGE
1570  4393  AF              4.0          XRA     A               ;CLEAR THE ACC.
1571  4394  D3  D3          10.0          OUT     WMCCTL          ;CLEAR THE WMC/XMC RESTART BIT
1572  4396  3E  01          7.0          MVI     A,1             ;GET THE BYTE COUNT - NO DATA GROUP
1573  4398  D3  D4          10.0          OUT     BYTCNT          ;LOAD BITS 7-0
1574  439A  AF              4.0          XRA     A
1575  439B  D3  D4          10.0          OUT     BYTCNT          ;CLEAR THE BYTE COUNTER BITS 15-8
1576  439D  32  85  46      13.0          STA     TRANSCNT        ;INITIALIZE THE TRANSITION COUNT
1577  43A0  32  86  46      13.0          STA     TRANSCNT+1
1578  43A3  32  80  46      13.0          STA     DPCNT           ;CLEAR THE PARITY TEST COUNTER
1579  43A6  3E  88          7.0          MVI     A,@210
1580  43A8  D3  DB          10.0          OUT     DDRCTL          ;SET THE DDR CONTROL TO 'IN'
1581  43AA  3E  50          7.0          MVI     A,$50           ;LOAD THE IMAGE MODE COMMAND
1582  43AC  D3  D0          10.0          OUT     DATACTL         ;SET THE DATA CONTROL MODE
1583  43AE  3E  20          7.0          MVI     A,@40           ;LOAD THE RESIDUAL CHARACTER VALUE
1584                                                            ;1 DATA BYTE IN RESID GROUP
1585  43B0  D3  D1          10.0          OUT     RESCHR          ;SET THE RESIDUAL CHARACTER
1586  43B2  3E  05          7.0          MVI     A,5             ;LOAD PAD COUNT VALUE
1587  43B4  3C              4.0          INR     A               ;CORRECT FOR THE INTEL 8253 CHIP
1588  43B5  D3  D5          10.0          OUT     PADCNT          ;WRITE TO WMC BITS 7-0
1589  43B7  AF              4.0          XRA     A               ;LOAD HI VALUE
1590  43B8  D3  D5          10.0          OUT     PADCNT          ;WRITE TO WMC
1591  43BA  3E  FF          7.0          MVI     A,$FF           ;LOAD ACC WITH ALL ONES
1592  43BC  D3  D2          10.0          OUT     TRKENA          ;ENABLE PARITY TRACK
1593  43BE  D3  D2          10.0          OUT     TRKENA          ;ENABLE DATA TRACKS
1594  43C0  06  B4          7.0          MVI     B,180           ;LOAD THE CLOCK COUNT
1595
1596  43C2  CD  6A  45      18.0  CLKRS:  CALL    CLKSYS          ;CLOCK THE THING
1597  43C5  05              4.0          DCR     B               ;
1598  43C6  C2  C2  43      10.0          JNZ     CLKRS           ;CONTINUE WITH RESTART CLOCKING
1599
1600  43C9  3E  D8          7.0          MVI     A,W.ENAB!X.ENAB!W.GCR!W.WRITE
1601  43CB  D3  D3          10.0          OUT     WMCCTL          ;START THE OPERATION
1602  43CD  06  FF          7.0  CLK1:   MVI     B,$FF           ;SET TIMEOUT COUNT TO 256
1603                                                            ;
1604  43CF  CD  6A  45      18.0  CLK:    CALL    CLKSYS          ;SINGLE STEP THE WRITE PATH
1605                                                            ;
1606  43D2  DB  D0          10.0          IN      WMCSTA          ;GET THE WMC STATUS
1607  43D4  E6  01          7.0          ANI     X.WCLK          ;X.WCLK SET?
1608  43D6  C2  EB  43      10.0          JNZ     BGNPRE          ;YES-BEGINNING OF PREAMBLE
1609  43D9  05              4.0          DCR     B               ;NO-DECREMENT CLOCK
1610  43DA  C2  CF  43      10.0          JNZ     CLK             ;CONTINUE UNTIL ZERO
1611  43DD                                ERR     TSTEND,ABOCNO
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  43DD  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43E0  02                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43E1  00                                     .BYTE
 (1)  43E2  CD  15  28      18.0          ABOCNO::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43E5  DA  18  28      10.0                  JC      TSTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
1612                                ;>WRITE PATH TIMEOUT WHILE WAITING FOR "XL CLK" TO SET
1613                                ;>FATAL ERROR - TEST ABORTED
1614
1615  43E8  C3  18  28      10.0          JMP     TSTEND
```

```
1616
1617  43EB  CD 6A 45   18.0  BGNPRE: CALL   CLKSYS          ;CLOCK THE SYSTEM
1618  43EE  CD 6A 45   18.0          CALL   CLKSYS          ;CLOCK AGAIN
1619  43F1  DB 40      10.0          IN     TSTS            ;GET THE DATA READ
1620  43F3  47           4.0          MOV    B,A             ;SAVE IT IN B=ADATA
1621  43F4                            ROUT   ADATA           ;STORE ACTUAL DATA IN CAS
 (1)  43F4  D3 94       10.0                 OUT    ADATA           ;WRITE AC INTO ADATA
 (1)  43F6  7F           4.0                 MOV    A,A             ;RETRY LINK
1622                                                       ;
1623  43F7  2A 85 46   16.0          LHLD   TRANSCNT        ;GET THE TRANSITION COUNT
1624  43FA  23           6.0          INX    H               ;UPDATE
1625  43FB  22 85 46   16.0          SHLD   TRANSCNT        ;RESTORE THE TRANSITION COUNT
1626  43FE  7D           4.0          MOV    A,L             ;GET TRANSCOUNT LOW
1627  43FF                            ROUT   R05L            ;WRITE TO CAS
 (1)  43FF  D3 8A       10.0                 OUT    R05L            ;WRITE AC INTO R05L
 (1)  4401  7F           4.0                 MOV    A,A             ;RETRY LINK
1628  4402  7C           4.0          MOV    A,H             ;GET TRANSCOUNT HIGH
1629  4403                            ROUT   R05H            ;WRITE TO CAS
 (1)  4403  D3 8B       10.0                 OUT    R05H            ;WRITE AC INTO R05H
 (1)  4405  7F           4.0                 MOV    A,A             ;RETRY LINK
1630  4406  2A 83 46   16.0          LHLD   POINTER         ;GET POINTER TO EXPECTED DATA
1631  4409  7E           7.0          MOV    A,M             ;GET EXPECTED DATA BYTE
1632  440A  32 7F 46   13.0          STA    XDATA           ;SAVE FOR LATER
1633  440D  B8           4.0          CMP    B               ;COMPARE WITH ACTUAL
1634  440E  CA 26 44   10.0          JZ     PRECN1          ;CONTINUE IF EQUAL
1635                                                       ;ELSE-ERROR
1636  4411                            ROUT   EDATA           ;STORE EXPECTED DATA IN CAS
 (1)  4411  D3 95       10.0                 OUT    EDATA           ;WRITE AC INTO EDATA
 (1)  4413  7F           4.0                 MOV    A,A             ;RETRY LINK
1637  4414  3A 82 46   13.0  XDLPC1: LDA    EXDLPC          ;GET THE CURRENT COUNT AT ERROR
1638  4417  A7           4.0          ANA    A
1639  4418  C2 21 44   10.0          JNZ    XDLPC2          ;STAY AT THIS COUNT IF NOT ZERO
1640  441B  3A 85 46   13.0          LDA    TRANSCNT        ;GET THE COUNT AT THE ERROR
1641  441E  32 82 46   13.0          STA    EXDLPC          ;SAVE IT
1642  4421                    XDLPC2: ERRB   XDLOOP,PRECN1,@4
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4421  CD 12 28   18.0                       CALL   ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0003                                  MSGN   =      MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4424  03                                    .BYTE  MSGN            ;MESSAGE NUMBER ID
 (1)  4425  04                                    .BYTE  @4              ;PRINT ROUTINE NUMBER
 (1)  4426  CD 15 28   18.0               PRECN1::      CALL   CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4429  DA 2F 44   10.0                       JC     XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1643                            ;>DATA STATE RECEIVED FROM THE XMC INCORRECT
1644  442C  C3 4A 44   10.0          JMP    XDLPC           ;SKIP OVER ERROR LOOP CONTROL
1645
1646
1647
1648                            ;HERE IS THE ERROR LOOP CONTROL
1649
1650  442F  3A 94 4F   13.0  XDLOOP: LDA    LPFLG           ;GET THE LOOP ON ERROR FLAG
1651  4432  A7           4.0          ANA    A
1652  4433  C2 3C 44   10.0          JNZ    XDLP1           ;JUMP IF FLAG IS SET
1653  4436  32 82 46   13.0          STA    EXDLPC          ;ELSE - CLEAR THE COUNTER
1654  4439  C3 4A 44   10.0          JMP    XDLPC           ;AND CONTINUE TESTING
```

```
1655
1656  443C  3A 85 46    13.0  XDLP1:  LDA   TRANSCNT        ;GET THE LOW BYTE TRANSITION COUNT
1657  443F  47           4.0          MOV   B,A             ;TEMP SAVE
1658  4440  3A 82 46    13.0          LDA   EXDLPC          ;GET THE TRANS. COUNT AT ERROR
1659  4443  B8           4.0          CMP   B               ;SAME?
1660  4444  CA 11 43    10.0          JZ    T1LOOP          ;YUP - LOOP NOW!
1661  4447  C3 4A 44    10.0          JMP   XDLPC           ;NO - CONTINUE TILL A MATCH...
1662
1663  444A  DB 48        10.0  XDLPC:  IN    PSTAT           ;GET THE PORT ADDRESS
1664  444C  0F           4.0          RRC                   ;RIGHT JUSTIFY
1665  444D  0F           4.0          RRC                   ;
1666  444E  E6 01        7.0          ANI   $01             ;REMOVE OTHER BITS
1667  4450  57           4.0          MOV   D,A             ;STORE ACTUAL PARITY IN REG D
1668
1669  4451                            ROUT  ADATA           ;STORE ACTUAL PARITY IN CAS
 (1)  4451  D3 94        10.0                OUT     ADATA         ;WRITE AC INTO ADATA
 (1)  4453  7F           4.0                 MOV     A,A           ;RETRY LINK
1670  4454  2A 83 46    16.0          LHLD  POINTER
1671  4457  01 FB 45    10.0          LXI   B,GXPDAT        ;LOAD BEGINNING ADDRESS OF EXPECTED DATA
1672  445A  78           4.0          MOV   A,B             ;MAKE THE TWOS COMPLEMENT OF REGISTER
1673  445B  2F           4.0          CMA                   ;PAIR B & C
1674  445C  47           4.0          MOV   B,A
1675  445D  79           4.0          MOV   A,C
1676  445E  2F           4.0          CMA
1677  445F  4F           4.0          MOV   C,A
1678  4460  03           6.0          INX   B
1679  4461  09          10.0          DAD   B               ;ADD POINTER AND BEGINNING OF DATA TABLE
1680  4462  D2 89 44    10.0          JNC   PPREAM          ;CHECK PREAMBLE PARITY TRACK
1681
1682  4465  2A 83 46    16.0          LHLD  POINTER         ;RESET H & L REGS
1683  4468  01 14 46    10.0          LXI   B,GDATEN        ;GET THE ENDING ADDRESS OF EXPECTED DATA
1684  446B  78           4.0          MOV   A,B             ;MAKE THE TWOS COMPLEMENT OF REGISTER
1685  446C  2F           4.0          CMA                   ;PAIR B & C
1686  446D  47           4.0          MOV   B,A
1687  446E  79           4.0          MOV   A,C
1688  446F  2F           4.0          CMA
1689  4470  4F           4.0          MOV   C,A
1690  4471  03           6.0          INX   B
1691  4472  09          10.0          DAD   B               ;ADD POINTER AND END OF DATA TABLE
1692  4473  DA 89 44    10.0          JC    PPOST           ;CHECK POSTAMBLE PARITY TRACK
1693
1694  4476  01 64 46    10.0  DPTST:  LXI   B,PXPTBL        ;POINT TO EXPECTED PARITY DATA TABLE
1695  4479  2A 80 46    16.0          LHLD  DPCNT           ;POINT TO THE TABLE OFFSET NUMBER
1696  447C  23           6.0          INX   H               ;UPDATE IT
1697  447D  22 80 46    16.0          SHLD  DPCNT           ;SAVE NEW COUNT
1698  4480  09          10.0          DAD   B               ;MAKE REAL POINTER
1699  4481  7E           7.0          MOV   A,M             ;GET EXPECTED PARITY STATE
1700  4482  A7           4.0          ANA   A               ;SET CONDITION BITS
1701  4483  FA D2 44    10.0          JM    PTYC            ;JMP IF NO COMPARE ON THIS DATA
1702  4486  C3 91 44    10.0          JMP   PTY             ;TEST THE PARITY BIT
1703
1704                                ;HERE TO CHECK THE PREAMBLE/POSTAMBLE PARITY TRACK BIT
1705
1706  4489                          PPOST:
```

```
1707  4489  3A  7F  46   13.0  PPREAM: LDA   XDATA          ;GET THE DATA EXPECTED
1708  448C  E6  01        7.0          ANI   1              ;PARITY TRACK IS SAME
1709  448E  C3  91  44   10.0          JMP   PTY            ;CHECK THE PARITY
1710
1711  4491  BA              4.0  PTY:    CMP   D              ;SEE IF EXPECTED=ACTUAL
1712  4492  CA  AE  44   10.0          JZ    PTSTOK         ;JMP IF OK
1713  4495                        ROU.  EDATA          ;SAVE THE EXPECTED DATA
 (1)  4495  D3  95        10.0                OUT   EDATA          ;WRITE AC INTO EDATA
 (1)  4497  7F             4.0                MOV   A,A            ;RETRY LINK
1714  4498  7A              4.0          MOV   A,D            ;ERROR - GET THE ACTUAL DATA
1715  4499                        POUT  ADATA          ;SAVE FOR PRINTOUT
 (1)  4499  D3  94        10.0                OUT   ADATA          ;WRITE AC INTU ADATA
 (1)  449B  7F             4.0                MOV   A,A            ;RETRY LINK
1716
1717  449C  3A  82  46   13.0  PTY1:   LDA   EXDLPC         ;GET THE COUNT AT THE ERROR
1718  449F  A7              4.0          ANA   A
1719  44AC  C2  A9  44   10.0          JNZ   PTY2           ;JUMP IF AT AN ERROR NOW
1720  44A3  3A  85  46   13.0          LDA   TRANSCNT       ;GET THE COUNT AT ERROR
1721  44A6  32  82  46   13.0          STA   EXDLPC         ;SAVE IT
1722  44A9                        PTY2:   ERRB  XPLOOP,PTSTOK,@4
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44A9  CD  12  28   18.0                CALL  ERLPB          ;PROCESS ERROR - DO 2.3
 (1)      0004                               MSGN  =     MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44AC  04                               .BYTE MSGN           ;MESSAGE NUMBER ID
 (1)  44AD  04                               .BYTE @4             ;PRINT ROUTINE NUMBER
 (1)  44AE  CD  15  28   18.0          PTSTOK::      CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44B1  DA  B7  44   10.0                JC    XPLOOP         ;LOOP ADDRESS IF LOOP SPECIFIED
1723                        ;>PARITY STATE RECEIVED FROM THE XMC INCORRECT
1724  44B4  C3  D2  44   10.0          JMP   PTYC           ;JMP OVER CODE
1725
1726  44B7  3A  94  4F   13.0  XPLOOP: LDA   LPFLG          ;GET THE LOOP ON ERROR FLAG
1727  44BA  A7              4.0          ANA   A
1728  44BB  C2  C4  44   10.0          JNZ   XPLP1          ;JUMP IF FLAG IS SET
1729  44BE  32  82  46   13.0          STA   EXDLPC         ;ELSE - CLEAR THE COUNTER
1730  44C1  C3  D2  44   10.0          JMP   PTYC           ;AND CONTINUE TESTING
1731
1732  44C4  3A  85  46   13.0  XPLP1:  LDA   TRANSCNT       ;GET THE LOW BYTE TRANSITION COUNT
1733  44C7  47              4.0          MOV   B,A            ;TEMP SAVE
1734  44C8  3A  82  46   13.0          LDA   EXDLPC         ;GET THE TRANS. COUNT AT ERROR
1735  44CB  B8              4.0          CMP   B              ;SAME?
1736  44CC  CA  11  43   10.0          JZ    T1LOOP         ;YUP - LOOP NOW!
1737  44CF  C3  D2  44   10.0          JMP   PTYC           ;NO - CONTINUE TILL A MATCH...
1738
1739  44D2  2A  83  46   16.0  PTYC:   LHLD  POINTER        ;NO-UPDATE THE
1740  44D5  23              6.0          INX   H              ;TABLE POINTER
1741  44D6  22  83  46   16.0          SHLD  POINTER        ;ADDRESS
1742
1743  44D9  06  0E             7.0          MVI   B,14           ;LOAD THE CLOCK COUNT
1744  44DB  CD  6A  45   18.0  1$:     CALL  CLKSYS         ;CLOCK THE WRITE PATH
1745  44DE  05              4.0          DCR   B              ;DECREMENT THE COUNT
1746  44DF  C2  DB  44   10.0          JNZ   1$             ;CONTINUE UNTIL 8 CLOCKS
1747  44E2  DB  D0        10.0          IN    WMCSTA
1748  44E4  E6  01             7.0          ANI   X.WCLK
1749  44E6  C2  EE  44   10.0          JNZ   PRECN3
```

```
1750  44E9                                        ERR     T1LOOP,PRECN3,@4
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44E9  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0005                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44EC  05                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44ED  04                                           .BYTE   @4
 (1)  44EE  CD  15  28      18.0                PRECN3::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44F1  DA  11  43      10.0                        JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1751                                    ;>TRANSLATOR "XL WCLK" NOT SET AFTER EIGHT CLOCK CYCLES
1752                                    ;>IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
1753
1754  44F4  2A  83  46      16.0                LHLD    POINTER         ;GET THE TABLE POINTER
1755  44F7  01  63  46      10.0                LXI     B,GTBLEN        ;GET END OF TABLE ADDRESS
1756  44FA  78              4.0                 MOV     A,B             ;FROM THE
1757  44FB  2F              4.0                 CMA                     ;TWO'S COMPLIMENT
1758  44FC  47              4.0                 MOV     B,A             ;OF THE TABLE END
1759  44FD  79              4.0                 MOV     A,C             ;ADDRESS
1760  44FE  2F              4.0                 CMA
1761  44FF  4F              4.0                 MOV     C,A
1762  4500  03              6.0                 INX     B
1763  4501  09              10.0                DAD     B               ;END OF TABLE?
1764  4502  DA  08  45      10.0                JC      RECEND          ;YES
1765  4505  C3  EB  43      10.0                JMP     BGNPRE
1766
1767  4508  06  FF          7.0         RECEND: MVI     B,$FF           ;LOAD THE TIMEOUT COUNT
1768  450A  CD  6A  45      18.0         1$:    CALL    CLKSYS          ;CLOCK THE SYSTEM
1769  450D  DB  D0          10.0                IN      WMCSTA          ;INPUT THE WMC STATUS
1770  450F  E6  40          7.0                 ANI     W.DONN          ;WMC DONE?
1771  4511  CA  24  45      10.0                JZ      RECCN1          ;YES-EXIT
1772  4514  DB  D0          10.0                IN      WMCSTA          ;GET WMC STATUS
1773  4516  E6  80          7.0                 ANI     X.DONN          ;XMC DONE
1774  4518  CA  48  45      10.0                JZ      SEQERR          ;YES-ERROR
1775  451B  05              4.0                 DCR     B               ;NO-DECREMENT THE COUNT
1776  451C  C2  0A  45      10.0                JNZ     1$              ;CONTINUE IF NOT ZERO
1777  451F                                      ERR     RECEND,RECCN1
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  451F  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0006                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4522  06                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4523  00                                           .BYTE
 (1)  4524  CD  15  28      18.0                RECCN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4527  DA  08  45      10.0                        JC      RECEND          ;LOOP ADDRESS IF LOOP SPECIFIED
1778                                    ;>'WMC NOT DONE" TIMEOUT WAITING FOR IT TO GO TO ZERO
1779
1780  452A  06  FF          7.0                 MVI     B,$FF           ;LOAD THE TIMEOUT COUNT
1781  452C  CD  6A  45      18.0         3$:    CALL    CLKSYS          ;CLOCK THE SYSTEM
1782  452F  DB  D0          10.0                IN      WMCSTA          ;INPUT THE WMC STATUS
1783  4531  E6  80          7.0                 ANI     X.DONN          ;XMC-DONE?
1784  4533  CA  3F  45      10.0                JZ      RECCN3          ;YES-EXIT
1785  4536  05              4.0                 DCR     B               ;NO-DECREMENT THE COUNT
1786  4537  C2  2C  45      10.0                JNZ     3$              ;CONTINUE IF NOT ZERO
```

```
1788  453A                                    ERR     T1LOOP,RECCN3
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  453A  CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0007                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  453D  07                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  453E  00                                       .BYTE
 (1)  453F  CD  15  28        18.0            RECCN3::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4542  DA  11  43        10.0                    JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1789                                    ;>''XMC NOT DONE'' TIMEOUT WAITING FOR IT TO GO TO ZERO
1790
1791  4545  C3  67  45        10.0            JMP     EXIT
1792
1793  4548                                    SEQERR: ERR     T1LOOP,RECCN2
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4548  CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0008                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  454B  08                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  454C  00                                       .BYTE
 (1)  454D  CD  15  28        18.0            RECCN2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4550  DA  11  43        10.0                    JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1794                                    ;>''XMC NOT DONE'' RESET BEFORE ''WMC NOT DONE'' RESET
1795
1796  4553                                            ENDTST  T1LOOP
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4553                                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4553  CD  06  28        18.0                    CALL    REQST
 (2)  4556  00                                       .BYTE                   ;DATA PATTERN NUMBER
 (2)  4557  00  00                                   .WORD                   ;SYSTEM '"' COUNT
 (2)  4559  00  00                                   .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  455B  00                                       .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  455C  07                                       .BYTE   7               ;REQUEST CODE
 (1)  455D  3A  9A  4F        13.0                    LDA     ITERA           ;GET ITERATION COUNT
 (1)  4560  3D                 4.0                    DCR     A               ;DOWNCOUNT
 (1)  4561  32  9A  4F        13.0                    STA     ITERA           ;SAVE COUNT
 (1)  4564  F2  11  43        10.0                    JP      T1LOOP          ;DO TEST UNTIL TILL = 0
1797
1798  4567  C3  18  28        10.0    EXIT:   JMP     TSTEND
1799
1800
1801  456A  3A  87  46        13.0    CLKSYS: LDA     CCTLWD          ;GET THE SOFTWARE CLOCK CONTROL IMAGE
1802  456D  F6  40             7.0            ORI     SSCLK           ;ADD IN 'CLK' BIT
1803  456F  D3  F0            10.0            OUT     CLKCTL          ;LOAD CLOCK CONTROL
1804  4571  E6  BF             7.0            ANI     @277            ;STRIP OFF CLOCK BIT
1805  4573  D3  F0            10.0            OUT     CLKCTL          ;LOAD CLOCK CONTROL WORD
1806  4575  C9                10.0            RET                     ;EXIT-CLOCK CYCLE
```

```
1808                                    .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
1809  4576                              $SUB
 (1)                                    ;*********************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
1810                                    ;*CLEAR ALL TU PORTS
1811  4576                              $D
 (1)                                    ;*********************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
1812                                    ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
1813                                    ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
1814                                    ;*AND LOOP MODES.
1815  4576                              $P
 (1)                                    ;*********************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
1816                                    ;*BGNSUB
1817                                    ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
1818                                    ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
1819                                    ;*    CLEAR PORT SELECT FOR TRANSPORT
1820                                    ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
1821                                    ;*    CLEAR PORT DIAGNOSTIC CONTROL
1822                                    ;*    CLEAR PORT AMTIE WORD
1823                                    ;*ENDSUB
1824  4576                              $
 (1)                                    ;*********************************************************************
1825  4576  F5              12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
1826  4577  C5              12.0        PUSH    B
1827  4578  06  00           7.0        MVI     B,0             ;START TO CLEAR AT PORT #0
1828
1829  457A  DB  E0          10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
1830  457C  E6  80           7.0        ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
1831  457E  B0               4.0        ORA     B               ;ADD IN THE SELECTED PORT #
1832  457F  D3  E0          10.0        OUT     MBSEL           ;RESET TO THIS PORT
1833  4581  3E  80           7.0        MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
1834  4583  D3  40          10.0        OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
1835  4585  AF               4.0        XRA     A               ;CLEAR TU COMMAND A
1836  4586  D3  40          10.0        OUT     TCMD            ;
1837  4588  3E  81           7.0        MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
1838  458A  D3  40          10.0        OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
1839  458C  3E  00           7.0        MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
1840  458E  D3  40          10.0        OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
1841  4590  AF               4.0        XRA     A
1842  4591  D3  44          10.0        OUT     TAMT            ;CLEAR AMTIE WORD
1843  4593  D3  48          10.0        OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
1844  4595  D3  4C          10.0        OUT     PENAB           ;CLEAR PORT ENABLE WORD
1845  4597  04               4.0        INR     B               ;POINT TO THE NEXT PORT TO CLEAR
1846  4598  78               4.0        MOV     A,B
1847  4599  FE  04           7.0        CPI     4               ;DONE?
1848  459B  C2  7A  45       10.0        JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
1849  459E  C1               10.0        POP     B               ;RESET B & C
1850  459F  F1               10.0        POP     PSW             ;ALL DONE
1851  45A0  C9               10.0        RET                     ;EXIT
```

```
1852
1853
1854
1855  45A1  FF                    GXPTBL: .BYTE   $FF             ;PREAMBLE TERM GROUP (10101)
1856                                                               ;CHARACTER 1 = ALL ONES
1857  45A2  FF                            .BYTE   $FF             ;PREAMBLE TERM GROUP
1858                                                               ;CHARACTER 2 = ALL ZEROS
1859  45A3  00                            .BYTE   $00             ;PREAMBLE TERM GROUP
1860                                                               ;CHARACTER 3 = ALL ONES
1861  45A4  00                            .BYTE   $00             ;PREAMBLE TERM GROUP
1862                                                               ;CHARACTER 4 = ALL ZEROS
1863  45A5  FF                            .BYTE   $FF             ;PREAMBLE TERM GROUP
1864                                                               ;CHARACTER 5 = ALL ONES
1865
1866  45A6  FF                            .BYTE   $FF             ;PREAMBLE SECOND GROUP (01111)
1867                                                               ;CHARACTER 1 = ALL ZEROS
1868  45A7  00                            .BYTE   $00             ;PREAMBLE SECOND GROUP
1869                                                               ;CHARACTER 2 = ALL ONES
1870  45A8  FF                            .BYTE   $FF             ;PREAMBLE SECOND GROUP
1871                                                               ;CHARACTER 3 = ALL ONES
1872  45A9  00                            .BYTE   $00             ;PREAMBLE SECOND GROUP
1873                                                               ;CHARACTER 4 = ALL ONES
1874  45AA  FF                            .BYTE   $FF             ;PREAMBLE SECOND GROUP
1875                                                               ;CHARACTER 5 = ALL ONES
1876                                                               ;
1877  45AB  00                            .BYTE   $00             ;PREAMBLE SYNC GROUP 1 (14 GRPS OF 11111)
1878                                                               ;CHARACTER 1 = ALL ONES
1879  45AC  FF                            .BYTE   $FF             ;PREAMBLE SYNC GROUP 1
1880                                                               ;CHARACTER 2 = ALL ONES
1881  45AD  00                            .BYTE   $00             ;PREAMBLE SYNC GROUP 1
1882                                                               ;CHARACTER 3 = ALL ONES
1883  45AE  FF                            .BYTE   $FF             ;PREAMBLE SYNC GROUP 1
1884                                                               ;CHARACTER 4 = ALL ONES
1885  45AF  00                            .BYTE   $00             ;PREAMBLE SYNC GROUP 1
1886                                                               ;CHARACTER 5 = ALL ONES
1887                                                               ;
1888  45B0  FF                            .BYTE   $FF             ;PREAMBLE SYNC GROUP 2
1889                                                               ;CHARACTER 1 = ALL ONES
1890  45B1  00                            .BYTE   $00             ;PREAMBLE SYNC GROUP 2
1891                                                               ;CHARACTER 2 = ALL ONES
1892  45B2  FF                            .BYTE   $FF             ;PREAMBLE SYNC GROUP 2
1893                                                               ;CHARACTER 3 = ALL ONES
1894  45B3  00                            .BYTE   $00             ;PREAMBLE SYNC GROUP 2
1895                                                               ;CHARACTER 4 = ALL ONES
1896  45B4  FF                            .BYTE   $FF             ;PREAMBLE SYNC GROUP 2
1897                                                               ;CHARACTER 5 = ALL ONES
1898                                                               ;
1899  45B5  00                            .BYTE   $00             ;PREAMBLE SYNC GROUP 3
1900                                                               ;CHARACTER 1 = ALL ONES
1901  45B6  FF                            .BYTE   $FF             ;PREAMBLE SYNC GROUP 3
1902                                                               ;CHARACTER 2 = ALL ONES
1903  45B7  00                            .BYTE   $00             ;PREAMBLE SYNC GROUP 3
1904                                                               ;CHARACTER 3 = ALL ONES
1905  45B8  FF                            .BYTE   $FF             ;PREAMBLE SYNC GROUP 3
```

```
1906
1907  45B9  00                    .BYTE   $00             ;CHARACTER 4 - ALL ONES
1908                                                      ;PREAMBLE SYNC GROUP 3
1909                                                      ;CHARACTER 5 = ALL ONES
1910  45BA  FF                    .BYTE   $FF             ;
1911                                                      ;PREAMBLE SYNC GROUP 4
1912  45BB  00                    .BYTE   $00             ;CHARACTER 1 = ALL ONES
1913                                                      ;PREAMBLE SYNC GROUP 4
1914  45BC  FF                    .BYTE   $FF             ;CHARACTER 2 = ALL ONES
1915                                                      ;PREAMBLE SYNC GROUP 4
1916  45BD  00                    .BYTE   $00             ;CHARACTER 3 = ALL ONES
1917                                                      ;PREAMBLE SYNC GROUP 4
1918  45BE  FF                    .BYTE   $FF             ;CHARACTER 4 = ALL ONES
1919                                                      ;PREAMBLE SYNC GROUP 4
1920                                                      ;CHARACTER 5 = ALL ONES
1921  45BF  00                    .BYTE   $00             ;
1922                                                      ;PREAMBLE SYNC GROUP 5
1923  45C0  FF                    .BYTE   $FF             ;CHARACTER 1 = ALL ONES
1924                                                      ;PREAMBLE SYNC GROUP 5
1925  45C1  00                    .BYTE   $00             ;CHARACTER 2 = ALL ONES
1926                                                      ;PREAMBLE SYNC GROUP 5
1927  45C2  FF                    .BYTE   $FF             ;CHARACTER 3 = ALL ONES
1928                                                      ;PREAMBLE SYNC GROUP 5
1929  45C3  00                    .BYTE   $00             ;CHARACTER 4 - ALL ONES
1930                                                      ;PREAMBLE SYNC GROUP 5
1931                                                      ;CHARACTER 5 = ALL ONES
1932  45C4  FF                    .BYTE   $FF             ;
1933                                                      ;PREAMBLE SYNC GROUP 6
1934  45C5  00                    .BYTE   $00             ;CHARACTER 1 = ALL ONES
1935                                                      ;PREAMBLE SYNC GROUP 6
1936  45C6  FF                    .BYTE   $FF             ;CHARACTER 2 = ALL ONES
1937                                                      ;PREAMBLE SYNC GROUP 6
1938  45C7  00                    .BYTE   $00             ;CHARACTER 3 = ALL ONES
1939                                                      ;PREAMBLE SYNC GROUP 6
1940  45C8  FF                    .BYTE   $FF             ;CHARACTER 4 = ALL ONES
1941                                                      ;PREAMBLE SYNC GROUP 6
1942                                                      ;CHARACTER 5 = ALL ONES
1943  45C9  00                    .BYTE   $00             ;
1944                                                      ;PREAMBLE SYNC GROUP 7
1945  45CA  FF                    .BYTE   $FF             ;CHARACTER 1 = ALL ONES
1946                                                      ;PREAMBLE SYNC GROUP 7
1947  45CB  00                    .BYTE   $00             ;CHARACTER 2 = ALL ONES
1948                                                      ;PREAMBLE SYNC GROUP 7
1949  45CC  FF                    .BYTE   $FF             ;CHARACTER 3 = ALL ONES
1950                                                      ;PREAMBLE SYNC GROUP 7
1951  45CD  00                    .BYTE   $00             ;CHARACTER 4 - ALL ONES
1952                                                      ;PREAMBLE SYNC GROUP 7
1953                                                      ;CHARACTER 5 = ALL ONES
1954  45CE  FF                    .BYTE   $FF             ;
1955                                                      ;PREAMBLE SYNC GROUP 8
1956  45CF  00                    .BYTE   $00             ;CHARACTER 1 = ALL ONES
1957                                                      ;PREAMBLE SYNC GROUP 8
1958  45D0  FF                    .BYTE   $FF             ;CHARACTER 2 = ALL ONES
1959                                                      ;PREAMBLE SYNC GROUP 8
                                                          ;CHARACTER 3 = ALL ONES
```

```
1960  45D1  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 8
1961                                     ;CHARACTER 4 = ALL ONES
1962  45D2  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 8
1963                                     ;CHARACTER 5 = ALL ONES
1964                                     ;
1965  45D3  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 9
1966                                     ;CHARACTER 1 = ALL ONES
1967  45D4  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 9
1968                                     ;CHARACTER 2 = ALL ONES
1969  45D5  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 9
1970                                     ;CHARACTER 3 = ALL ONES
1971  45D6  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 9
1972                                     ;CHARACTER 4 = ALL ONES
1973  45D7  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 9
1974                                     ;CHARACTER 5 = ALL ONES
1975                                     ;
1976  45D8  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 10
1977                                     ;CHARACTER 1 = ALL ONES
1978  45D9  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 10
1979                                     ;CHARACTER 2 = ALL ONES
1980  45DA  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 10
1981                                     ;CHARACTER 3 = ALL ONES
1982  45DB  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 10
1983                                     ;CHARACTER 4 = ALL ONES
1984  45DC  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 10
1985                                     ;CHARACTER 5 = ALL ONES
1986                                     ;
1987  45DD  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 11
1988                                     ;CHARACTER 1 = ALL ONES
1989  45DE  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 11
1990                                     ;CHARACTER 2 = ALL ONES
1991  45DF  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 11
1992                                     ;CHARACTER 3 = ALL ONES
1993  45E0  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 11
1994                                     ;CHARACTER 4 = ALL ONES
1995  45E1  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 11
1996                                     ;CHARACTER 5 = ALL ONES
1997                                     ;
1998  45E2  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 12
1999                                     ;CHARACTER 1 = ALL ONES
2000  45E3  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 12
2001                                     ;CHARACTER 2 = ALL ONES
2002  45E4  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 12
2003                                     ;CHARACTER 3 = ALL ONES
2004  45E5  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 12
2005                                     ;CHARACTER 4 = ALL ONES
2006  45E6  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 12
2007                                     ;CHARACTER 5 = ALL ONES
2008                                     ;
2009  45E7  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 13
2010                                     ;CHARACTER 1 = ALL ONES
2011  45E8  FF          .BYTE  $FF      ;PREAMBLE SYNC GROUP 13
2012                                     ;CHARACTER 2 = ALL ONES
2013  45E9  00          .BYTE  $00      ;PREAMBLE SYNC GROUP 13
```

```
2014
2015  45EA  FF                        .BYTE   $FF           ;CHARACTER 3 = ALL ONES
                                                           ;PREAMBLE SYNC GROUP 13
2016
                                                           ;CHARACTER 4 - ALL ONES
2017  45EB  00                        .BYTE   $00           ;PREAMBLE SYNC GROUP 13
2018
                                                           ;CHARACTER 5 = ALL ONES
2019
                                                           ;
2020  45EC  FF                        .BYTE   $FF           ;PREAMBLE SYNC GROUP 14
2021
                                                           ;CHARACTER 1 = ALL ONES
2022  45ED  00                        .BYTE   $00           ;PREAMBLE SYNC GROUP 14
2023
                                                           ;CHARACTER 2 = ALL ONES
2024  45EE  FF                        .BYTE   $FF           ;PREAMBLE SYNC GROUP 14
2025
                                                           ;CHARACTER 3 = ALL ONES
2026  45EF  00                        .BYTE   $00           ;PREAMBLE SYNC GROUP 14
2027
                                                           ;CHARACTER 4 = ALL ONES
2028  45F0  FF                        .BYTE   $FF           ;PREAMBLE SYNC GROUP 14
2029
                                                           ;CHARACTER 5 = ALL ONES
2030
                                                           ;
2031  45F1  FF                        .BYTE   $FF           ;MARK ONE GROUP
2032
                                                           ;CHARACTER 1 = ALL ZEROS
2033  45F2  FF                        .BYTE   $FF           ;MARK ONE GROUP
2034
                                                           ;CHARACTER 2 = ALL ZEROS
2035  45F3  00                        .BYTE   $00           ;MARK ONE GROUP
2036
                                                           ;CHARACTER 3 = ALL ONES
2037  45F4  FF                        .BYTE   $FF           ;MARK ONE GROUP
2038
                                                           ;CHARACTER 4 = ALL ONES
2039  45F5  00                        .BYTE   $00           ;MARK ONE GROUP
2040
                                                           ;CHARACTER 5 = ALL ONES
2041
                                                           ;
2042                              ;THERE IS NO DATA GROUP ... ONLY 1 DATA BYTE IN THE RESIDUAL GROUP.
2043                              ;ECC FOR DATA GRP = 0.
2044                              ;THIS IS TRANSITION # 86.
2045
2046                              ;END MARK = 11111 (ALL 1'S)
2047
2048  45F6  FF                            .BYTE   $FF       ;END MARK GROUP
2049
                                                           ;CHARACTER 1 = ALL ONES
2050  45F7  00                            .BYTE   $00       ;END MARK GROUP
2051
                                                           ;CHARACTER 2 = ALL ONES
2052  45F8  FF                            .BYTE   $FF       ;END MARK GROPU
2053
                                                           ;CHARACTER 3 = ALL ONES
2054  45F9  00                            .BYTE   $00       ;END MARK GROUP
2055
                                                           ;CHARACTER 4 = ALL ONES
2056  45FA  FF                            .BYTE   $FF       ;END MARK GROUP
2057
                                                           ;CHARACTER 5 = ALL ONES
2058
                                                           ;
2059
2060
2061                              ;1 DATA BYTE AND 3 PAD BYTES OF ALL 0'S (11001)
2062                              ;TRANSITION #91.
2063
2064  45FB  00                    GXPDAT: .BYTE   @0        ;RESIDUAL GROUP - AFTER TRANSLATION - 1ST HALF
2065
                                                           ;CHARACTER 1
2066  45FC  FF                            .BYTE   $FF       ;RESIDUAL GROUP - AFTER TRANSLATION
2067
                                                           ;CHARACTER 2
```

```
2068  45FD  FF                        .BYTE   $FF             ;RESIDUAL GROUP - AFTER TRANSLATION
2069                                                          ;CHARACTER 3
2070  45FE  FF                        .BYTE   $FF             ;RESIDUAL GROUP - AFTER TRANSLATION
2071                                                          ;CHARACTER 4
2072  45FF  00                        .BYTE   @0              ;RESIDUAL GROUP - AFTER TRANSLATION
2073                                                          ;CHARACTER 5
2074
2075                          ;2 PAD BYTES
2076                          ;ACRC = XXX
2077                          ;ECC = XXX
2078                          ;TRANSITION #96.
2079
2080  4600  FF                        .BYTE   @377            ;RESIDUAL GROUP - AFTER TRANSLATION - 2ND HALF
2081                                                          ;CHARACTER 1
2082  4601  BD                        .BYTE   @275            ;RESIDUAL GROUP - AFTER TRANSLATION
2083                                                          ;CHARACTER 2
2084  4602  BD                        .BYTE   @275            ;RESIDUAL GROUP - AFTER TRANSLATION
2085                                                          ;CHARACTER 3
2086  4603  42                        .BYTE   @102            ;RESIDUAL GROUP - AFTER TRANSLATION
2087                                                          ;CHARACTER 4
2088  4604  A4                        .BYTE   @244            ;RESIDUAL GROUP - AFTER TRANSLATION
2089                                                          ;CHARACTER 5
2090                                                          ;
2091                          ;4 CRC BYTES OF XXX
2092                          ;TRANSITION #101.
2093
2094  4605  10                        .BYTE   @020            ;CRC GROUP - AFTER TRANSLATION - 1ST HALF
2095                                                          ;CHARACTER 1
2096  4606  EF                        .BYTE   @357            ;CRC GROUP - AFTER TRANSLATION
2097                                                          ;CHARACTER 2
2098  4607  A4                        .BYTE   @244            ;CRC GROUP - AFTER TRANSLATION
2099                                                          ;CHARACTER 3
2100  4608  EF                        .BYTE   @357            ;CRC GROUP - AFTER TRANSLATION
2101                                                          ;CHARACTER 4
2102  4609  10                        .BYTE   @020            ;CRC GROUP - AFTER TRANSLATION
2103                                                          ;CHARACTER 5
2104
2105                          ;2 CRC OF XXX
2106                          ;RESID OF 000
2107                          ;ECC OF XXX
2108                          ;TRANSITION #106.
2109
2110  460A  E6                        .BYTE   @346            ;CRC GROUP - AFTER TRANSLATION - 2ND HALF
2111                                                          ;CHARACTER 6
2112  460B  39                        .BYTE   @071            ;CRC GROUP - AFTER TRANSLATION
2113                                                          ;CHARACTER 7
2114  460C  72                        .BYTE   @162            ;CRC GROUP - AFTER TRANSLATION
2115                                                          ;CHARACTER 8
2116  460D  14                        .BYTE   @024            ;CRC GROUP - AFTER TRANSLATION
2117                                                          ;CHARACTER 9
2118  460E  A9                        .BYTE   @251            ;CRC GROUP - AFTER TRANSLATION
2119                                                          ;CHARACTER 10
2120                                                          ;
2121
```

```
2122
2123  460F  56                        .BYTE   @126          ;MARK TWO
2124                                                        ;CHARACTER 1 = ALL ONES
2125  4610  A9                        .BYTE   @251          ;MARK TWO
2126                                                        ;CHARACTER 2 = ALL ONES
2127  4611  56                        .BYTE   @126          ;MARK TWO
2128                                                        ;CHARACTER 3 = ALL ONES
2129  4612  56                        .BYTE   @126          ;MARK TWO
2130                                                        ;CHARACTER 4 = ALL ZEROS
2131  4613  56                        .BYTE   @126          ;MARK TWO
2132                                                        ;CHARACTER 5 = ALL ZEROS
2133                                                        ;
2134  4614  A9                  GDATEN: .BYTE   @251          ;POSTAMBLE SYNC GROUP 1
2135                                                        ;CHARACTER 1 = ALL ONES
2136  4615  56                        .BYTE   @126          ;POSTAMBLE SYNC GROUP 1
2137                                                        ;CHARACTER 2 - ALL ONES
2138  4616  A9                        .BYTE   @251          ;POSTAMBLE SYNC GROUP 1
2139                                                        ;CHARACTER 3 = ALL ONES
2140  4617  56                        .BYTE   @126          ;POSTAMBLE SYNC GROUP 1
2141                                                        ;CHARACTER 4 = ALL ONES
2142  4618  A9                        .BYTE   @251          ;POSTAMBLE SYNC GROUP 1
2143                                                        ;CHARACTER 5 = ALL ONES
2144                                                        ;
2145  4619  56                        .BYTE   @126          ;POSTAMBLE SYNC GROUP 2
2146                                                        ;CHARACTER 1 = ALL ONES
2147  461A  A9                        .BYTE   @251          ;POSTAMBLE SYNC GROUP 2
2148                                                        ;CHARACTER 2 - ALL ONES
2149  461B  56                        .BYTE   @126          ;POSTAMBLE SYNC GROUP 2
2150                                                        ;CHARACTER 3 = ALL ONES
2151  461C  A9                        .BYTE   @251          ;POSTAMBLE SYNC GROUP 2
2152                                                        ;CHARACTER 4 = ALL ONES
2153  461D  56                        .BYTE   @126          ;POSTAMBLE SYNC GROUP 2
2154                                                        ;CHARACTER 5 = ALL ONES
2155                                                        ;
2156  461E  A9                        .BYTE   @251          ;POSTAMBLE SYNC GROUP 3
2157                                                        ;CHARACTER 1 = ALL ONES
2158  461F  56                        .BYTE   @126          ;POSTAMBLE SYNC GROUP 3
2159                                                        ;CHARACTER 2 - ALL ONES
2160  4620  A9                        .BYTE   @251          ;POSTAMBLE SYNC GROUP 3
2161                                                        ;CHARACTER 3 = ALL ONES
2162  4621  56                        .BYTE   @126          ;POSTAMBLE SYNC GROUP 3
2163                                                        ;CHARACTER 4 = ALL ONES
2164  4622  A9                        .BYTE   @251          ;POSTAMBLE SYNC GROUP 3
2165                                                        ;CHARACTER 5 = ALL ONES
2166                                                        ;
2167  4623  56                        .BYTE   @126          ;POSTAMBLE SYNC GROUP 4
2168                                                        ;CHARACTER 1 = ALL ONES
2169  4624  A9                        .BYTE   @251          ;POSTAMBLE SYNC GROUP 4
2170                                                        ;CHARACTER 2 - ALL ONES
2171  4625  56                        .BYTE   @126          ;POSTAMBLE SYNC GROUP 4
2172                                                        ;CHARACTER 3 = ALL ONES
2173  4626  A9                        .BYTF   @251          ;POSTAMBLE SYNC GROUP 4
2174                                                        ;CHARACTER 4 = ALL ONES
2175  4627  56                        .BYTE   @126          ;POSTAMBLE SYNC GROUP 4
```

```
2176                                                    ;CHARACTER 5 = ALL ONES
2177                                                    ;
2178  4628  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 5
2179                                                    ;CHARACTER 1 = ALL ONES
2180  4629  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 5
2181                                                    ;CHARACTER 2 - ALL ONES
2182  462A  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 5
2183                                                    ;CHARACTER 3 = ALL ONES
2184  462B  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 5
2185                                                    ;CHARACTER 4 = ALL ONES
2186  462C  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 5
2187                                                    ;CHARACTER 5 = ALL ONES
2188                                                    ;
2189  462D  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 6
2190                                                    ;CHARACTER 1 = ALL ONES
2191  462E  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 6
2192                                                    ;CHARACTER 2 - ALL ONES
2193  462F  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 6
2194                                                    ;CHARACTER 3 = ALL ONES
2195  4630  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 6
2196                                                    ;CHARACTER 4 = ALL ONES
2197  4631  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 6
2198                                                    ;CHARACTER 5 = ALL ONES
2199                                                    ;
2200  4632  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 7
2201                                                    ;CHARACTER 1 = ALL ONES
2202  4633  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 7
2203                                                    ;CHARACTER 2 - ALL ONES
2204  4634  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 7
2205                                                    ;CHARACTER 3 = ALL ONES
2206  4635  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 7
2207                                                    ;CHARACTER 4 = ALL ONES
2208  4636  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 7
2209                                                    ;CHARACTER 5 = ALL ONES
2210                                                    ;
2211  4637  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 8
2212                                                    ;CHARACTER 1 = ALL ONES
2213  4638  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 8
2214                                                    ;CHARACTER 2 - ALL ONES
2215  4639  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 8
2216                                                    ;CHARACTER 3 = ALL ONES
2217  463A  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 8
2218                                                    ;CHARACTER 4 = ALL ONES
2219  463B  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 8
2220                                                    ;CHARACTER 5 = ALL ONES
2221                                                    ;
2222  463C  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 9
2223                                                    ;CHARACTER 1 = ALL ONES
2224  463D  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 9
2225                                                    ;CHARACTER 2 - ALL ONES
2226  463E  A9                    .BYTE   @251          ;POSTAMBLE SYNC GROUP 9
2227                                                    ;CHARACTER 3 = ALL ONES
2228  463F  56                    .BYTE   @126          ;POSTAMBLE SYNC GROUP 9
2229                                                    ;CHARACTER 4 = ALL ONES
```

```
2230  4640  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 9
2231                                        ;CHARACTER 5 = ALL ONES
2232                                        ;
2233  4641  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 10
2234                                        ;CHARACTER 1 = ALL ONES
2235  4642  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 10
2236                                        ;CHARACTER 2 - ALL ONES
2237  4643  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 10
2238                                        ;CHARACTER 3 = ALL ONES
2239  4644  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 10
2240                                        ;CHARACTER 4 = ALL ONES
2241  4645  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 10
2242                                        ;CHARACTER 5 = ALL ONES
2243                                        ;
2244  4646  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 11
2245                                        ;CHARACTER 1 = ALL ONES
2246  4647  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 11
2247                                        ;CHARACTER 2 - ALL ONES
2248  4648  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 11
2249                                        ;CHARACTER 3 = ALL ONES
2250  4649  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 11
2251                                        ;CHARACTER 4 = ALL ONES
2252  464A  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 11
2253                                        ;CHARACTER 5 = ALL ONES
2254                                        ;
2255  464B  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 12
2256                                        ;CHARACTER 1 = ALL ONES
2257  464C  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 12
2258                                        ;CHARACTER 2 - ALL ONES
2259  464D  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 12
2260                                        ;CHARACTER 3 = ALL ONES
2261  464E  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 12
2262                                        ;CHARACTER 4 = ALL ONES
2263  464F  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 12
2264                                        ;CHARACTER 5 = ALL ONES
2265                                        ;
2266  4650  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 13
2267                                        ;CHARACTER 1 = ALL ONES
2268  4651  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 13
2269                                        ;CHARACTER 2 - ALL ONES
2270  4652  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 13
2271                                        ;CHARACTER 3 = ALL ONES
2272  4653  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 13
2273                                        ;CHARACTER 4 = ALL ONES
2274  4654  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 13
2275                                        ;CHARACTER 5 = ALL ONES
2276                                        ;
2277  4655  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 14
2278                                        ;CHARACTER 1 = ALL ONES
2279  4656  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 14
2280                                        ;CHARACTER 2 - ALL ONES
2281  4657  56          .BYTE   @126        ;POSTAMBLE SYNC GROUP 14
2282                                        ;CHARACTER 3 = ALL ONES
2283  4658  A9          .BYTE   @251        ;POSTAMBLE SYNC GROUP 14
```

```
2284
2285  4659  56                          .BYTE   @126            ;CHARACTER 4 = ALL ONES
2286                                                            ;POSTAMBLE SYNC GROUP 14
2287                                                            ;CHARACTER 5 = ALL ONES
2288  465A  A9                          .BYTE   @251            ;
2289                                                            ;POSTAMBLE SECOND GROUP
2290  465B  56                          .BYTE   @126            ;CHARACTER 1 = ALL ONES
2291                                                            ;POSTAMBLE SECOND GROUP
2292  465C  A9                          .BYTE   @251            ;CHARACTER 2 = ALL ONES
2293                                                            ;POSTAMBLE SECOND GROUP
2294  465D  56                          .BYTE   @126            ;CHARACTER 3 = ALL ONES
2295                                                            ;POSTAMBLE SECOND GROUP
2296  465E  56                          .BYTE   @126            ;CHARACTER 4 = ALL ONES
2297                                                            ;POSTAMBLE SECOND GROUP
2298                                                            ;CHARACTER 5 = ALL ZEROS
2299  465F  A9                          .BYTE   @251            ;
2300                                                            ;POSTAMBLE TERM GROUP
2301  4660  A9                          .BYTE   @251            ;CHARACTER 1 = ALL ONES
2302                                                            ;POSTAMBLE TERM GROUP
2303  4661  56                          .BYTE   @126            ;CHARACTER 2 = ALL ZEROS
2304                                                            ;POSTAMBLE TERM GROUP
2305  4662  56                          .BYTE   @126            ;CHARACTER 3 = ALL ONES
2306                                                            ;POSTAMBLE TERM GROUP
2307  4663  56                GTBLEN:   .BYTE   @126            ;CHARACTER 4 = ALL ZEROS
2308                                                            ;POSTAMBLE TERM GROUP
                                                                ;CHARACTER 5 = 0'S TO RESET TRACKS
```

```
2311                                    .SBTTL  TABLE EXPECTED PARITY
2312
2313                                    ;HERE IS THE TABLE OF EXPECTED PARITY STATES FOR RESIDUAL GROUP
2314                                    ;NO BYTES IN DATA GROUP.
2315
2316  4664   00                         PXPTBL: .BYTE   0               ;THIS IS A DUMMY ENTRY
2317
2318                                    ;RESIDUAL GRP
2319                                    ;1ST GRP IS 1 DATA BYTE OF 0 AND 3 PADS SO 4 X 5 IS 0000 = 11001 FOR DATA
2320                                    ;AND PARITY IS 1111 = 0111
2321
2322  4665   01                               .BYTE   1
2323  4666   00                               .BYTE   0
2324  4667   01                               .BYTE   1
2325  4668   00                               .BYTE   0
2326  4669   01                               .BYTE   1
2327
2328                                    ;2ND GRP IS 2 PAD BYTES + ACRC OF XXX + ECC OF XXX
2329                                    ;'P' 4 X 5 IS
2330
2331  466A   01                               .BYTE   1
2332  466B   00                               .BYTE   0
2333  466C   01                               .BYTE   1
2334  466D   00                               .BYTE   0
2335  466E   00                               .BYTE   0
2336
2337                                    ;CRC GRP 1ST HALF IS 4 CRC BYTES OF XXX
2338                                    ;'P' 4 X 5 IS
2339
2340  466F   00                               .BYTE   0
2341  4670   01                               .BYTE   1
2342  4671   00                               .BYTE   0
2343  4672   01                               .BYTE   1
2344  4673   00                               .BYTE   0
2345
2346                                    ;2ND HALF OF CRC GRP IS 2 CRC OF XXX + RESID BYTE OF XXX + ECC BYTE OF XXX
2347                                    ;'P' 4 X 5 IS
2348
2349  4674   00                               .BYTE   0
2350  4675   01                               .BYTE   1
2351  4676   00                               .BYTE   0
2352  4677   00                               .BYTE   0
2353  4678   01                               .BYTE   1
2354
2355                                    ;HERE IS THE MARK-2 PARITY TABLE
2356
2357  4679   00                               .BYTE   0               ;MARK 2 IS 11100
2358  467A   01                               .BYTE   1
2359  467B   00                               .BYTE   0
2360  467C   00                               .BYTE   0
2361  467D   00                               .BYTE   0
```

```
                                        .SBTTL  PROGRAM VARIABLES

467E  00                        UNITMP: .BYTE   0               ;UNIT MAP
467F  00                        XDATA:  .BYTE   0               ;COPY OF EXPECTED DATA
4680  00  00                     DPCNT:  .WORD   0               ;PARITY TEST COUNTER
4682  00                        EXDLPC: .BYTE   0               ;TRANSITION COUNT AT ERROR
4683  00  00                     POINTER: .WORD  0               ;POINTER TO EXPECTED DATA TABLE
4685  00  00                     TRANSCNT: .WORD 0               ;TRANSITION COUNT
4687  00                        CCTLWD: .BYTE   0               ;CLOCK CONTROL WORD

                                        .END

0000
```

```
A      =%0007          ABOCNO  43E2  G        ADATA  = 0094          AMTIEP= 0001
AMTIE7= 0002           ARAIDF= 0098           ASAVE   4F9B           ATTCD   4F97
AXNUM   4F91           B      =%0000          BADST  = 0090          BGNPRE  43EB
BIT0   = 0001          BIT1   = 0002          BIT15  = 8000          BIT2   = 0004
BIT3   = 0008          BIT4   = 0010          BIT5   = 0020          BIT6   = 0040
BIT7   = 0080          BIT8   = 0100          BRKXCT= 4F00           BRKPBC= 4F0A
BRKRAM= 4F10           BRKSTR= 4E60           BYTEL   4F23           BSAVE   4F9C
BYTCNT= 00D4           BYTEH   4F24           CASSTA= 00A0           C      =%0001
CASCT   4F21           CASCTL= 00A0           CBYTH  = 008B          CATTH  = 0089
CATTL  = 0088          CBUSST= 00A1           CDG2H  = 0093          CBYTL  = 008A
CCTLWD  4687           CDG1H  = 0087          CDG3L  = 0094          CDVTH  = 008D
CDG2L  = 0092          CDG3H  = 0095          CH0TIE= 0020           CH1TIE= 0021
CDVTL  = 008C          CHPTIE= 0028           CH4TIE= 0024           CH5TIE= 0025
CH2TIE= 0022           CH3TIE= 0023           CKLOP  = 2815          CLEAR   4576
CH6TIE= 0026           CH7TIE= 0027           CLKRS   43C2           CLKSYS  456A
CLK     43CF           CLKCTL= 00F0           CLRLP   457A           CMC0H  = 0099
CLK1    43CD           CLOCK   4F26           CMC3L  = 009E          CMC2H  = 009D
CMC0L  = 0098          CMC1H  = 009B          CRCWRD= 0018           CMINH  = 0097
CMC2L  = 009C          CMC3H  = 009F          CTCH   = 0085          CSAVE   4F9D
CMINL  = 0096          CNTCTL= 00D7           CXCTH  = 0081          CTCL   = 0084
CSRLH  = 0091          CSRLL  = 0090          C.     = 0001          CXCTL  = 0080
CTSTH  = 008F          CTSTL  = 008E          C.DTU  = 0003          C.AVAI= 0080
CXINH  = 0083          CXINL  = 0082          C.FNCT= 003E           C.DVA  = 0008
C.DP   = 0008          C.DSE  = 0010          C.NSA  = 0080          C.GO   = 0001
C.FAIL= 00FC           C.FMT  = 0070          C.SKPC= 000F           C.RCT  = 00FC
C.INTC= 00FE           C.MAIN= 0020           DATACT= 00D0           C.TAPE= 0040
C.SER  = 0080          C.SHR  = 0040          DDRA   = 00D8          DBUS    4F28
C.WCS  = 0002          D      =%0002          DDRC   = 00DA          DDRAIN= 0010
DBUSCT= 00C0           DBUSST= 00C0           DIAFLG  4F22           DDRCIN= 0001
DDRB   = 00D9          DDRBIN= 0002           DONE1  = 0045          DIAGPG= 4300
DDRCO  = 0088          DDRCTL= 00DB           DSAVE   4F9E           DONINT= 0010
DIAGRM= 4F90           DIARD  = 000B          D.ATH1= 0002           DSE    = 0006
DPCNI   4680           DPTST   4476           D.NTHR= 0004           D.EOTD= 0010
DUMMY   4367  G        D.ATH0= 0001           ECCBAD= 0042           D.TACH= 0008
D.LAGC= 0020           D.NOTW= 0040           ECCTST= 000E           ECCCOR= 0019
D.WR4  = 0080          E      =%0003          ERLP   = 2809          EDATA  = 0095
ECCOK  = 0041          ECCSTA= 001A           ERNUM   4F90           ERLPA  = 280F
EOTCLR= 0003           ERFLG   4F93           EXIT    4567           ERRCNT= 00D6
ERLPB  = 2812          ERLPE  = 280C          FOUND   4370           E.ACRC= 0010
ESAVE   4F9F           EXDLPC  4682           GDATEN  4614           E.PNTR= 0008
E.AMT  = 0020          E.CDP  = 0080          GXPTBL  45A1           E.UNC  = 0004
E.RPE  = 0040          E.STEC= 0001           INTSTA= 00E0           FWDTST= 0061
FIFORD= 006A           FORMAT  4F25           I5.5   = 0010          GOODTM= 0092
GCRID  = 0089          GCRSET= 0002           KCLR   = 007B          H      =%0004
GTBLEN  4663           GXPDAT  45FB           KEYBRD= 00C8           ITERA   4F9A
HLSAVE  4FA0           IE     = 0008          KEY12  = 006F          I6.5   = 0020
I.PWR  = 0020          I.RMPE= 0040           KEY16  = 005F          KDEP   = 003F
I7.5   = 0040          KCALL  = 005F          KEY2   = 0079          KEY1   = 0078
KENAB  = 0078          KEXAM  = 003E          KEY5   = 0074          KEY13  = 005C
KEY10  = 006D          KEY11  = 006E          KEY9   = 006C          KEY17  = 003C
KEY14  = 005D          KEY15  = 005E          KINTA  = 006F          KEY20  = 003F
KEY18  = 003D          KEY19  = 003E                                 KEY6   = 0075
KEY3   = 007A          KEY4   = 007B
KEY7   = 0076          KEY8   = 0077
```

```
KLDAD = 003D            KN0   = 003C            KN1   = 005C            KN2   = 005D
KN3   = 005E            KN4   = 006C            KN5   = 006D            KN6   = 006E
KN7   = 0074            KN8   = 0075            KN9   = 0076            KU2   = 0079
KU3   = 007A            KU8   = 0077            L     =%0005            LBLANK= 000F
LCE   = 000B            LCH   = 000C            LCL   = 000D            LCP   = 000E
LC0   = 0000            LC1   = 0001            LC2   = 0002            LC3   = 0003
LC4   = 0004            LC5   = 0005            LC6   = 0006            LC7   = 0007
LC8   = 0008            LC9   = 0009            LDLEDA= 00CA            LDLEDB= 00CB
LDLEDC= 00CC            LDLEDD= 00CD            LDLEDE= 00CE            LDLEDF= 00CF
LKDIAG= 2800            LKKBD = 004C            LKKEY = 0049            LKLWMG= 0058
LKLWMP= 0055            LKLWPG= 0052            LKLWPP= 004F            LKMOD7= 0046
LKOPR = 0046            LPFLG   4F94            LPNUM   4F92            M     =%0006
MBSEL = 00E0            MB.A  = 0008            MB.B  = 0004            MEMTOP= 4FFF
MINUS = 000A            MM    = 8000            MSE   = 0008            MSGN  = 0008
MTACLR= 0000            MT.ARA= 0020            MT.CPE= 0080            MT.DSE= 0001
MT.FWD= 0040            MT.INH= 0008            MT.LWR= 0004            MT.MOT= 0002
MT.NWT= 0080            MT.PEC= 0040            MT.PSB= 0004            MT.PS0= 0001
MT.PS1= 0002            MT.REV= 0020            MT.WRT= 0010            MT.Z  = 0008
M.ATA = 0080            M.CAPE= 0020            M.CONT= 0080            M.DEM = 0020
M.EBL = 0010            M.EXC = 0008            M.FAIL= 0008            M.ILR = 0010
M.INIT= 0010            M.OCC = 0020            M.ONLI= 0001            M.PE  = 0040
M.PORT= 0080            M.RDEN= 0002            M.RDPE= 0008            M.RUN = 0004
M.SCLK= 0001            M.TRA = 0040            M.UNIT= 0007            M.WCLK= 0040
M.WCLN= 0080            M.WREN= 0080            M5.5  = 0001            M6.5  = 0002
M7.5  = 0004            NOTCAP= 0088            OKAY  = 00FF            OPRRAM= 4300
OPSTRT= 0058            OPVER = 0040            PADCNT= 00D5            PADCRC= 0080
PDIAG = 0048            PEID  = 008A            PENAB = 004C            PESET = 0001
PL    = 00B1            POINTE  4683            PPOST   4489            PPREAM  4489
PRDD  = 004C            PRECN1  4426  G         PRECN3  44EE  G         PRENF = 009C
PS    = 00B2            PSTAT = 0048            PSW   =%0009            PTSTOK  44AE  G
PTY     4491            PTYC    44D2            PTY1    449C            PTY2    44A9
PXPTBL  4664            P.AMTP= 0001            P.BCTC= 0040            P.CMDP= 0020
P.INIE= 0080            P.LCS = 0040            P.LWR = 0020            P.RDP = 0002
P.RPEN= 0004            P.RPST= 0002            P.RPOE= 0020            P.RP1E= 0010
P.RP2E= 0020            P.RP3E= 0010            P.SING= 0080            P.STAT= 0002
P.STPE= 0080            P.TACH= 0008            P.TUPR= 0010            P.WCSP= 0004
P.WDS = 0040            P.WFLP= 0001            P.WPEN= 0010            P.WPOE= 0008
P.WP1E= 0004            P.WP2E= 0008            P.WP3E= 0004            P.5VOK= 0002
QUE   = 281B            QUEM  = 281E            RAM   = 0010            RARA  = 0006
RARAI = 0004            RCHBDO= 0048            RCHBD1= 0047            RCHOK = 0046
RCHTST= 000C            RCLRT = 000D            RCMD  = 000B            RCMLP = 0003
RCONT = 0080            RDATA = 0017            RDCLK = 0010            RDON  = 0011
READG = 0007            RECCN1  4524  G         RECCN2  454D  G         RECCN3  453F  G
RECEND  4508            REND  = 0014            REQST = 2806            RESCHR= 00D1
REVTST= 0064            REWIND= 0004            RFIFOL= 0008            RGCLK = 0002
RGCRI = 0003            RIBG  = 0001            RILL  = 0012            RINST = 000C
RMCTST= 0008            RMK2  = 0013            RNOP  = 0000            RPATH = 0001
RPBAD = 0044            RPCHI = 0001            RPCLK = 0003            RPCTL = 0009
RPEI  = 0002            RPFAIL= 0000            RPF1  = 009D            RPF2  = 009E
RPOK  = 0043            RPOSTN= 0016            RPSTA = 0015            RRCMT = 000A
RSTAT = 0002            RTIEB = 000A            RTIER = 0030            RTM   = 0005
RUNKI = 0009            RUPTST= 005E            RWDUNL= 0005            R.AMT = 0001
R.BOP = 0008            R.DATA= 0040            R.DON = 0002            R.DRDY= 0010
R.END = 0010            R.ILL = 0004            R.JVOK= 0004            R.MK2 = 0008
```

```
R.PLOD= 0008    R.PLO0= 0010    R.PLO1= 0020    R.POST= 0020
R.STNM= 0002    R.STOP= 0004    R.STPC= 0001    R.TBJN= 0080
R.TSTD= 004C    R.VOK = 0080    R00H  = 0081    R00L  = 0080
R01H  = 0083    R01L  = 0082    R02H  = 0085    R02L  = 0084
R03H  = 0087    R03L  = 0086    R04H  = 0089    R04L  = 0088
R05H  = 008B    R05L  = 008A    R06H  = 008D    R06L  = 008C
R07H  = 008F    R07L  = 008E    R10H  = 0091    R10L  = 0090
R11H  = 0093    R11L  = 0092    R12H  = 0095    R12L  = 0094
R13H  = 0097    R13L  = 0096    R14H  = 0099    R14L  = 0098
R15H  = 009B    R15L  = 009A    R16H  = 009D    R16L  = 009C
R17H  = 009F    R17L  = 009E    R7.5  = 0010    SELCLR= 0000
SEQERR  4548    SETATA= 00A1    SID   = 0080    SOD   = 0080
SOE   = 0040    SP    =%0008    SSCLK = 0040    SSTEP = 0005
STACK = 4FFF    STATRM= 4F20    STPCT   4F20    STRSP = 5000
TADR00= 0080    TADR01= 0081    TADR02= 0082    TADR03= 0083
TADR04= 0084    TADR05= 0085    TADR06= 0086    TADR07= 0087
TADR10= 0088    TADR11= 0089    TADR12= 008A    TADR13= 008B
TAMT  = 0044    TASEL = 0080    TCMD  = 0040    TC.FWD= 0040
TC.INH= 0008    TC.LWR= 0004    TC.REV= 0020    TC.WRT= 0010
TEMP    4F99    TEST1   4300    TMF   = 0099    TMRDY = 0040
TRANSC  4685    TRKENA= 00D2    TSET  = 2803    TSTEND= 2818
TSTS  = 0040    TUSEL0= 00D1    TUSEL1= 00D2    TU78  = 0010
T.ATH0= 0001    T.ATH1= 0002    T.BOT = 0004    T.EOT = 0002
T.FPT = 0001    T.NTHR= 0004    T.ONL = 0020    T.PES = 0008
T.PSBJ= 0020    T.PSOJ= 0008    T.PS1J= 0010    T.RDY = 0080
T.RDY0= 0040    T.RWD = 0010    T.SCLK= 0002    T1LOOP  4311
T1LPA   433F    UIBG  = 00A1    UNITMP  467E    VALFC   4F98
VALTB   4F95    VELTST= 005B    WDR.P = 0010    WMCCTL= 00D3
WMCERR= 00DA    WMCSTA= 00D0    WRTCLK= 0000    WRTDAT= 00D3
W.ACRC= 0004    W.CRC = 0008    W.DIAG= 0002    W.DONN= 0040
W.ECC = 0010    W.ENAB= 0080    W.ERR = 0020    W.FMT = 0070
W.GCR = 0010    W.LEFT= 0004    W.ONES= 0020    W.RESI= 0002
W.REV = 0004    W.ROME= 0010    W.RST = 0001    W.SKIP= 000F
W.WRIT= 0008    W.XFER= 0020    X     =%000A    XDATA   467F
XDLOOP  442F    XDLPC   444A    XDLPC1  4414    XDLPC2  4421
XDLP1   443C    XPLOOP  44B7    XPLP1   44C4    X.DONN= 0080
X.ENAB= 0040    X.PEPE= 0002    X.ROME= 0001    X.WCLK= 0001
Y     =%000B    .     = 4688

ERRORS DETECTED:  0

*XMC4.A78/PTP,XMC4=NLIST,PARAM,MACRO,LIST,XMC4
RUN-TIME: 4 6 0 SECONDS
CORE USED:  10K
```

3 "MEMORY MAP" EXTENDED ADDRESS BIT
19 READ MICRO CONTROLLER CONTROL & STATUS WORDS
139 TRANSPORT/PORT COMMAND + STATUS WORDS
380 COMMON ADDRESS SPACE REGISTERS
444 MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485 MASSBUS CONTROL WORD DEFINITIONS
557 WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709 KEYBOARD/DISPLAY CONTROL + STATUS WORD
797 MISCELLANEOUS VARIABLE DEFINITIONS
801 RAM ADDRESS DEFINITIONS
819 LINKAGE TABLE ADDRESSES
842 BIT DEFINITIONS
856 RIM AND SIM INSTRUCTION BIT DEFINITIONS
877 INTERRUPTABLE INSTRUCTION MACROS
924 DIAGNOSTIC VARIABLES
945 COMMON VARIABLE STORAGE
959 MACRO CALL DEFINITIONS
1047 TM78 MICRO-DIAGNOSTIC MACRO ALLS
1101 8080 MACRO CALLS - ERROR A' TEST INITIALIZATION
1340 TEST 1 - XMC - GCR 4X5 TR SLATION SUBGROUP 0000(2) TEST
1465 TEST 2 - XMC - GCR 4X5 TRANSLATION SUBGROUP 01 TEST
1589 TEST 3 - XMC - GCR 4X5 TRANSLATION SUBGROUP 02 TEST
1713 TEST 4 - XMC - GCR 4X5 TRANSLATION SUBGROUP 03 TEST
1837 TEST 5 - XMC - GCR 4X5 TRANSLATION SUBGROUP 04 TEST
1961 TEST 6 - XMC - GCR 4X5 TRANSLATION SUBGROUP 05 TEST
2085 TEST 7 - XMC - GCR 4X5 TRANSLATION SUBGROUP 06 TEST
2209 TEST 10 - XMC - GCR 4X5 TRANSLATION SUBGROUP 07 TEST
2333 TEST 11 - XMC - GCR 4X5 TRANSLATION SUBGROUP 10 TEST
2457 TEST 12 - XMC - GCR 4X5 TRANSLATION SUBGROUP 11 TEST
2581 TEST 13 - XMC - GCR 4X5 TRANSLATION SUBGROUP 12 TEST
2705 TEST 14 - XMC - GCR 4X5 TRANSLATION SUBGROUP 13 TEST
2829 TEST 15 - XMC - GCR 4X5 TRANSLATION SUBGROUP 14 TEST
2953 TEST 16 - XMC - GCR 4X5 TRANSLATION SUBGROUP 15 TEST
3077 TEST 17 - XMC - GCR 4X5 TRANSLATION SUBGROUP 16 TEST
3201 TEST 20 - XMC - GCR 4X5 TRANSLATION SUBGROUP 17 TEST
3324 FINDIT - SUBROUTINE TO GET A TEST PORT FOR TESTING USE
3461 XFRSUB - SUBROUTINE TO CLOCK DATA THRU TRANSLATOR & CHECK IT
3958 SUBROUTINE CLEAR ALL TU PORTS

```
                ;*************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;               "BYTE/SCLK COUNT NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               MM = DATA FORMAT FROM CAS REGISTER 2
                ;               NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;               BYTE-SCLK COUNT = LLL
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               LLL = AS ABOVE
                ;               MM  = AS ABOVE
                ;               NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;               TRANSITION COUNT = LLL
                ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;               EXPECTED 18 BITS =      E EEEEEE
                ;               ACTUAL 18 BITS =        A AAAAAA
                ;
                ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;               OF CAS REG 15 LOW BYTE.
                ;
                ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;               17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;               AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;               "SUBGROUP NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;*************************************************************************
```

```
1016                    ; ************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ; ************************************************************************
1035                    ; ************************************************************************
1036                    ;DATA PATTERN CODES
1037
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ; ************************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  XMC5 - WRITE PATH 4 X 5 TRANSLATIONS
                                .MACRO  SETLPC  ARG1,ARG2
                                        LXI     H,ARG1          ;GET THE LOOP-ON-ERR ADDRESS
                                        SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
                                        LXI     H,ARG2          ;GET THE CONTINUE ADDRESS
                                        SHLD    ERCADR+1        ;SAVE IT
                                        XRA     A
                                .ENDM
                                .SBTTL  TEST 1 - XMC - GCR 4X5 TRANSLATION SUBGROUP 0000(2) TEST
                                ;ID     XMC5-TRANSLATOR MICRO CONTROLLER PART #5
1342  4300                      ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1343                            ;*XMC - GCR 4 X 5 TEST
1344  4300                      SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1345                            ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
1346                            ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
1347                            ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
1348                            ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 0000(2) TO TEST TRANSLATION ROM
1349                            ;*PATTERN 11001(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
1350                            ;*
1351                            ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
1352                            ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
1353                            ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
1354                            ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
1355  4300                      SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1356                            ;*BGNTST
1357                            ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
1358                            ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
1359                            ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
1360                            ;*      :       ELSE - CONTINUE
1361                            ;*      ENDIF
1362                            ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
1363                            ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
1364                            ;*      BGNDO
1365                            ;*      :       INIT THE INTERNAL COUNTER REGISTERS
1366                            ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
1367                            ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
1368                            ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
1369                            ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
1370                            ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
1371                            ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
1372                            ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
1373                            ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
1374                            ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
1375                            ;*      :       BGNDO
1376                            ;*      :       :       CLOCK THE SYSTEM
```

```
1377                    ;*      :       :       DECREMENT THE LOOP COUNT
1378                    ;*      :       :       DO UNTIL LOOP COUNTER = 0
1379                    ;*      :       ENDDO
1380                    ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
1381                    ;*      :       BGNDO
1382                    ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
1383                    ;*      :       :       BGNDO
1384                    ;*      :       :       :       CLOCK THE SYSTEM
1385                    ;*      :       :       :       DECREMENT THE LOOP COUNT
1386                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
1387                    ;*      :       :       ENDDO
1388                    ;*      :       :       IF TIMEOUT COUNT = 0
1389                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
1390                    ;*      :       :       :       ELSE - CONTINUE
1391                    ;*      :       :       ENDIF
1392                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
1393                    ;*      :       ENDDO
1394                    ;*      :       BGNDO
1395                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
1396                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
1397                    ;*      :       :       :       THEN - CONTINUE
1398                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
1399                    ;*      :       :       ENDIF
1400                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
1401                    ;*      :       :       BGNDO
1402                    ;*      :       :       :       CLOCK THE SYSTEM
1403                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
1404                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
1405                    ;*      :       :       ENDDO
1406                    ;*      :       :       IF 'WCLK' IS SET
1407                    ;*      :       :       :       THEN - CONTINUE
1408                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
1409                    ;*      :       :       ENDIF
1410                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
1411                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
1412                    ;*      :       ENDDO
1413                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
1414                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
1415                    ;*      ENDDO
1416                    ;*ENDTST
1417  4300              SE
 (1)                    ; ***************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
1418                    ;*XMC5 MICRO TEST 01
1419                    ;*XMC5 MICRO ERROR 01
1420                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0000(2)
1421                    ;*M8958, M8959, M8955'S
1422                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
1423                    ;*ACTUAL = NNNN
1424                    ;*EXPECTED = NNNN
1425                    ;*TRANSITION COUNT = LLL
1426                    ;*
1427                    ;*XMC5 MICRO TEST 01
```

```
                                        ;*XMC5 MICRO ERROR 21
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0000(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
                                        ;*FATAL ERROR - TEST ABORTED
                                        ;*
                                        ;*XMC5 MICRO TEST 01
                                        ;*XMC5 MICRO ERROR 22
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0000(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 61
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0000(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
 1446  4300                             S
  (1)                                   ; ************************************************************************
 1447  4300                             TEST1:  TESTX   1
  (1)  4300  3E  01            7.0                 MVI     A,1             ;DEFINE THE TEST NUMBER
  (1)  4302  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
 1448                                   ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 0000(2)
 1449                                   ;&M8958, M8959, M8955'S
 1450
 1451  4305  AF                   4.0  T1RPT:  XRA     A                  ;SET FOR SUBGROUP 00 TEST
 1452  4306  32  D6  48           13.0         STA     XFRNUM             ;SAVE THE XFR NUMBER
 1453  4309  CD  52  47           18.0         CALL    FINDIT             ;GET THE PORT NUMBER
 1454  430C                               SETLPC  T1LOOP,T1CONT
  (1)  430C  21  19  43           10.0                 LXI     H,T1LOOP                ;GET THE LOOP-ON-ERR ADDRESS
  (1)  430F  22  E4  48           16.0                 SHLD    LPLADR+1        ;SAVE FOR ERROR LOOPING
  (1)  4312  21  2A  43           10.0                 LXI     H,T1CONT                ;GET THE CONTINUE ADDRESS
  (1)  4315  22  E7  48           16.0                 SHLD    ERCADR+1        ;SAVE IT
  (1)  4318  AF                    4.0                 XRA     A
 1455
 1456  4319  CD  2F  48           18.0  T1LOOP: CALL    XFRSUB             ;CLOCK DATA THRU TRANSLATOR & CHECK IT
 1457  431C  CA  24  43           10.0         JZ      T1DEO              ;JUMP IF NO ERROR DETECTED
 1458  431F                               ERRB    XDLOOP,T1DEO,4
  (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  431F  CD  12  28           18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0001                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4322  01                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4323  04                                       .BYTE   4               ;PRINT ROUTINE NUMBER
  (1)  4324  CD  15  28           18.0         T1DEO:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  4327  DA  89  48           10.0                 JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
 1459                                   ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC
 1460
 1461  432A  CD  A4  48           18.0  T1CONT: CALL    NXTBYT             ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
 1462  432D  C2  19  43           10.0         JNZ     T1LOOP             ;LOOP TILL 5 BYTES CHECKED
 1463  4330                               ENDTST  T1RPT
  (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  4330                             REQ     7                  ;FAKE CALL TO KEEP TEST ALIVE
  (2)  4330  CD  06  28           18.0                 CALL    REQST
```

```
(2)  4333  00                                          .BYTE                    ;DATA PATTERN NUMBER
(2)  4334  00  00                                      .WORD                    ;SYSTEM '"' COUNT
(2)  4336  00  00                                      .WORD                    ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
(2)  4338  00                                          .BYTE                    ;DATA COMPARE FLAG IF =1
(2)  4339  07                                          .BYTE   7                ;REQUEST CODE
(1)  433A  3A  9A  4F      13.0        LDA     ITERA           ;GET ITERATION COUNT
(1)  433D  3D                4.0       DCR     A               ;DOWNCOUNT
(1)  433E  32  9A  4F      13.0        STA     ITERA           ;SAVE COUNT
(1)  4341  F2  05  43      10.0        JP      T1RPT           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 2 - XMC - GCR 4X5 TRANSLATION SUBGROUP 01 TEST
1466  4344                      ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
1468  4344                      SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 0001(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 11011(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
1479  4344                      SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
1510                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
1511                    ;*      :       :       ENDDO
1512                    ;*      :       :       IF TIMEOUT COUNT = 0
1513                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
1514                    ;*      :       :       :       ELSE - CONTINUE
1515                    ;*      :       :       ENDIF
1516                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
1517                    ;*      :       ENDDO
1518                    ;*      :       BGNDO
1519                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
1520                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
1521                    ;*      :       :       :       THEN - CONTINUE
1522                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
1523                    ;*      :       :       ENDIF
1524                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
1525                    ;*      :       :       BGNDO
1526                    ;*      :       :       :       CLOCK THE SYSTEM
1527                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
1528                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
1529                    ;*      :       :       ENDDO
1530                    ;*      :       :       IF 'WCLK' IS SET
1531                    ;*      :       :       :       THEN - CONTINUE
1532                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
1533                    ;*      :       :       ENDIF
1534                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
1535                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
1536                    ;*      :       ENDDO
1537                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
1538                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
1539                    ;*      ENDDO
1540                    ;*ENDTST
1541  4344              SE
 (1)                    ; ***************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
1542                    ;*XMC5 MICRO TEST 02
1543                    ;*XMC5 MICRO ERROR 02
1544                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0001(2)
1545                    ;*M8958, M8959, M8955'S
1546                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
1547                    ;*ACTUAL = NNNN
1548                    ;*EXPECTED = NNNN
1549                    ;*TRANSITION COUNT = LLL
1550                    ;*
1551                    ;*XMC5 MICRO TEST 02
1552                    ;*XMC5 MICRO ERROR 21
1553                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0001(2)
1554                    ;*M8958, M8959, M8955'S
1555                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
1556                    ;*FATAL ERROR - TEST ABORTED
1557                    ;*
1558                    ;*XMC5 MICRO TEST 02
1559                    ;*XMC5 MICRO ERROR 22
1560                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0001(2)
```

```
1561                                    ;*M8958, M8959, M8955'S
1562                                    ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
1563                                    ;*FATAL ERROR - TEST ABORTED.
1564                                    ;*
1565                                    ;*XMC5 MICRO TEST 02
1566                                    ;*XMC5 MICRO ERROR 23
1567                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0001(2)
1568                                    ;*M8958, M8959, M8955'S
1569                                    ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
1570  4344                              S
 (1)                                    ; ***************************************************************************
1571  4344                              TEST2:  TESTX   2
 (1)  4344  3E  02          7.0                         MVI     A,2             ;DEFINE THE TEST NUMBER
 (1)  4346  CD  03  28     18.0                         CALL    TSET            ;SETUP THE TEST
1572                                    ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 0001(2)
1573                                    ;&M8958, M8959, M8955'S
1574
1575  4349  3E  01          7.0  T2RPT:  MVI     A,1             ;SET FOR SUBGROUP 01 TEST
1576  434B  32  D6  48       13.0        STA     XFRNUM          ;SAVE THE XFR NUMBER
1577  434E  CD  52  47       18.0        CALL    FINDIT          ;GET THE PORT NUMBER
1578  4351                              SETLPC  T2LOOP,T2CONT
 (1)  4351  21  5E  43       10.0                LXI     H,T2LOOP                ;GET THE LOOP-ON-ERR ADDRESS
 (1)  4354  22  E4  48       16.0                SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  4357  21  6F  43       10.0                LXI     H,T2CONT                ;GET THE CONTINUE ADDRESS
 (1)  435A  22  E7  48       16.0                SHLD    ERCADR+1        ;SAVE IT
 (1)  435D  AF                4.0                XRA     A
1579
1580  435E  CD  2F  48       18.0  T2LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
1581  4361  CA  69  43       10.0        JZ      T2DEO           ;JUMP IF NO ERROR DETECTED
1582  4364                              ERRB    XDLOOP,T2DE0,4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4364  CD  12  28       18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0002                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4367  02                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4368  04                       .BYTE   4               ;PRINT ROUTINE NUMBER
 (1)  4369  CD  15  28       18.0    T2DE0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  436C  DA  89  48       10.0                JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1583                                    ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC
1584
1585  436F  CD  A4  48       18.0  T2CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
1586  4372  C2  5E  43       10.0        JNZ     T2LOOP          ;LOOP TILL 5 BYTES CHECKED
1587  4375                              ENDTST  T2RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4375                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4375  CD  06  28       18.0                CALL    REQST
 (2)  4378  00                       .BYTE                   ;DATA PATTERN NUMBER
 (2)  4379  00  00                   .WORD                   ;SYSTEM "" COUNT
 (2)  437B  00  00                   .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  437D  00                       .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  437E  07                       .BYTE   7               ;REQUEST CODE
 (1)  437F  3A  9A  4F       13.0        LDA     ITERA           ;GET ITERATION COUNT
 (1)  4382  3D                4.0        DCR     A               ;DOWNCOUNT
 (1)  4383  32  9A  4F       13.0        STA     ITERA           ;SAVE COUNT
 (1)  4386  F2  49  43       10.0        JP      T2RPT           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 3 - XMC - GCR 4X5 TRANSLATION SUBGROUP 02 TEST
4389                            ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
4389                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 0010(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 10010(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
4389                            SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPFLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
1634                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
1635                    ;*      :       :       ENDDO
1636                    ;*      :       :       IF TIMEOUT COUNT = 0
1637                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
1638                    ;*      :       :       :       ELSE - CONTINUE
1639                    ;*      :       :       ENDIF
1640                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
1641                    ;*      :       ENDDO
1642                    ;*      :       BGNDO
1643                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
1644                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
1645                    ;*      :       :       :       THEN - CONTINUE
1646                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
1647                    ;*      :       :       ENDIF
1648                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
1649                    ;*      :       :       BGNDO
1650                    ;*      :       :       :       CLOCK THE SYSTEM
1651                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
1652                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
1653                    ;*      :       :       ENDDO
1654                    ;*      :       :       IF 'WCLK' IS SET
1655                    ;*      :       :       :       THEN - CONTINUE
1656                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
1657                    ;*      :       :       ENDIF
1658                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
1659                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
1660                    ;*      :       ENDDO
1661                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
1662                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
1663                    ;*      ENDDO
1664                    ;*ENDTST
1665  4389              SE
 (1)                    ; ***************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
1666                    ;*XMC5 MICRO TEST 03
1667                    ;*XMC5 MICRO ERROR 03
1668                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0010(2)
1669                    ;*M8958, M8959, M8955'S
1670                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
1671                    ;*ACTUAL = NNNN
1672                    ;*EXPECTED = NNNN
1673                    ;*TRANSITION COUNT = LLL
1674                    ;*
1675                    ;*XMC5 MICRO TEST 03
1676                    ;*XMC5 MICRO ERROR 21
1677                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0010(2)
1678                    ;*M8958, M8959, M8955'S
1679                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
1680                    ;*FATAL ERROR - TEST ABORTED
1681                    ;*
1682                    ;*XMC5 MICRO TEST 03
1683                    ;*XMC5 MICRO ERROR 22
1684                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0010(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED.
                                        ;*
                                        ;*XMC5 MICRO TEST 03
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0010(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
1694  4389                              S
 (1)                                    ; ************************************************************************************
1695  4389                              TEST3:  TESTX   3
 (1)  4389  3E  03            7.0                     MVI     A,3             ;DEFINE THE TEST NUMBER
 (1)  438B  CD  03  28       18.0                     CALL    TSET            ;SETUP THE TEST
1696                                    ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 0010(2)
1697                                    ;&M8958, M8959, M8955'S
1698
1699  438E  3E  02             7.0  T3RPT:  MVI     A,2             ;SET FOR SUBGROUP 02 TEST
1700  4390  32  D6  48          13.0        STA     XFRNUM          ;SAVE THE XFR NUMBER
1701  4393  CD  52  47          18.0        CALL    FINDIT          ;GET THE PORT NUMBER
1702  4396                              SETLPC  T3LOOP,T3CONT
 (1)  4396  21  A3  43          10.0                LXI     H,T3LOOP                ;GET THE LOOP-ON-ERR ADDRESS
 (1)  4399  22  E4  48          16.0                SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  439C  21  B4  43          10.0                LXI     H,T3CONT                ;GET THE CONTINUE ADDRESS
 (1)  439F  22  E7  48          16.0                SHLD    ERCADR+1        ;SAVE IT
 (1)  43A2  AF                   4.0                XRA     A
1703
1704  43A3  CD  2F  48          18.0  T3LOOP: CALL     XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
1705  43A6  CA  AE  43          10.0          JZ      T3DE0           ;JUMP IF NO ERROR DETECTED
1706  43A9                              ERRB    XDLOOP,T3DE0,4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  43A9  CD  12  28          18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0003                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43AC  03                        .BYTF   MSGN            ;MESSAGE NUMBER ID
 (1)  43AD  04                        .BYTE   4               ;PRINT ROUTINE NUMBER
 (1)  43AE  CD  15  28          18.0  T3DE0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43B1  DA  89  48          10.0          JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1707                                    ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC
1708
1709  43B4  CD  A4  48          18.0  T3CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
1710  43B7  C2  A3  43          10.0          JNZ     T3LOOP          ;LOOP TILL 5 BYTES CHECKED
1711  43BA                              ENDTST  T3RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43BA                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43BA  CD  06  28          18.0                CALL    REQST
 (2)  43BD  00                        .BYTE                   ;DATA PATTERN NUMBER
 (2)  43BE  00  00                    .WORD                   ;SYSTEM "*" COUNT
 (2)  43C0  00  00                    .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43C2  00                        .BYTE                   ;DATA COMPARE FLAG IF  1
 (2)  43C3  07                        .BYTE   7               ;REQUEST CODE
 (1)  43C4  3A  9A  4F          13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  43C7  3D                   4.0          DCR     A               ;DOWNCOUNT
 (1)  43C8  32  9A  4F          13.0          STA     ITERA           ;SAVE COUNT
 (1)  43CB  F2  8E  43          10.0          JP      T3RPT           ;DO TEST UNTIL TILL - 0
```

```
                                .SBTTL  TEST 4 - XMC - GCR 4X5 TRANSLATION SUBGROUP 03 TEST
43CE                            ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
43CE                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 0011(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 10011(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
43CE                            SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
1758                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
1759                    ;*      :       :       ENDDO
1760                    ;*      :       :       IF TIMEOUT COUNT = 0
1761                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
1762                    ;*      :       :       :       ELSE - CONTINUE
1763                    ;*      :       :       ENDIF
1764                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
1765                    ;*      :       ENDDO
1766                    ;*      :       BGNDO
1767                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
1768                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
1769                    ;*      :       :       :       THEN - CONTINUE
1770                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
1771                    ;*      :       :       ENDIF
1772                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
1773                    ;*      :       :       BGNDO
1774                    ;*      :       :       :       CLOCK THE SYSTEM
1775                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
1776                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
1777                    ;*      :       :       ENDDO
1778                    ;*      :       :       IF 'WCLK' IS SET
1779                    ;*      :       :       :       THEN - CONTINUE
1780                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
1781                    ;*      :       :       ENDIF
1782                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
1783                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
1784                    ;*      :       ENDDO
1785                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
1786                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
1787                    ;*      ENDDO
1788                    ;*ENDTST
1789  43CE              SE
 (1)                    ; ***************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
1790                    ;*XMC5 MICRO TEST 04
1791                    ;*XMC5 MICRO ERROR 04
1792                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0011(2)
1793                    ;*M8958, M8959, M8955'S
1794                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
1795                    ;*ACTUAL = NNNN
1796                    ;*EXPECTED = NNNN
1797                    ;*TRANSITION COUNT = LLL
1798                    ;*
1799                    ;*XMC5 MICRO TEST 04
1800                    ;*XMC5 MICRO ERROR 21
1801                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0011(2)
1802                    ;*M8958, M8959, M8955'S
1803                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
1804                    ;*FATAL ERROR - TEST ABORTED
1805                    ;*
1806                    ;*XMC5 MICRO TEST 04
1807                    ;*XMC5 MICRO ERROR 22
1808                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0011(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED.
                                        ;*
                                        ;*XMC5 MICRO TEST 04
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0011(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
43CE                                    S
 (1)                                    ; ************************************************************************
43CE                                    TEST4:  TESTX   4
 (1)  43CE  3E  04            7.0                       MVI     A,4             ;DEFINE THE TEST NUMBER
 (1)  43D0  CD  03  28       18.0                       CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 0011(2)
                                        ;&M8958, M8959, M8955'S

43D3  3E  03                 7.0  T4RPT:  MVI     A,3             ;SET FOR SUBGROUP 03 TEST
43D5  32  D6  48            13.0          STA     XFRNUM          ;SAVE THE XFR NUMBER
43D8  CD  52  47            18.0          CALL    FINDIT          ;GET THE PORT NUMBER
43DB                                      SETLPC  T4LOOP,T4CONT
 (1)  43DB  21  E8  43       10.0                  LXI     H,T4LOOP                ;GET THE LOOP-ON-ERR ADDRESS
 (1)  43DE  22  E4  48       16.0                  SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  43E1  21  F9  43       10.0                  LXI     H,T4CONT                ;GET THE CONTINUE ADDRESS
 (1)  43E4  22  E7  48       16.0                  SHLD    ERCADR+1        ;SAVE IT
 (1)  43E7  AF                4.0                  XRA     A

43E8  CD  2F  48            18.0  T4LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
43EB  CA  F3  43            10.0          JZ      T4DEO           ;JUMP IF NO ERROR DETECTED
43EE                                      ERRB    XDLOOP,T4DEO,4
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  43EE  CD  12  28       18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0004                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43F1  04                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43F2  04                                     .BYTE   4               ;PRINT ROUTINE NUMBER
 (1)  43F3  CD  15  28       18.0          T4DEO:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43F6  DA  89  48       10.0                  JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC

43F9  CD  A4  48            18.0  T4CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
43FC  C2  E8  43            10.0          JNZ     T4LOOP          ;LOOP TILL 5 BYTES CHECKED
43FF                                      ENDTST  T4RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43FF                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43FF  CD  06  28       18.0                  CALL    REQST
 (2)  4402  00                                     .BYTE                   ;DATA PATTERN NUMBER
 (2)  4403  00  00                                 .WORD                   ;SYSTEM "" COUNT
 (2)  4405  00  00                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4407  00                                     .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4408  07                                     .BYTE   7               ;REQUEST CODE
 (1)  4409  3A  9A  4F       13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  440C  3D                4.0          DCR     A               ;DOWNCOUNT
 (1)  440D  32  9A  4F       13.0          STA     ITERA           ;SAVE COUNT
 (1)  4410  F2  D3  43       10.0          JP      T4RPT           ;DO TEST UNTIL TILI = 0
```

```
                                .SBTTL  TEST 5 - XMC - GCR 4X5 TRANSLATION SUBGROUP 04 TEST
4413                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
4413                            SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 0100(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 11101(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
4413                            SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TFST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
1882                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
1883                    ;*      :       :       ENDDO
1884                    ;*      :       :       IF TIMEOUT COUNT = 0
1885                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
1886                    ;*      :       :       :       ELSE - CONTINUE
1887                    ;*      :       :       ENDIF
1888                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
1889                    ;*      :       ENDDO
1890                    ;*      :       BGNDO
1891                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
1892                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
1893                    ;*      :       :       :       THEN - CONTINUE
1894                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
1895                    ;*      :       :       ENDIF
1896                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
1897                    ;*      :       :       BGNDO
1898                    ;*      :       :       :       CLOCK THE SYSTEM
1899                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
1900                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
1901                    ;*      :       :       ENDDO
1902                    ;*      :       :       IF 'WCLK' IS SET
1903                    ;*      :       :       :       THEN - CONTINUE
1904                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
1905                    ;*      :       :       ENDIF
1906                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
1907                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
1908                    ;*      :       ENDDO
1909                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
1910                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
1911                    ;*      ENDDO
1912                    ;*ENDTST
1913  4413              SE
 (1)                    ;*************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
1914                    ;*XMC5 MICRO TEST 05
1915                    ;*XMC5 MICRO ERROR 05
1916                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0100(2)
1917                    ;*M8958, M8959, M8955'S
1918                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
1919                    ;*ACTUAL = NNNN
1920                    ;*EXPECTED = NNNN
1921                    ;*TRANSITION COUNT = LLL
1922                    ;*
1923                    ;*XMC5 MICRO TEST 05
1924                    ;*XMC5 MICRO ERROR 21
1925                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0100(2)
1926                    ;*M8958, M8959, M8955'S
1927                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
1928                    ;*FATAL ERROR - TEST ABORTED
1929                    ;*
1930                    ;*XMC5 MICRO TEST 05
1931                    ;*XMC5 MICRO ERROR 22
1932                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0100(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 05
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0100(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
4413                                    S
 (1)                                    ; ***************************************************************************
4413                                    TEST5:  TESTX   5
 (1)  4413  3E  05          7.0                          MVI     A,5             ;DEFINE THE TEST NUMBER
 (1)  4415  CD  03  28     18.0                          CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 0100(2)
                                        ;&M8958, M8959, M8955'S

4418  3E  04                7.0  T5RPT:  MVI     A,4             ;SET FOR SUBGROUP 04 TEST
441A  32  D6  48           13.0           STA     XFRNUM          ;SAVE THE XFR NUMBER
441D  CD  52  47           18.0           CALL    FINDIT          ;GET THE PORT NUMBER
4420                                      SETLPC  T5LOOP,T5CONT
 (1)  4420  21  2D  44     10.0                   LXI     H,T5LOOP                ;GET THE LOOP-ON-ERR ADDRESS
 (1)  4423  22  E4  48     16.0                   SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  4426  21  3E  44     10.0                   LXI     H,T5CONT                ;GET THE CONTINUE ADDRESS
 (1)  4429  22  E7  48     16.0                   SHLD    ERCADR+1        ;SAVE IT
 (1)  442C  AF              4.0                   XRA     A

442D  CD  2F  48           18.0  T5LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
4430  CA  38  44           10.0          JZ      T5DE0           ;JUMP IF NO ERROR DETECTED
4433                                     ERRB    XDLOOP,T5DE0,4
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4433  CD  12  28     18.0                   CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0005                                 MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4436  05                                   .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4437  04                                   .BYTE   4               ;PRINT ROUTINE NUMBER
 (1)  4438  CD  15  28     18.0           T5DE0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  443B  DA  89  48     10.0                   JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC

443E  CD  A4  48           18.0  T5CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
4441  C2  2D  44           10.0          JNZ     T5LOOP          ;LOOP TILL 5 BYTES CHECKED
4444                                     ENDTST  T5RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4444                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4444  CD  06  28     18.0                   CALL    REQST
 (2)  4447  00                                    .BYTE                   ;DATA PATTERN NUMBER
 (2)  4448  00  00                                .WORD                   ;SYSTEM "" COUNT
 (2)  444A  00  00                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  444C  00                                    .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  444D  07                                    .BYTE   7               ;REQUEST CODE
 (1)  444E  3A  9A  4F     13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1)  4451  3D              4.0           DCR     A               ;DOWNCOUNT
 (1)  4452  32  9A  4F     13.0           STA     ITERA           ;SAVE COUNT
 (1)  4455  F2  18  44     10.0           JP      T5RPT           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 6 - XMC - GCR 4X5 TRANSLATION SUBGROUP 05 TEST
4458                            ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
4458                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 0101(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 10101(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
4458                            SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
2006                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
2007                    ;*      :       :       ENDDO
2008                    ;*      :       :       IF TIMEOUT COUNT = 0
2009                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
2010                    ;*      :       :       :       ELSE - CONTINUE
2011                    ;*      :       :       ENDIF
2012                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
2013                    ;*      :       ENDDO
2014                    ;*      :       BGNDO
2015                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
2016                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
2017                    ;*      :       :       :       THEN - CONTINUE
2018                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
2019                    ;*      :       :       ENDIF
2020                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
2021                    ;*      :       :       BGNDO
2022                    ;*      :       :       :       CLOCK THE SYSTEM
2023                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
2024                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
2025                    ;*      :       :       ENDDO
2026                    ;*      :       :       IF 'WCLK' IS SET
2027                    ;*      :       :       :       THEN - CONTINUE
2028                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
2029                    ;*      :       :       ENDIF
2030                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
2031                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
2032                    ;*      :       ENDDO
2033                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
2034                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
2035                    ;*      ENDDO
2036                    ;*ENDTST
2037  4458              SE
 (1)                    ; ***************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
2038                    ;*XMC5 MICRO TEST 06
2039                    ;*XMC5 MICRO ERROR 06
2040                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0101(2)
2041                    ;*M8958, M8959, M8955'S
2042                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
2043                    ;*ACTUAL = NNNN
2044                    ;*EXPECTED = NNNN
2045                    ;*TRANSITION COUNT = LLL
2046                    ;*
2047                    ;*XMC5 MICRO TEST 06
2048                    ;*XMC5 MICRO ERROR 21
2049                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0101(2)
2050                    ;*M8958, M8959, M8955'S
2051                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
2052                    ;*FATAL ERROR - TEST ABORTED
2053                    ;*
2054                    ;*XMC5 MICRO TEST 06
2055                    ;*XMC5 MICRO ERROR 22
2056                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0101(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 06
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0101(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
4458                                    S
 (1)                                    ; ***************************************************************************
4458                                    TEST6:  TESTX   6
 (1) 4458  3E  06          7.0                           MVI     A,6             ;DEFINE THE TEST NUMBER
 (1) 445A  CD  03  28     18.0                           CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 0101(2)
                                        ;&M8958, M8959, M8955'S

445D  3E  05              7.0   T6RPT:  MVI     A,5             ;SET FOR SUBGROUP 05 TEST
445F  32  D6  48         13.0           STA     XFRNUM          ;SAVE THE XFR NUMBER
4462  CD  52  47         18.0           CALL    FINDIT          ;GET THE PORT NUMBER
4465                                    SETLPC  T6LOOP,T6CONT
 (1) 4465  21  72  44     10.0                   LXI     H,T6LOOP                ;GET THE LOOP-ON-ERR ADDRESS
 (1) 4468  22  E4  48     16.0                   SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1) 446B  21  83  44     10.0                   LXI     H,T6CONT                ;GET THE CONTINUE ADDRESS
 (1) 446E  22  E7  48     16.0                   SHLD    ERCADR+1        ;SAVE IT
 (1) 4471  AF              4.0                   XRA     A
4472  CD  2F  48         18.0   T6LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
4475  CA  7D  44         10.0           JZ      T6DE0           ;JUMP IF NO ERROR DETECTED
4478                                    ERRB    XDLOOP,T6DE0,4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1) 4478  CD  12  28     18.0                   CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0006                                 MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 447B  06                                   .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 447C  04                                   .BYTE   4               ;PRINT ROUTINE NUMBER
 (1) 447D  CD  15  28     18.0           T6DE0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1) 4480  DA  89  48     10.0                   JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC

4483  CD  A4  48         18.0   T6CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
4486  C2  72  44         10.0           JNZ     T6LOOP          ;LOOP TILL 5 BYTES CHECKED
4489                                    END'ST  T6RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2) 4489                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2) 4489  CD  06  28     18.0                   CALL    REQST
 (2) 448C  09                                   .BYTE                   ;DATA PATTERN NUMBER
 (2) 448D  00  00                               .WORD                   ;SYSTEM "" COUNT
 (2) 448F  00  00                               .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 4491  00                                   .BYTE                   ;DATA COMPARE FLAG IF =1
 (2) 4492  07                                   .BYTE   7               ;REQUEST CODE
 (1) 4493  3A  9A  4F     13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1) 4496  3D              4.0           DCR     A               ;DOWNCOUNT
 (1) 4497  32  9A  4F     13.0           STA     ITERA           ;SAVE COUNT
 (1) 449A  F2  5D  44     10.0           JP      T6RPT           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 7 - XMC - GCR 4X5 TRANSLATION SUBGROUP 06 TEST
449D                            ST
 (1)                            ; ******************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
449D                            SD
 (1)                            ; ******************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 0110(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 10110(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
449D                            SP
 (1)                            ; ******************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
2130                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
2131                    ;*      :       :       ENDDO
2132                    ;*      :       :       IF TIMEOUT COUNT = 0
2133                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
2134                    ;*      :       :       :       ELSE - CONTINUE
2135                    ;*      :       :       ENDIF
2136                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
2137                    ;*      :       ENDDO
2138                    ;*      :       BGNDO
2139                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
2140                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
2141                    ;*      :       :       :       THEN - CONTINUE
2142                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
2143                    ;*      :       :       ENDIF
2144                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
2145                    ;*      :       :       BGNDO
2146                    ;*      :       :       :       CLOCK THE SYSTEM
2147                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
2148                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
2149                    ;*      :       :       ENDDO
2150                    ;*      :       :       IF 'WCLK' IS SET
2151                    ;*      :       :       :       THEN - CONTINUE
2152                    ;*      :       :       :       ELSE - REPORT IMPROPFR TRANSITIÜN RATE
2153                    ;*      :       :       ENDIF
2154                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
2155                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
2156                    ;*      :       ENDDO
2157                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
2158                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
2159                    ;*      ENDDO
2160                    ;*ENDTSf
2161  449D              SE
 (1)                    ; ***************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
2162                    ;*XMC5 MICRO TEST 07
2163                    ;*XMC5 MICRO ERROR 07
2164                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0110(2)
2165                    ;*M8958, M8959, M8955'S
2166                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
2167                    ;*ACTUAL = NNNN
2168                    ;*EXPECTED = NNNN
2169                    ;*TRANSITION COUNT = LLL
2170                    ;*
2171                    ;*XMC5 MICRO TEST 07
2172                    ;*XMC5 MICRO ERROR 21
2173                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0110(2)
2174                    ;*M8958, M8959, M8955'S
2175                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
2176                    ;*FATAL ERROR - TEST ABORTED
2177                    ;*
2178                    ;*XMC5 MICRO TEST 07
2179                    ;*XMC5 MICRO ERROR 22
2180                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0110(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 07
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0110(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
449D                                    S
 (1)                                    ; ************************************************************************
449D                                    TEST7:  TESTX   7
 (1) 449D  3E 07           7.0                           MVI     A,7             ;DEFINE THE TEST NUMBER
 (1) 449F  CD 03 28       18.0                           CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 0110(2)
                                        ;&M8958, M8959, M8955'S

44A2  3E 06               7.0  T7RPT:  MVI     A,6             ;SET FOR SUBGROUP 06 TEST
44A4  32 D6 48           13.0          STA     XFRNUM          ;SAVE THE XFR NUMBER
44A7  CD 52 47           18.0          CALL    FINDIT          ;GET THE PORT NUMBER
44AA                                   SETLPC  T7LOOP,T7CONT
 (1) 44AA  21 B7 44       10.0                  LXI     H,T7LOOP                ;GET THE LOOP-ON-ERR ADDRESS
 (1) 44AD  22 E4 48       16.0                  SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1) 44B0  21 C8 44       10.0                  LXI     H,T7CONT                ;GET THE CONTINUE ADDRESS
 (1) 44B3  22 E7 48       16.0                  SHLD    ERCADR+1        ;SAVE IT
 (1) 44B6  AF              4.0                  XRA     A

44B7  CD 2F 48           18.0  T7LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
44BA  CA C2 44           10.0          JZ      T7DEO           ;JUMP IF NO ERROR DETECTED
44BD                                   ERRB    XDLOOP,T7DEO,4
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1) 44BD  CD 12 28       18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0007                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 44C0  07                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 44C1  04                                  .BYTE   4               ;PRINT ROUTINE NUMBER
 (1) 44C2  CD 15 28       18.0          T7DEO:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1) 44C5  DA 89 48       10.0                  JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC

44C8  CD A4 48           18.0  T7CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
44CB  C2 B7 44           10.0          JNZ     T7LOOP          ;LOOP TILL 5 BYTES CHECKED
44CE                                   ENDTST  T7RPT
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2) 44CE                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2) 44CE  CD 06 28       18.0                  CALL    REQST
 (2) 44D1  00                                  .BYTE                   ;DATA PATTERN NUMBER
 (2) 44D2  00 00                               .WORD                   ;SYSTEM "" COUNT
 (2) 44D4  00 00                               .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 44D6  00                                  .BYTE                   ;DATA COMPARE FLAG IF =1
 (2) 44D7  07                                  .BYTE   7               ;REQUEST CODE
 (1) 44D8  3A 9A 4F       13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1) 44DB  3D              4.0          DCR     A               ;DOWNCOUNT
 (1) 44DC  32 9A 4F       13.0          STA     ITERA           ;SAVE COUNT
 (1) 44DF  F2 A2 44       10.0          JP      T7RPT           ;DO TEST UNTIL TILL - 0
```

```
                                .SBTTL  TEST 10 - XMC - GCR 4X5 TRANSLATION SUBGROUP 07 TEST
44E2                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
44E2                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BF 4 BYTES OF 0111(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 10111(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
44E2                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMECUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
2254                                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
2255                                    ;*      :       :       ENDDO
2256                                    ;*      :       :       IF TIMEOUT COUNT = 0
2257                                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
2258                                    ;*      :       :       :       ELSE - CONTINUE
2259                                    ;*      :       :       ENDIF
2260                                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
2261                                    ;*      :       ENDDO
2262                                    ;*      :       BGNDO
2263                                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
2264                                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
2265                                    ;*      :       :       :       THEN - CONTINUE
2266                                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
2267                                    ;*      :       :       ENDIF
2268                                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
2269                                    ;*      :       :       BGNDO
2270                                    ;*      :       :       :       CLOCK THE SYSTEM
2271                                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
2272                                    ;*      :       :       :       DO UNTIL LOOP COUNTER - 0
2273                                    ;*      :       :       ENDDO
2274                                    ;*      :       :       IF 'WCLK' IS SET
2275                                    ;*      :       :       :       THEN - CONTINUE
2276                                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
2277                                    ;*      :       :       ENDIF
2278                                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
2279                                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
2280                                    ;*      :       ENDDO
2281                                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
2282                                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
2283                                    ;*      ENDDO
2284                                    ;*ENDTST
2285  44E2                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
2286                                    ;*XMC5 MICRO TEST 10
2287                                    ;*XMC5 MICRO ERROR 10
2288                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0111(2)
2289                                    ;*M8958, M8959, M8955'S
2290                                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
2291                                    ;*ACTUAL = NNNN
2292                                    ;*EXPECTED = NNNN
2293                                    ;*TRANSITION COUNT = LLL
2294                                    ;*
2295                                    ;*XMC5 MICRO TEST 10
2296                                    ;*XMC5 MICRO ERROR 21
2297                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0111(2)
2298                                    ;*M8958, M8959, M8955'S
2299                                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
2300                                    ;*FATAL ERROR - TEST ABORTED
2301                                    ;*
2302                                    ;*XMC5 MICRO TEST 10
2303                                    ;*XMC5 MICRO ERROR 22
2304                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0111(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 10
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 0111(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
44E2                                    S
 (1)                                    ; *****************************************************************************
44E2                                    TEST10: TESTX   @10
 (1) 44E2  3E  08          7.0                         MVI     A,@10           ;DEFINE THE TEST NUMBER
 (1) 44E4  CD  03  28     18.0                         CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 0111(2)
                                        ;&M8958, M8959, M8955'S

44E7  3E  07              7.0   T10RPT: MVI     A,7             ;SET FOR SUBGROUP 07 TEST
44E9  32  D6  48         13.0           STA     XFRNUM          ;SAVE THE XFR NUMBER
44EC  CD  52  47         18.0           CALL    FINDIT          ;GET THE PORT NUMBER
44EF                                    SETLPC  T10LOOP,T10CONT
 (1) 44EF  21  FC  44     10.0                   LXI     H,T10LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 (1) 44F2  22  E4  48     16.0                   SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1) 44F5  21  0D  45     10.0                   LXI     H,T10CONT               ;GET THE CONTINUE ADDRESS
 (1) 44F8  22  E7  48     16.0                   SHLD    ERCADR+1        ;SAVE IT
 (1) 44FB  AF              4.0                   XRA     A

44FC  CD  2F  48         18.0   T10LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
44FF  CA  07  45         10.0            JZ      T10DE0          ;JUMP IF NO ERROR DETECTED
4502                                    ERRB    XDLOOP,T10DE0,4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1) 4502  CD  12  28     18.0                   CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)     0008                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 4505  08                                   .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 4506  04                                   .BYTE   4               ;PRINT ROUTINE NUMBER
 (1) 4507  CD  15  28     18.0           T10DE0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1) 450A  DA  89  48     10.0                   JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC

450D  CD  A4  48         18.0   T10CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
4510  C2  FC  44         10.0            JNZ     T10LOOP         ;LOOP TILL 5 BYTES CHECKED
4513                                    ENDTST  T10RPT
 (1)                                    ;TEST ITERATION CONTROL   ONCE FOR QUICK VERIFY
 (2) 4513                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2) 4513  CD  06  28     18.0                   CALL    REQST
 (2) 4516  00                                   .BYTE                   ;DATA PATTERN NUMBER
 (2) 4517  00  00                               .WORD                   ;SYSTEM "" COUNT
 (2) 4519  00  00                               .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 451B  00                                   .BYTE                   ;DATA COMPARE FLAG IF  1
 (2) 451C  07                                   .BYTE   7               ;REQUEST CODE
 (1) 451D  3A  9A  4F     13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1) 4520  3D              4.0           DCR     A               ;DOWNCOUNT
 (1) 4521  32  9A  4F     13.0           STA     ITERA           ;SAVE COUNT
 (1) 4524  F2  E7  44     10.0           JP      T10RPT          ;DO TEST UNTIL TILL - 0
```

```
                                .SBTTL  TEST 11 - XMC - GCR 4X5 TRANSLATION SUBGROUP '0 TEST
2334  4527                      ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
2336  4527                      SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 1000(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 11010(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
2347  4527                      SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
2378                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
2379                    ;*      :       :       ENDDO
2380                    ;*      :       :       IF TIMEOUT COUNT = 0
2381                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
2382                    ;*      :       :       :       ELSE - CONTINUE
2383                    ;*      :       :       ENDIF
2384                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
2385                    ;*      :       ENDDO
2386                    ;*      :       BGNDO
2387                    ;*      .       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
2388                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
2389                    ;*      :       :       :       THEN - CONTINUE
2390                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
2391                    ;*      :       :       ENDIF
2392                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
2393                    ;*      .       :       BGNDO
2394                    ;*      :       :       :       CLOCK THE SYSTEM
2395                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
2396                    ;*      :       :       :       DO UNTIL LOOP COUNTER   0
2397                    ;*      :       :       ENDDO
2398                    ;*      :       :       IF 'WCLK' IS SET
2399                    ;*      :       :       :       THEN - CONTINUE
2400                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
2401                    ;*      :       :       ENDIF
2402                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
2403                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
2404                    ;*      :       ENDDO
2405                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
2406                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
2407                    ;*      ENDDO
2408                    ;*ENDTST
2409  4527              SE
 (1)                    ; ***************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
2410                    ;*XMC5 MICRO TEST 11
2411                    ;*XMC5 MICRO ERROR 11
2412                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1000(2)
2413                    ;*M8958, M8959, M8955'S
2414                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
2415                    ;*ACTUAL = NNNN
2416                    ;*EXPECTED = NNNN
2417                    ;*TRANSITION COUNT = LLL
2418                    ;*
2419                    ;*XMC5 MICRO TEST 11
2420                    ;*XMC5 MICRO ERROR 21
2421                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1000(2)
2422                    ;*M8958, M8959, M8955'S
2423                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
2424                    ;*FATAL ERROR - TEST ABORTED
2425                    ;*
2426                    ;*XMC5 MICRO TEST 11
2427                    ;*XMC5 MICRO ERROR 22
2428                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1000(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED.
                                        ;*
                                        ;*XMC5 MICRO TEST 11
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1000(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
4527                                    S
 (1)                                    ; ***************************************************************************
4527                                    TEST11: TESTX   @11
 (1)  4527  3E  09           7.0                 MVI     A,@11           ;DEFINE THE TEST NUMBER
 (1)  4529  CD  03  28      18.0                 CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 1000(2)
                                        ;&M8958, M8959, M8955'S

452C  3E  08                7.0  T11RPT: MVI     A,@10           ;SET FOR SUBGROUP 10 TEST
452E  32  D6  48           13.0          STA     XFRNUM          ;SAVE THE XFR NUMBER
4531  CD  52  47           18.0          CALL    FINDIT          ;GET THE PORT NUMBER
4534                                      SETLPC  T11LOOP,T11CONT
 (1)  4534  21  41  45      10.0                  LXI     H,T11LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 (1)  4537  22  E4  48      16.0                  SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  453A  21  52  45      10.0                  LXI     H,T11CONT               ;GET THE CONTINUE ADDRESS
 (1)  453D  22  E7  48      16.0                  SHLD    ERCADR+1        ;SAVE IT
 (1)  4540  AF               4.0                  XRA     A

4541  CD  2F  48           18.0  T11LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
4544  CA  4C  45           10.0           JZ      T11DE0          ;JUMP IF NO ERROR DETECTED
4547                                      ERRB    XDLOOP,T11DE0,4
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4547  CD  12  28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0009                          MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  454A  09                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  454B  04                            .BYTE   4               ;PRINT ROUTINE NUMBER
 (1)  454C  CD  15  28      18.0          T11DE0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  454F  DA  89  48      10.0                  JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC

4552  CD  A4  48           18.0  T11CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
4555  C2  41  45           10.0           JNZ     T11LOOP         ;LOOP TILL 5 BYTES CHECKED
4558                                      ENDTST  T11RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                        RFQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4558                                      CALL    REQST
 (2)  4558  CD  06  28      18.0                .BYTE                   ;DATA PATTERN NUMBER
 (2)  455B  00                                  .WORD                   ;SYSTEM "" COUNT
 (2)  455C  00  00                              .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  455E  00  00                              .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4560  00                                  .BYTE   7               ;REQUEST CODE
 (1)  4562  3A  9A  4F      13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4565  3D               4.0          DCR     A               ;DOWNCOUNT
 (1)  4566  32  9A  4F      13.0          STA     ITERA           ;SAVE COUNT
 (1)  4569  F2  2C  45      10.0          JP      T11RPT          ;DO TEST UNTIL TILL   0
```

```
                                .SBTTL  TEST 12 - XMC - GCR 4X5 TRANSLATION SUBGROUP 11 TEST
456C                            ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
456C                            SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 1001(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 01001(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
456C                            SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
2502                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
2503                    ;*      :       :       ENDDO
2504                    ;*      :       :       IF TIMEOUT COUNT = 0
2505                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
2506                    ;*      :       :       :       ELSE - CONTINUE
2507                    ;*      :       :       ENDIF
2508                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
2509                    ;*      :       ENDDO
2510                    ;*      :       BGNDO
2511                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
2512                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
2513                    ;*      :       :       :       THEN - CONTINUE
2514                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
2515                    ;*      :       :       ENDIF
2516                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
2517                    ;*      :       :       BGNDO
2518                    ;*      :       :       :       CLOCK THE SYSTEM
2519                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
2520                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
2521                    ;*      :       :       ENDDO
2522                    ;*      :       :       IF 'WCLK' IS SET
2523                    ;*      :       :       :       THEN - CONTINUE
2524                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
2525                    ;*      :       :       ENDIF
2526                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
2527                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
2528                    ;*      :       ENDDO
2529                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
2530                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
2531                    ;*      ENDDO
2532                    ;*ENDTST
2533  456C             SE
 (1)                    ; ***************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
2534                    ;*XMC5 MICRO TEST 12
2535                    ;*XMC5 MICRO ERROR 12
2536                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1001(2)
2537                    ;*M8958, M8959, M8955'S
2538                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
2539                    ;*ACTUAL = NNNN
2540                    ;*EXPECTED = NNNN
2541                    ;*TRANSITION COUNT = LLL
2542                    ;*
2543                    ;*XMC5 MICRO TEST 12
2544                    ;*XMC5 MICRO ERROR 21
2545                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1001(2)
2546                    ;*M8958, M8959, M8955'S
2547                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
2548                    ;*FATAL ERROR - TEST ABORTED
2549                    ;*
2550                    ;*XMC5 MICRO TEST 12
2551                    ;*XMC5 MICRO ERROR 22
2552                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1001(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 12
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1001(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
456C                                    S
 (1)                                    ; ************************************************************************
456C                                    TEST12: TESTX   @12
 (1) 456C  3E  0A           7.0                      MVI     A,@12           ;DEFINE THE TEST NUMBER
 (1) 456E  CD  03  28      18.0                      CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 1001(2)
                                        ;&M8958, M8959, M8955'S

4571  3E  09                7.0  T12RPT: MVI     A,@11           ;SET FOR SUBGROUP 11 TEST
4573  32  D6  48           13.0          STA     XFRNUM          ;SAVE THE XFR NUMBER
4576  CD  52  47           18.0          CALL    FINDIT          ;GET THE PORT NUMBER
4579                                      SETLPC  T12LOOP,T12CONT
 (1) 4579  21  86  45        10.0                LXI     H,T12LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 (1) 457C  22  E4  48        16.0                SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1) 457F  21  97  45        10.0                LXI     H,T12CONT               ;GET THE CONTINUE ADDRESS
 (1) 4582  22  E7  48        16.0                SHLD    ERCADR+1        ;SAVE IT
 (1) 4585  AF                 4.0                XRA     A

4586  CD  2F  48           18.0  T12LOOP: CALL   XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
4589  CA  91  45           10.0          JZ      T12DE0          ;JUMP IF NO ERROR DETECTED
458C                                      ERRB    XDLOOP,T12DE0,4
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1) 458C  CD  12  28        18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000A                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 458F  0A                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 4590  04                                .BYTE   4               ;PRINT ROUTINE NUMBER
 (1) 4591  CD  15  28        18.0          T12DE0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1) 4594  DA  89  48        10.0                JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC

4597  CD  A4  48           18.0  T12CONT: CALL   NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
459A  C2  86  45           10.0          JNZ     T12LOOP         ;LOOP TILL 5 BYTES CHECKED
459D                                      ENDTST  T12RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2) 459D                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2) 459D  CD  06  28        18.0                CALL    REQST
 (2) 45A0  00                                    .BYTE                   ;DATA PATTERN NUMBER
 (2) 45A1  00  00                                .WORD                   ;SYSTEM "" COUNT
 (2) 45A3  00  00                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 45A5  00                                    .BYTE                   ;DATA COMPARE FLAG IF  1
 (2) 45A6  07                                    .BYTE   7               ;REQUEST CODE
 (1) 45A7  3A  9A  4F        13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1) 45AA  3D                 4.0          DCR     A               ;DOWNCOUNT
 (1) 45AB  32  9A  4F        13.0          STA     ITERA           ;SAVE COUNT
 (1) 45AE  F2  71  45        10.0          JP      T12RPT          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 13 - XMC - GCR 4X5 TRANSLATION SUBGROUP 12 TEST
45B1                            ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
45B1                            SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 1010(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 01010(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
45B1                            SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
2626                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
2627                    ;*      :       :       ENDDO
2628                    ;*      :       :       IF TIMEOUT COUNT = 0
2629                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
2630                    ;*      :       :       :       ELSE - CONTINUE
2631                    ;*      :       :       ENDIF
2632                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
2633                    ;*      :       ENDDO
2634                    ;*      :       BGNDO
2635                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
2636                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
2637                    ;*      :       :       :       THEN - CONTINUE
2638                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
2639                    ;*      :       :       ENDIF
2640                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
2641                    ;*      :       :       BGNDO
2642                    ;*      :       :       :       CLOCK THE SYSTEM
2643                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
2644                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
2645                    ;*      :       :       ENDDO
2646                    ;*      :       :       IF 'WCLK' IS SET
2647                    ;*      :       :       :       THEN - CONTINUE
2648                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
2649                    ;*      :       :       ENDIF
2650                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
2651                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
2652                    ;*      :       ENDDO
2653                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
2654                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
2655                    ;*      ENDDO
2656                    ;*ENDTST
2657  45B1              SE
 (1)                    ; ***************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
2658                    ;*XMC5 MICRO TEST 13
2659                    ;*XMC5 MICRO ERROR 13
2660                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1010(2)
2661                    ;*M8958, M8959, M8955'S
2662                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
2663                    ;*ACTUAL = NNNN
2664                    ;*EXPECTED = NNNN
2665                    ;*TRANSITION COUNT = LLL
2666                    ;*
2667                    ;*XMC5 MICRO TEST 13
2668                    ;*XMC5 MICRO ERROR 21
2669                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1010(2)
2670                    ;*M8958, M8959, M8955'S
2671                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
2672                    ;*FATAL ERROR - TEST ABORTED
2673                    ;*
2674                    ;*XMC5 MICRO TEST 13
2675                    ;*XMC5 MICRO ERROR 22
2676                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1010(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 13
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1010(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
45B1                                    S
 (1)                                    ; ************************************************************************
45B1                                    TEST13: TESTX   @13
 (1) 45B1  3E  0B          7.0                       MVI     A,@13           ;DEFINE THE TEST NUMBER
 (1) 45B3  CD  03  28     18.0                       CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 1010(2)
                                        ;&M8958, M8959, M8955'S

45B6  3E  0A              7.0   T13RPT: MVI     A,@12           ;SET FOR SUBGROUP 12 TEST
45B8  32  D6  48         13.0           STA     XFRNUM          ;SAVE THE XFR NUMBER
45BB  CD  52  47         18.0           CALL    FINDIT          ;GET THE PORT NUMBER
45BE                                    SETLPC  T13LOOP,T13CONT
 (1) 45BE  21  CB  45     10.0                   LXI     H,T13LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 (1) 45C1  22  E4  48     16.0                   SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1) 45C4  21  DC  45     10.0                   LXI     H,T13CONT               ;GET THE CONTINUE ADDRESS
 (1) 45C7  22  E7  48     16.0                   SHLD    ERCADR+1        ;SAVE IT
 (1) 45CA  AF               4.0                  XRA     A

45CB  CD  2F  48         18.0   T13LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
45CE  CA  D6  45         10.0            JZ      T13DEO          ;JUMP IF NO ERROR DETECTED
45D1                                    ERRB    XDLOOP,T13DE0,4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1) 45D1  CD  12  28     18.0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000B                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 45D4  0B                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 45D5  04                                       .BYTE   4               ;PRINT ROUTINE NUMBER
 (1) 45D6  CD  15  28     18.0                T13DEO::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1) 45D9  DA  89  48     10.0                       JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC

45DC  CD  A4  48         18.0   T13CONT: CALL    NXTBYT          ;CLOCK TIL NEXT BYTE AVAIL FROM XMC
45DF  C2  CB  45         10.0            JNZ     T13LOOP         ;LOOP TILL 5 BYTES CHECKED
45E2                                    ENDTST  T13RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2) 45E2                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2) 45E2  CD  06  28     18.0                       CALL    REQST
 (2) 45E5  00                                       .BYTE                   ;DATA PATTERN NUMBER
 (2) 45E6  00  00                                   .WORD                   ;SYSTEM "M" COUNT
 (2) 45E8  00  00                                   .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 45EA  00                                       .BYTE                   ;DATA COMPARE FLAG IF =1
 (2) 45EB  07                                       .BYTE   7               ;REQUEST CODE
 (1) 45EC  3A  9A  4F     13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1) 45EF  3D               4.0          DCR     A               ;DOWNCOUNT
 (1) 45F0  32  9A  4F     13.0           STA     ITERA           ;SAVE COUNT
 (1) 45F3  F2  B6  45     10.0           JP      T13RPT          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 14 - XMC - GCR 4X5 TRANSLATION SUBGROUP 13 TEST
45F6                            ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
45F6                            SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 1011(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 01011(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
45F6                            SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
2750                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
2751                    ;*      :       :       ENDDO
2752                    ;*      :       :       IF TIMEOUT COUNT = 0
2753                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
2754                    ;*      :       :       :       ELSE - CONTINUE
2755                    ;*      :       :       ENDIF
2756                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
2757                    ;*      :       ENDDO
2758                    ;*      :       BGNDO
2759                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
2760                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
2761                    ;*      :       :       :       THEN - CONTINUE
2762                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
2763                    ;*      :       :       ENDIF
2764                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
2765                    ;*      :       :       BGNDO
2766                    ;*      :       :       :       CLOCK THE SYSTEM
2767                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
2768                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
2769                    ;*      :       :       ENDDO
2770                    ;*      :       :       IF 'WCLK' IS SET
2771                    ;*      :       :       :       THEN - CONTINUE
2772                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
2773                    ;*      :       :       ENDIF
2774                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
2775                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
2776                    ;*      :       ENDDO
2777                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
2778                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
2779                    ;*      ENDDO
2780                    ;*ENDTST
2781  45F6              SE
 (1)                    ; ***************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
2782                    ;*XMC5 MICRO TEST 14
2783                    ;*XMC5 MICRO ERROR 14
2784                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1011(2)
2785                    ;*M8958, M8959, M8955'S
2786                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
2787                    ;*ACTUAL = NNNN
2788                    ;*EXPECTED = NNNN
2789                    ;*TRANSITION COUNT = LLL
2790                    ;*
2791                    ;*XMC5 MICRO TEST 14
2792                    ;*XMC5 MICRO ERROR 21
2793                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1011(2)
2794                    ;*M8958, M8959, M8955'S
2795                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
2796                    ;*FATAL ERROR - TEST ABORTED
2797                    ;*
2798                    ;*XMC5 MICRO TEST 14
2799                    ;*XMC5 MICRO ERROR 22
2800                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1011(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED!
                                        ;*
                                        ;*XMC5 MICRO TEST 14
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1011(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
45F6                                    S
 (1)                                    ; ************************************************************************
45F6                                    TEST14: TESTX   @14
 (1)  45F6  3E  0C            7.0                   MVI     A,@14           ;DEFINE THE TEST NUMBER
 (1)  45F8  CD  03  28       18.0                   CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 1011(2)
                                        ;&M8958, M8959, M8955'S

45FB  3E  0B                 7.0  T14RPT: MVI     A,@13           ;SET FOR SUBGROUP 13 TEST
45FD  32  D6  48            13.0         STA     XFRNUM          ;SAVE THE XFR NUMBER
4600  CD  52  47            18.0         CALL    FINDIT          ;GET THE PORT NUMBER
4603                                     SETLPC  T14LOOP,T14CONT
 (1)  4603  21  10  46       10.0                 LXI     H,T14LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 (1)  4606  22  E4  48       16.0                 SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  4609  21  21  46       10.0                 LXI     H,T14CONT               ;GET THE CONTINUE ADDRESS
 (1)  460C  22  E7  48       16.0                 SHLD    ERCADR+1        ;SAVE IT
 (1)  460F  AF                4.0                 XRA     A

4610  CD  2F  48            18.0  T14LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
4613  CA  1B  46            10.0           JZ      T14DEO          ;JUMP IF NO ERROR DETECTED
4616                                       ERRB    XDLOOP,T14DEO,4
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4616  CD  12  28       18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000C                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4619  0C                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  461A  04                                 .BYTE   4               ;PRINT ROUTINE NUMBER
 (1)  461B  CD  15  28       18.0         T14DEO::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  461E  DA  89  48       10.0                 JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC

4621  CD  A4  48            18.0  T14CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
4624  C2  10  46            10.0           JNZ     T14LOOP         ;LOOP TILL 5 BYTES CHECKED
4627                                       ENDTST  T14RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
                                        REQ     ?               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4627                                      CALL    REQST
 (2)  4627  CD  06  28       18.0               .BYTE                   ;DATA PATTERN NUMBER
 (2)  462A  00                                  .WORD                   ;SYSTEM "" COUNT
 (2)  462B  00  00                              .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  462D  00  00                              .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  462F  00                                  .BYTE   7               ;REQUEST CODE
 (2)  4630  07
 (1)  4631  3A  9A  4F       13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1)  4634  3D                4.0           DCR     A               ;DOWNCOUNT
 (1)  4635  32  9A  4F       13.0           STA     ITERA           ;SAVE COUNT
 (1)  4638  F2  FB  45       10.0           JP      T14RPT          ;DO TEST UNTIL TILL = 0
```

```
2829                                .SBTTL  TEST 15 - XMC - GCR 4X5 TRANSLATION SUBGROUP 14 TEST
2830  463B                         ST
 (1)                                ; ***************************************************************************
 (1)                                ;*TEST TITLE
 (1)                                ;*---- -----
2831                                ;*XMC - GCR 4 X 5 TEST
2832  463B                         SD
 (1)                                ; ***************************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
2833                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
2834                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
2835                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
2836                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 1100(2) TO TEST TRANSLATION ROM
2837                                ;*PATTERN 11110(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
2838                                ;*
2839                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
2840                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
2841                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
2842                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
2843  463B                         SP
 (1)                                ; ***************************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
2844                                ;*BGNTST
2845                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
2846                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
2847                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
2848                                ;*      :       ELSE - CONTINUE
2849                                ;*      ENDIF
2850                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
2851                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
2852                                ;*      BGNDO
2853                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
2854                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
2855                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
2856                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
2857                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
2858                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
2859                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
2860                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
2861                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
2862                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
2863                                ;*      :       BGNDO
2864                                ;*      :       :       CLOCK THE SYSTEM
2865                                ;*      :       :       DECREMENT THE LOOP COUNT
2866                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
2867                                ;*      :       ENDDO
2868                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
2869                                ;*      :       BGNDO
2870                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
2871                                ;*      :       :       BGNDO
2872                                ;*      :       :       :       CLOCK THE SYSTEM
2873                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
2874                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
2875                    ;*      :       :       ENDDO
2876                    ;*      :       :       IF TIMEOUT COUNT = 0
2877                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
2878                    ;*      :       :       :       ELSE - CONTINUE
2879                    ;*      :       :       ENDIF
2880                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
2881                    ;*      :       ENDDO
2882                    ;*      :       BGNDO
2883                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
2884                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
2885                    ;*      :       :       :       THEN - CONTINUE
2886                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
2887                    ;*      :       :       ENDIF
2888                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
2889                    ;*      :       :       BGNDO
2890                    ;*      :       :       :       CLOCK THE SYSTEM
2891                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
2892                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
2893                    ;*      :       :       ENDDO
2894                    ;*      :       :       IF 'WCLK' IS SET
2895                    ;*      :       :       :       THEN - CONTINUE
2896                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
2897                    ;*      :       :       ENDIF
2898                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
2899                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
2900                    ;*      :       ENDDO
2901                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
2902                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
2903                    ;*      ENDDO
2904                    ;*ENDTST
2905  463B              SE
 (1)                    ; ***************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
2906                    ;*XMC5 MICRO TEST 15
2907                    ;*XMC5 MICRO ERROR 15
2908                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1100(2)
2909                    ;*M8958, M8959, M8955'S
2910                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
2911                    ;*ACTUAL = NNNN
2912                    ;*EXPECTED = NNNN
2913                    ;*TRANSITION COUNT = LLL
2914                    ;*
2915                    ;*XMC5 MICRO TEST 15
2916                    ;*XMC5 MICRO ERROR 21
2917                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1100(2)
2918                    ;*M8958, M8959, M8955'S
2919                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
2920                    ;*FATAL ERROR - TEST ABORTED
2921                    ;*
2922                    ;*XMC5 MICRO TEST 15
2923                    ;*XMC5 MICRO ERROR 22
2924                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1100(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED.
                                        ;*
                                        ;*XMC5 MICRO TEST 15
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1100(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
463B                                    $
 (1)                                    ; *****************************************************************************
463B                                    TEST15: TESTX   @15
 (1) 463B  3E  0D           7.0                  MVI     A,@15           ;DEFINE THE TEST NUMBER
 (1) 463D  CD  03  28      18.0                  CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 1100(2)
                                        ;&M8958, M8959, M8955'S

4640  3E  0C           7.0  T15RPT: MVI     A,@14           ;SET FOR SUBGROUP 14 TEST
4642  32  D6  48        13.0         STA     XFRNUM          ;SAVE THE XFR NUMBER
4645  CD  52  47        18.0         CALL    FINDIT          ;GET THE PORT NUMBER
4648                               SETLPC  T15LOOP,T15CONT
 (1) 4648  21  55  46      10.0                  LXI     H,T15LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 (1) 464B  22  E4  48      16.0                  SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1) 464E  21  66  46      10.0                  LXI     H,T15CONT               ;GET THE CONTINUE ADDRESS
 (1) 4651  22  E7  48      16.0                  SHLD    ERCADR+1        ;SAVE IT
 (1) 4654  AF               4.0                  XRA     A
4655  CD  2F  48        18.0  T15LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
4658  CA  60  46        10.0         JZ      T15DEO          ;JUMP IF NO ERROR DETECTED
465B                               ERRB    XDLOOP,T15DEO,4
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1) 465B  CD  12  28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000D                                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 465E  0D                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 465F  04                                    .BYTE   4               ;PRINT ROUTINE NUMBER
 (1) 4660  CD  15  28      18.0                T15DEO::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1) 4663  DA  89  48      10.0                  JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC

4666  CD  A4  48        18.0  T15CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
4669  C2  55  46        10.0         JNZ     T15LOOP         ;LOOP TILL 5 BYTES CHECKED
466C                               ENDTST  T15RPT
 (1)                               ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2) 466C                          REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2) 466C  CD  06  28      18.0                  CALL    REQST
 (2) 466F  00                                    .BYTE                   ;DATA PATTERN NUMBER
 (2) 4670  00  00                                .WORD                   ;SYSTEM "" COUNT
 (2) 4672  00  00                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 4674  00                                    .BYTE                   ;DATA COMPARE FLAG IF -1
 (2) 4675  07                                    .BYTE   7               ;REQUEST CODE
 (1) 4676  3A  9A  4F      13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1) 4679  3D               4.0         DCR     A               ;DOWNCOUNT
 (1) 467A  32  9A  4F      13.0         STA     ITERA           ;SAVE COUNT
 (1) 467D  F2  40  46      10.0         JP      T15RPT          ;DO TEST UNTIL TILL - 0
```

```
                                .SBTTL  TEST 16 - XMC - GCR 4X5 TRANSLATION SUBGROUP 15 TEST
 4680                           ST
  (1)                           ;*********************************************************************
  (1)                           ;*TEST TITLE
  (1)                           ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
 4680                           SD
  (1)                           ;*********************************************************************
  (1)                           ;*DESCRIPTION
  (1)                           ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 1101(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 01101(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
 4680                           SP
  (1)                           ;*********************************************************************
  (1)                           ;*PROCEDURE
  (1)                           ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       :       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER - 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
                                ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
                                ;*      :       :       ENDDO
                                ;*      :       :       IF TIMEOUT COUNT = 0
                                ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
                                ;*      :       :       :       ELSE - CONTINUE
                                ;*      :       :       ENDIF
                                ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
                                ;*      :       ENDDO
                                ;*      :       BGNDO
                                ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
                                ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
                                ;*      :       :       :       THEN - CONTINUE
                                ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
                                ;*      :       :       ENDIF
                                ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       :       DECREMENT THE LOOP COUNTER
                                ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       :       ENDDO
                                ;*      :       :       IF 'WCLK' IS SET
                                ;*      :       :       :       THEN - CONTINUE
                                ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
                                ;*      :       :       ENDIF
                                ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
                                ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
                                ;*      :       ENDDO
                                ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
                                ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
                                ;*      ENDDO
                                ;*ENDTST
4680                            SE
 (1)                            ; ************************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*XMC5 MICRO TEST 16
                                ;*XMC5 MICRO ERROR 16
                                ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1101(2)
                                ;*M8958, M8959, M8955'S
                                ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
                                ;*ACTUAL   NNNN
                                ;*EXPECTED = NNNN
                                ;*TRANSITION COUNT = LLL
                                ;*
                                ;*XMC5 MICRO TEST 16
                                ;*XMC5 MICRO ERROR 21
                                ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1101(2)
                                ;*M8958, M8959, M8955'S
                                ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*XMC5 MICRO TEST 16
                                ;*XMC5 MICRO ERROR 22
                                ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1101(2)
```

```
3049                                    ;*M8958, M8959, M8955'S
3050                                    ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
3051                                    ;*FATAL ERROR - TEST ABORTED!
3052                                    ;*
3053                                    ;*XMC5 MICRO TEST 16
3054                                    ;*XMC5 MICRO ERROR 23
3055                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1101(2)
3056                                    ;*M8958, M8959, M8955'S
3057                                    ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
3058  4680                              S
 (1)                                    ; *****************************************************************************
3059  4680                              TEST16: TESTX   @16
 (1)  4680  3E  0E           7.0                        MVI     A,@16           ;DEFINE THE TEST NUMBER
 (1)  4682  CD  03  28      18.0                        CALL    TSET            ;SETUP THE TEST
3060                                    ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 1101(2)
3061                                    ;&M8958, M8959, M8955'S
3062
3063  4685  3E  0D           7.0  T16RPT: MVI     A,@15           ;SET FOR SUBGROUP 15 TEST
3064  4687  32  D6  48       13.0         STA     XFRNUM          ;SAVE THE XFR NUMBER
3065  468A  CD  52  47       18.0         CALL    FINDIT          ;GET THE PORT NUMBER
3066  468D                              SETLPC  T16LOOP,T16CONT
 (1)  468D  21  9A  46       10.0                LXI     H,T16LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 (1)  4690  22  E4  48       16.0                SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  4693  21  AB  46       10.0                LXI     H,T16CONT               ;GET THE CONTINUE ADDRESS
 (1)  4696  22  E7  48       16.0                SHLD    ERCADR+1        ;SAVE IT
 (1)  4699  AF                4.0                XRA     A
3067
3068  469A  CD  2F  48       18.0  T16LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
3069  469D  CA  A5  46       10.0           JZ      T16DEO          ;JUMP IF NO ERROR DETECTED
3070  46A0                           ERRB    XDLOOP,T16DEO,4
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46A0  CD  12  28       18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000E                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46A3  0E                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46A4  04                               .BYTE   4               ;PRINT ROUTINE NUMBER
 (1)  46A5  CD  15  28       18.0        T16DEO::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46A8  DA  89  48       10.0                JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
3071                                    ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC
3072
3073  46AB  CD  A4  48       18.0  T16CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
3074  46AE  C2  9A  46       10.0           JNZ     T16LOOP         ;LOOP TILL 5 BYTES CHECKED
3075  46B1                           ENDTST  T16RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  46B1                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  46B1  CD  06  28       18.0                CALL    REQST
 (2)  46B4  00                               .BYTE                   ;DATA PATTERN NUMBER
 (2)  46B5  00  00                           .WORD                   ;SYSTEM "" COUNT
 (2)  46B7  00  00                           .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  46B9  00                               .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  46BA  07                               .BYTE   7               ;REQUEST CODE
 (1)  46BB  3A  9A  4F       13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1)  46BE  3D                4.0           DCR     A               ;DOWNCOUNT
 (1)  46BF  32  9A  4F       13.0           STA     ITERA           ;SAVE COUNT
 (1)  46C2  F2  85  46       10.0           JP      T16RPT          ;DO TEST UNTIL TILL - 0
```

```
                                .SBTTL  TEST 17 - XMC - GCR 4X5 TRANSLATION SUBGROUP 16 TES
                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*XMC - GCR 4 X 5 TEST
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
                                ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
                                ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
                                ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 1110(2) TO TEST TRANSLATION ROM
                                ;*PATTERN 01110(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
                                ;*
                                ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
                                ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
                                ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
                                ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TE T STARTED.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
                                ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
                                ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
                                ;*      :       ELSE - CONTINUE
                                ;*      ENDIF
                                ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
                                ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
                                ;*      BGNDO
                                ;*      :       INIT THE INTERNAL COUNTER REGISTERS
                                ;*      .       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
                                ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
                                ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
                                ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
                                ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
                                ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
                                ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
                                ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
                                ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
                                ;*      :       BGNDO
                                ;*      :       :       CLOCK THE SYSTEM
                                ;*      :       .       DECREMENT THE LOOP COUNT
                                ;*      :       :       DO UNTIL LOOP COUNTER = 0
                                ;*      :       ENDDO
                                ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
                                ;*      :       BGNDO
                                ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
                                ;*      :       :       BGNDO
                                ;*      :       :       :       CLOCK THE SYSTEM
                                ;*      :       :       .       DECREMENT THE LOOP COUNT
```

```
3122                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
3123                    ;*      :       :       ENDDO
3124                    ;*      :       :       IF TIMEOUT COUNT = 0
3125                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
3126                    ;*      :       :       :       ELSE - CONTINUE
3127                    ;*      :       :       ENDIF
3128                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
3129                    ;*      :       ENDDO
3130                    ;*      :       BGNDO
3131                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
3132                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
3133                    ;*      :       :       :       THEN - CONTINUE
3134                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
3135                    ;*      :       :       ENDIF
3136                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
3137                    ;*      :       :       BGNDO
3138                    ;*      :       :       :       CLOCK THE SYSTEM
3139                    ;*      :       :       :       DECREMENT THE LOOP COUNTER
3140                    ;*      :       :       :       DO UNTIL LOOP COUNTER = 0
3141                    ;*      :       :       ENDDO
3142                    ;*      :       :       IF 'WCLK' IS SET
3143                    ;*      :       :       :       THEN - CONTINUE
3144                    ;*      :       :       :       ELSE - REPORT IMPROPER TRANSITION RATE
3145                    ;*      :       :       ENDIF
3146                    ;*      :       :       POINT TO THE NEXT SET Of EXPECTED DATA IN TABLE
3147                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
3148                    ;*      :       ENDDO
3149                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
3150                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
3151                    ;*      ENDDO
3152                    ;*ENDTST
3153  46C5              SE
 (1)                    ; ***************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
3154                    ;*XMC5 MICRO TEST 17
3155                    ;*XMC5 MICRO ERROR 17
3156                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1110(2)
3157                    ;*M8958, M8959, M8955'S
3158                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
3159                    ;*ACTUAL = NNNN
3160                    ;*EXPECTED = NNNN
3161                    ;*TRANSITION COUNT = LLL
3162                    ;*
3163                    ;*XMC5 MICRO TEST 17
3164                    ;*XMC5 MICRO ERROR 21
3165                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1110(2)
3166                    ;*M8958, M8959, M8955'S
3167                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
3168                    ;*FATAL ERROR - TEST ABORTED
3169                    ;*
3170                    ;*XMC5 MICRO TEST 17
3171                    ;*XMC5 MICRO ERROR 22
3172                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1110(2)
```

```
                                        ;*M8958, M8959, M8955'S
                                        ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
                                        ;*FATAL ERROR - TEST ABORTED.
                                        ;*
                                        ;*XMC5 MICRO TEST 17
                                        ;*XMC5 MICRO ERROR 23
                                        ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1110(2)
                                        ;*M8958, M8959, M8955'S
                                        ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
46C5                                    S
 (1)                                    ; ************************************************************************************
46C5                                    TEST17: TESTX   @17
 (1)  46C5  3E  0F          7.0                      MVI     A,@17           ;DEFINE THE TEST NUMBER
 (1)  46C7  CD  03  28     18.0                      CALL    TSET            ;SETUP THE TEST
                                        ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 1110(2)
                                        ;&M8958, M8959, M8955'S

46CA  3F  0E               7.0  T17RPT: MVI     A,@16           ;SET FOR SUBGROUP 16 TEST
46CC  32  D6  48          13.0          STA     XFRNUM          ;SAVE THE XFR NUMBER
46CF  CD  52  47          18.0          CALL    FINDIT          ;GET THE PORT NUMBER
46D2                                    SETLPC  T17LOOP,T17CONT
 (1)  46D2  21  DF  46     10.0                  LXI     H,T17LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 (1)  46D5  22  E4  48     16.0                  SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  46D8  21  F0  46     10.0                  LXI     H,T17CONT               ;GET THE CONTINUE ADDRESS
 (1)  46DB  22  E7  48     16.0                  SHLD    ERCADR+1        ;SAVE IT
 (1)  46DE  AF              4.0                  XRA     A

46DF  CD  2F  48          18.0  T17LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
46E2  CA  EA  46          10.0           JZ      T17DE0          ;JUMP IF NO ERROR DETECTED
46E5                                     ERRB    XDLOOP,T17DE0,4
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46E5  CD  12  28     18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000F                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46E8  0F                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46E9  04                                  .BYTE   4               ;PRINT ROUTINE NUMBER
 (1)  46EA  CD  15  28     18.0          T17DE0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46ED  DA  89  48     10.0                  JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC

46F0  CD  A4  48          18.0  T17CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
46F3  C2  DF  46          10.0           JNZ     T17LOOP         ;LOOP TILL 5 BYTES CHECKED
46F6                                     ENDTST  T17RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  46F6                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  46F6  CD  06  28     18.0                  CALL    REQST
 (2)  46F9  00                                  .BYTE                   ;DATA PATTERN NUMBER
 (2)  46FA  00  00                              .WORD                   ;SYSTEM "" COUNT
 (2)  46FC  00  00                              .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  46FE  00                                  .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  46FF  07                                  .BYTE   7               ;REQUEST CODE
 (1)  4700  3A  9A  4F     13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4703  3D              4.0          DCR     A               ;DOWNCOUNT
 (1)  4704  32  9A  4F     13.0          STA     ITERA           ;SAVE COUNT
 (1)  4707  F2  CA  46     10.0          JP      T17RPT          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 20 - XMC - GCR 4X5 TRANSLATION SUBGROUP 17 TEST
3202  470A                     ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*----- -----
3203                            ;*XMC - GCR 4 X 5 TEST
3204  470A                     SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
3205                            ;*THIS TEST WILL COMMAND THE WRITE PATH TO ISSUE 16. GCR DATA TRANSFERS.
3206                            ;*EACH DATA TRANSFER WILL BE 4 BYTES IN LENGTH (1 SUBGROUP) AND WILL
3207                            ;*CONTAIN DATA THAT WILL CAUSE ALL 4 X 5 TRANSLATIONS TO BE ENVOKED. THE
3208                            ;*FIRST DATA TRANSFER WILL BE 4 BYTES OF 1111(2) TO TEST TRANSLATION ROM
3209                            ;*PATTERN 01111(2) - TO BE CHECKED AT THE SELECTED PORT BOARD.
3210                            ;*
3211                            ;*THE PREAMBLE, MARK 1 BURST, AND THE END MARK BURST WILL BE IGNORED
3212                            ;*DURING THIS TEST. ONLY THE DATA FROM THE TRANSLATED SUBGROUP WILL BE
3213                            ;*CHECKED. AFTER THE SUBGROUP DATA HAS BEEN TESTED FOR CORRECTNESS, THE
3214                            ;*DATA TRANSFER WILL BE ABORTED AND TGE NEXT DATA GROUP TEST STARTED.
3215  470A                     SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
3216                            ;*BGNTST
3217                            ;*      REQUEST A TU PORT FROM THE 'HOST' CPU
3218                            ;*      IF NO TU PORT SELECTED FROM THE 'HOST' CPU
3219                            ;*      :       THEN - REPORT A SELECT ERROR AND ABORT THE TEST
3220                            ;*      :       ELSE - CONTINUE
3221                            ;*      ENDIF
3222                            ;*      CALL SUBROUTINE 'CLEAR' TO INIT THE PORTS
3223                            ;*      INIT THE 'HOST' SELECTED PORT FOR TESTING DATA FROM TRANSLATOR
3224                            ;*      BGNDO
3225                            ;*      :       INIT THE INTERNAL COUNTER REGISTERS
3226                            ;*      :       INIT THE EXPECTED DATA TABLE POINTER & SUPPLY DATA POINTER
3227                            ;*      :       RESTART THE WMC/XMC - WRITE PATH & SELECT SINGLE STEP
3228                            ;*      :       LOAD THE BYTE COUNTER FOR A 4 BYTE DATA TRANSFER (COUNTER SET TO 1)
3229                            ;*      :       LOAD THE PAD COUNTER FOR 3 PAD CHARACTERS
3230                            ;*      :       SET DDR CONTROL TO 'IN' FOR DATA INJECTION TO WMC
3231                            ;*      :       SELECT PACKING MODE TO 'IMAGE' (1 SCLK = 1 BYTE OF DATA)
3232                            ;*      :       LOAD THE RESIDUAL CHAR FOR 4 DATA BYTES IN THE RESID GROUP
3233                            ;*      :       ENABLE ALL TRACKS (TRKENA REGISTER)
3234                            ;*      :       SETUP A LOOP COUNT TO CLOCK THRU THE WMC RESTART AREA
3235                            ;*      :       BGNDO
3236                            ;*      :       :       CLOCK THE SYSTEM
3237                            ;*      :       :       DECREMENT THE LOOP COUNT
3238                            ;*      :       :       DO UNTIL LOOP COUNTER = 0
3239                            ;*      :       ENDDO
3240                            ;*      :       ENABLE THE WRITE PATH FOR A GCR DATA TRANSFER
3241                            ;*      :       BGNDO
3242                            ;*      :       :       SETUP A WATCHDOG TIMEOUT LOOP COUNT
3243                            ;*      :       :       BGNDO
3244                            ;*      :       :       :       CLOCK THE SYSTEM
3245                            ;*      :       :       :       DECREMENT THE LOOP COUNT
```

```
3246                    ;*      :       :       :       DO UNTIL COUNT = 0 OR XMC WRITE CLOCK SET
3247                    ;*      :       :       ENDDO
3248                    ;*      :       :       IF TIMEOUT COUNT = 0
3249                    ;*      :       :       :       THEN - REPORT FAILURE AND ABORT (NO WCLK)
3250                    ;*      :       :       :       ELSE - CONTINUE
3251                    ;*      :       :       ENDIF
3252                    ;*      :       :       DO UNTIL WCLK AT START OF DATA SUBGROUP FROM XMC
3253                    ;*      :       ENDDO
3254                    ;*      :       BGNDO
3255                    ;*      :       :       GET ACTUAL DATA BYTE FROM XMC (IN PORT BOARD)
3256                    ;*      :       :       IF EXPECTED DATA = ACTUAL DATA FROM XMC
3257                    ;*      :       :       :       THEN - CONTINUE
3258                    ;*      :       :       :       ELSE - REPORT 4 X 5 TRANSLATION ERROR
3259                    ;*      :       :       ENDIF
3260                    ;*      :       :       SET LOOP COUNTER TO 14(8) FOR NEXT WCLK SIGNAL
3261                    ;*      :       :       BGNDO
3262                    ;*      :       :       .       CLOCK THE SYSTEM
3263                    ;*      :       :       .       DECREMENT THE LOOP COUNTER
3264                    ;*      .       :       :       DO UNTIL LOOP COUNTER = 0
3265                    ;*      :       :       ENDDO
3266                    ;*      :       :       IF 'WCLK' IS SET
3267                    ;*      :       :       :       THEN - CONTINUE
3268                    ;*      :       :       .       ELSE - REPORT IMPROPER TRANSITION RATE
3269                    ;*      :       :       ENDIF
3270                    ;*      :       :       POINT TO THE NEXT SET OF EXPECTED DATA IN TABLE
3271                    ;*      :       :       DO UNTIL 5 DATA BYTES CHECKED (1 SUBGROUP)
3272                    ;*      :       ENDDO
3273                    ;*      :       POINT TO THE NEXT SUBGROUP EXPECTED DATA TABLE
3274                    ;*      :       DO UNTIL ALL 16. DATA TRANSFERS HAVE BEEN DONE
3275                    ;*      ENDDO
3276                    ;*ENDTST
3277  470A              SE
 (1)                    ; ***************************************************************************
 (1)                    ;*ERRORS
 (1)                    ;*------
3278                    ;*XMC5 MICRO TEST 20
3279                    ;*XMC5 MICRO ERROR 20
3280                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1111(2)
3281                    ;*M8958, M8959, M8955'S
3282                    ;*SUBGROUP 4X5 DATA INCORRECT FROM XMC
3283                    ;*ACTUAL = NNNN
3284                    ;*EXPECTED = NNNN
3285                    ;*TRANSITION COUNT = LLL
3286                    ;*
3287                    ;*XMC5 MICRO TEST 20
3288                    ;*XMC5 MICRO ERROR 21
3289                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1111(2)
3290                    ;*M8958, M8959, M8955'S
3291                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
3292                    ;*FATAL ERROR - TEST ABORTED
3293                    ;*
3294                    ;*XMC5 MICRO TEST 20
3295                    ;*XMC5 MICRO ERROR 22
3296                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1111(2)
```

```
3297                                    ;*M8958, M8959, M8955'S
3298                                    ;*WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
3299                                    ;*FATAL ERROR - TEST ABORTED.
3300                                    ;*
3301                                    ;*XMC5 MICRO TEST 20
3302                                    ;*XMC5 MICRO ERROR 23
3303                                    ;*XMC5-4X5 TRANSLATION TEST SUBGROUP 1111(2)
3304                                    ;*M8958, M8959, M8955'S
3305                                    ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
3306  470A                              $
 (1)                                    ; *****************************************************************************
3307  470A                              TEST20: TESTX   @20
 (1)  470A  3E  10           7.0                  MVI     A,@20           ;DEFINE THE TEST NUMBER
 (1)  470C  CD  03  28      18.0                  CALL    TSET            ;SETUP THE TEST
3308                                    ;%XMC5-GCR-4X5 TRANSLATION TEST SUBGROUP 1111(2)
3309                                    ;&M8958, M8959, M8955'S
3310  470F  3E  0F           7.0       T20RPT: MVI     A,@17           ;SET FOR SUBGROUP 17 TEST
3311  4711  32  D6  48      13.0               STA     XFRNUM          ;SAVE THE XFR NUMBER
3312  4714  CD  52  47      18.0               CALL    FINDIT          ;GET THE PORT NUMBER
3313  4717                               SETLPC  T20LOOP,T20CONT
 (1)  4717  21  24  47      10.0                       LXI     H,T20LOOP               ;GET THE LOOP-ON-ERR ADDRESS
 (1)  471A  22  E4  48      16.0                       SHLD    LPCADR+1        ;SAVE FOR ERROR LOOPING
 (1)  471D  21  35  47      10.0                       LXI     H,T20CONT               ;GET THE CONTINUE ADDRESS
 (1)  4720  22  E7  48      16.0                       SHLD    ERCADR+1        ;SAVE IT
 (1)  4723  AF               4.0                       XRA     A
3314
3315  4724  CD  2F  48      18.0  T20LOOP: CALL    XFRSUB          ;CLOCK DATA THRU TRANSLATOR & CHECK IT
3316  4727  CA  2F  47      10.0           JZ      T20DE0          ;JUMP IF NO ERROR DETECTED
3317  472A                                 ERRB    XDLOOP,T20DE0,4
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  472A  CD  12  28      18.0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0010                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  472D  10                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  472E  04                                       .BYTE   4               ;PRINT ROUTINE NUMBER
 (1)  472F  CD  15  28      18.0                T20DE0::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4732  DA  89  48      10.0                       JC      XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
3318                                    ;>SUBGROUP 4X5 DATA INCORRECT FROM XMC
3319  4735  CD  A4  48      18.0  T20CONT: CALL    NXTBYT          ;CLOCK TILL NEXT BYTE AVAIL FROM XMC
3320  4738  C2  24  47      10.0           JNZ     T20LOOP         ;LOOP TILL 5 BYTES CHECKED
3321  473B                                 ENDTST  T20RPT
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  473B                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  473B  CD  06  28      18.0                       CALL    REQST
 (2)  473E  00                                       .BYTE                   ;DATA PATTERN NUMBER
 (2)  473F  00  00                                   .WORD                   ;SYSTEM "" COUNT
 (2)  4741  00  00                                   .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4743  0C                                       .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4744  07                                       .BYTE   7               ;REQUEST CODE
 (1)  4745  3A  9A  4F      13.0                LDA     ITERA           ;GET ITERATION COUNT
 (1)  4748  3D               4.0                DCR     A               ;DOWNCOUNT
 (1)  4749  32  9A  4F      13.0                STA     ITERA           ;SAVE COUNT
 (1)  474C  F2  0F  47      10.0                JP      T20RPT          ;DO TEST UNTIL TILL = 0
3322  474F  C3  18  28      10.0                JMP     TSTEND          ;ALL DONE ... EXIT
```

```
3324                                    .SBTTL  FINDIT - SUBROUTINE TO GET A TEST PORT FOR TESTING USE
3325                                    ;REQUEST THE 'HOST' CPU TO SELECT A PORT FOR TESTING
3326
3327  4752                              FINDIT: REQ     7
 (1)  4752  CD  06  28      18.0                 CALL    REQST
 (1)  4755  00                                   .BYTE                   ;DATA PATTERN NUMBER
 (1)  4756  00  00                               .WORD                   ;SYSTEM '"' COUNT
 (1)  4758  00  00                               .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  475A  00                                   .BYTE                   ;DATA COMPARE FLAG IF ='
 (1)  475B  07                                   .BYTE   7               ;REQUEST CODE
3328  475C                                    RIN     R12L            ;GET THE SELECTED PORT FROM 'HOST'
 (1)  475C  DB  94          10.0                 IN      R12L                ;READ R12L INTO AC
 (1)  475E  7F               4.0                 MOV     A,A                 ;RETRY LINK
3329  475F  32  D7  48       13.0         STA     UNITMP          ;SAVE IT
3330  4762  A7                4.0         ANA     A               ;SET THE CONDITION BITS
3331  4763  C2  74  47       10.0         JNZ     FOUND           ;SEE IF ANY WERE SELECTED IN 'HOST'
3332  4766                               ERR     TSTEND,NOUNIT
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4766  CD  09  28      18.0                 CALL    ERLP                ;PROCESS ERROR - DO 2.3
 (1)       0011                              MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4769  11                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  476A  00                               .BYTE
 (1)  476B  CD  15  28      18.0         NOUNIT::        CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.3
 (1)  476E  DA  18  28      10.0                 JC      TSTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
3333                                    ;>OPERATOR ERROR - NO TU78 UNIT SELECTED
3334                                    ;>FATAL ERROR - TEST ABORTED
3335  4771  C3  18  28      10.0         JMP     TSTEND
3336                                    ;HERE IF THE HOST CPU INDICATED A PORT TO USE
3337
3338  4774  CD  22  4A      18.0  FOUND:  CALL    CLEAR           ;INIT ALL THE PORTS IN THE PORT BOARD
3339  4777  3A  D7  48       13.0         LDA     UNITMP          ;GET THE RESPONSE FROM 'HOST' CPU
3340  477A  06  00            7.0         MVI     B,0             ;INIT 'B'
3341  477C  E6  01            7.0         ANI     1               ;IS UNIT 0 SELECTED?
3342  477E  C2  94  47       10.0         JNZ     FPORT           ;YUP - FOUND THE PORT TO USE
3343  4781  04                4.0         INR     B               ;NO - COUNT THE PORT
3344  4782  3A  D7  48       13.0         LDA     UNITMP          ;GET RESPONSE AGAIN
3345  4785  E6  02            7.0         ANI     2               ;PORT 1 SELECTED?
3346  4787  C2  94  47       10.0         JNZ     FPORT           ;YUP
3347  478A  04                4.0         INR     B               ;COUNT THE PORT
3348  478B  3A  D7  48       13.0         LDA     UNITMP          ;GET RESPONSE
3349  478E  E6  04            7.0         ANI     4               ;PORT 2 SELECTED?
3350  4790  C2  94  47       10.0         JNZ     FPORT           ;YUP
3351  4793  04                4.0         INR     B               ;NO - ASSUME PORT 3
3352
3353  4794                          FPORT:
3354  4794  DB  E0           10.0         IN      INTSTA          ;GET THE PORT SELECT STATUS
3355  4796  E6  80            7.0         ANI     BIT7
3356  4798  B0                4.0         ORA     B               ;ADD IN THE SELECTED PORT #
3357  4799  D3  E0           10.0         OUT     MBSEL           ;SELECT THE DESIRED PORT TO USE
3358  479B  3E  40            7.0         MVI     A,P.LCS         ;GET THE LOOP-COMMAND-STATUS BIT
3359  479D  D3  48           10.0         OUT     PDIAG           ;SELECT IT
3360  479F  AF                4.0         XRA     A
3361  47A0  D3  40           10.0         OUT     TCMD            ;CLEAR PORT COMMAND
3362  47A2  3E  10            7.0         MVI     A,P.WPEN        ;SET WRITE PATH ENABLE
```

```
3363  47A4  D3  4C          10.0        OUT     PENAB           ;NOW
3364
3365                                ;INIT THE COUNTER CONTROL AND COUNTERS
3366
3367  47A6                          INICNT
 (1)  47A6  3E  30           7.0                 MVI     A,@060          ;INIT THE
 (1)  47A8  D3  D7          10.0                 OUT     CNTCTL          ;BYTE COUNTER
 (1)  47AA  3E  70           7.0                 MVI     A,@160          ;INIT THE
 (1)  47AC  D3  D7          10.0                 OUT     CNTCTL          ;PAD COUNTER
 (1)  47AE  3E  B0           7.0                 MVI     A,@260          ;INIT THE
 (1)  47B0  D3  D7          10.0                 OUT     CNTCTL          ;ERROR COUNTER
3368  47B2                          CLRECT
 (1)  47B2  AF               4.0                 XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  47B3  D3  D6          10.0                 OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  47B5  D3  D6          10.0                 OUT     ERRCNT          ;CLEAR BITS 15-8
3369
3370                                ;INIT THE SUPPLY TABLE INDEX POINTER
3371  47B7  3E  FF           7.0          MVI    A,@377
3372  47B9  32  E1  48      13.0          STA    DPNUM
3373  47BC  AF               4.0          XRA    A
3374  47BD  32  DF  48      13.0          STA    CLKCNT          ;INIT THE 'SCLK' COUNTER
3375  47C0  32  D3  48      13.0          STA    TRANSCT         ;INIT THE TRANSITION COUNTER
3376  47C3                               ROUT   R05H            ;CLEAR HIGH BYTE OF TRANS CNT REG (LAS)
 (1)  47C3  D3  8B          10.0                 OUT     R05H            ;WRITE AC INTO R05H
 (1)  47C5  7F               4.0                 MOV     A,A             ;RETRY LINK
3377                                ;INIT POINTERS FOR TEST PASS
3378
3379  47C6  3A  D6  48      13.0          LDA    XFRNUM          ;GET THE SUBGROUP NUMBER (0 - 17)
3380  47C9  17               4.0          RAL                    ;MULT BY 4 - 2 LOCATIONS PER ENTRY
3381  47CA  17               4.0          RAL
3382  47CB  E6  FC           7.0          ANI    @374            ;STRIP OFF THE TWO LOW BITS
3383  47CD  4F               4.0          MOV    C,A             ;SAVE OFFSET INTO THE TABLE
3384  47CE  06  00           7.0          MVI    B,0
3385  47D0  21  E9  48      10.0          LXI    H,XFRTBL        ;POINT TO THE TABLE - BASIC POINTER
3386  47D3  09              10.0          DAD    B               ;ADD IN THE OFFSET VALUE TO TABLE
3387  47D4  5E               7.0          MOV    E,M             ;GET THE 'SUPPLY' DATA ADDRESS LOW BYTE
3388  47D5  23               6.0          INX    H               ;POINT TO THE HIGH BYTE OF ADDRESS
3389  47D6  56               7.0          MOV    D,M             ;GET THE HIGH BYTE OF ADDRESS
3390  47D7  EB               4.0          XCHG
3391  47D8  22  D8  48      16.0          SHLD   SUPADR          ;SAVE THE ADDRESS OF SUPPLY DATA TABLE
3392  47DB  EB               4.0          XCHG
3393  47DC  23               6.0          INX    H               ;POINT TO THE EXPECTED DATA TABLE ADDRESS
3394  47DD  5E               7.0          MOV    E,M             ;GET LOW BYTE OF ADDRESS
3395  47DE  23               6.0          INX    H               ;POINT TO THE HIGH BYTE OF ADDRESS
3396  47DF  56               7.0          MOV    D,M             ;GET THE HIGH BYTE OF ADDRESS
3397  47E0  EB               4.0          XCHG
3398  47E1  22  DA  48      16.0          SHLD   EXPADR          ;SAVE THE ADDRESS OF THE EXPECTED DATA TABLE
3399  47E4  EB               4.0          XCHG
3400  47E5  CD  FD  49      18.0          CALL   GENDAT          ;LOAD 1ST SET OF DATA FOR XFR
3401
3402                                ;ISSUE A SYSTEM 'RESTART'
3403
3404  47E8  3E  01           7.0  TSTRST: MVI    A,W.RST         ;GET THE RESTART BIT
3405  47EA  D3  D3          10.0          OUT    WMCCTL
```

```
3406  47EC  3E  05        7.0          MVI   A,SSTEP          ;GET SINGLE-STEP
3407  47EE  D3  F0       10.0          OUT   CLKCTL           ;SELECT IT
3408  47F0  32  D5  48   13.0          STA   CCT_WD           ;SAVE CLOCK CONTROL WORD
3409  47F3  CD  16  4A   18.0          CALL  CLKSYS           ;CLOCK THE SYSTEM TO MAKE THE RESTART
3410  47F6  CD  16  4A   18.0          CALL  CLKSYS           ;TAKE EFFECT
3411  47F9  AF            4.0          XRA   A
3412  47FA  D3  D3       10.0          OUT   WMCCTL           ;FINISH THE RESTART CYCLE
3413
3414                                ;LOAD PROPER BYTE COUNT & PAD COUNT FOR A 4 BYTE TRANSFER
3415
3416  47FC  3E  01        7.0          MVI   A,1
3417  47FE  D3  D4       10.0          OUT   BYTCNT           ;BYTE COUNTER LOADED WITH A 1 TO
3418  4800  AF            4.0          XRA   A                ;INDICATE NO DATA GROUP TO BE ISSUED
3419  4801  D3  D4       10.0          OUT   BYTCNT           ;JUST A RESIDUAL DATA GROUP
3420  4803  3E  03        7.0          MVI   A,3              ;PAD COUNTER FOR 4 BYTE DATA XFR = 3
3421  4805  D3  D5       10.0          OUT   PADCNT           ;2 PADS PLUS AN EXTRA COUNT TO SATISIFY
3422  4807  AF            4.0          XRA   A                ;THE INTEL COUNTER CHIP
3423  4808  D3  D5       10.0          OUT   PADCNT
3424                                ;INIT THE 'DDR' CONTROL TO 'OUT' SO DATA CAN BE INJECTED INTO THE WMC
3425
3426  480A  3E  88        7.0          MVI   A,@210
3427  480C  D3  DB       10.0          OUT   DDRCTL           ;SET DDR TO 'OUT'
3428
3429                                ;SELECT DATA FORMAT FOR THE XFR - 'IMAGE' MODE
3430
3431  480E  3E  D0        7.0          MVI   A,$SC!BIT7       ;IMAGE MODE - 1 BYTE EVERY 'SCLK' TIME
3432  4810  D3  D0       10.0          OUT   DATACTL
3433
3434                                ;SETUP THE RESIDUAL GROUP CONTROL WORD
3435
3436  4812  3E  83        7.0          MVI   A,@203           ;SET TO INDICATE 4 BYTES OF DATA IN RESID
3437  4814  D3  D1       10.0          OUT   RESCHR           ;DATA GROUP
3438
3439                                ;ENABLE ALL THE DATA TRACKS
3440
3441  4816  AF            4.0          XRA   A
3442  4817  D3  D2       10.0          OUT   TRKENA           ;PRESET ALL TRACKS TO 0 SO THE
3443  4819  D3  D2       10.0          OUT   TRKENA           ;PREAMBLE WILL HAVE A PREDICTED RESULT
3444  481B  3E  FF        7.0          MVI   A,@377
3445  481D  D3  D2       10.0          OUT   TRKENA           ;ENABLE TRACKS 7-0
3446  481F  D3  D2       10.0          OUT   TRKENA           ;ENABLE 'P' TRACK
3447
3448                                ;CLOCK THROUGH THE WMC RESTART AREA
3449
3450  4821  3E  B4        7.0          MVI   A,180            ;SETUP A LOOP COUNT
3451  4823  CD  16  4A   18.0  CLKRS:  CALL  CLKSYS           ;CLOCK THE SYSTEM
3452  4826  05            4.0          DCR   B                ;DECREMENT THE LOOP COUNTER
3453  4827  C2  23  48   10.0          JNZ   CLKRS            ;CLOCK TILL COUNTER =0
3454
3455                                ;START THE DATA TRANSFER - GCR  ATA WRITE XFR
3456
3457  482A  3E  D8        7.0          MVI   A,W.ENAB!X.ENAB.W.GCR!W.WRITE
3458  482C  D3  D3       10.0          OUT   WMCCTL           ;START THE XFR NOW
3459  482E  C9           10.0          RET                    ;EXIT THE 'FINDIT' SUBROUTINE
```

```
3461                                    .SBTTL  XFRSUB - SUBROUTINE TO CLOCK DATA THRU TRANSLATOR & CHECK IT
3462                                    ;'WATCHDOG' LOOP
3463
3464  482F  FA  5B  48   10.0  XFRSUB: JM      XMCDAT          ;JUMP IF HERE BEFORE (CALLED FROM NXTBYT)
3465  4832  06  FF          7.0  CLK1:   MVI     B,@377
3466
3467  4834  CD  B9  49   18.0  CLK:    CALL    CKSCLK          ;CLOCK SYSTEM & CHECK FOR SCLK
3468  4837  DA  56  48   10.0          JC      FEXIT           ;JUMP IF ERROR DETECTED
3469  483A  DB  D0       10.0          IN      WMCSTA          ;GET THE WMC STATUS
3470  483C  32  E2  48   13.0          STA     SWMCSTA         ;SAVE THIS STATUS
3471  483F  E6  01        7.0          ANI     X.WCLK          ;TRANSLATOR WRITE CLOCK SET?
3472  4841  C2  5B  48   10.0          JNZ     XMCDAT          ;YES - JUMP IF READY
3473  4844  05           4.0          DCR     B               ;NO - DECREMENT THE LOOP COUNTER
3474  4845  C2  34  48   10.0          JNZ     CLK             ;JUMP BACK IF LOOP COUNTER NOT -0
3475  4848                               ERR     TSTEND,CLKER0
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4848  CD  09  28   18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0012                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  484B  12                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  484C  00                                     .BYTE
 (1)  484D  CD  15  28   18.0                CLKER0::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4850  DA  18  28   10.0                       JC      TSTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
3476                                    ;>WRITE PATH FAILED TO SET TRANSLATOR WRITE CLOCK (WCLK)
3477  4853  C3  18  28   1C.0          JMP     TSTEND
3478
3479  4856  33           6.0  FEXIT:  INX     SP              ;CORRECT THE STACK SO FORCE EXIT WILL
3480  4857  33           6.0          INX     SP              ;WORK CORRECTLY
3481  4858  C3  E3  48   10.0          JMP     LPCADR          ;TAKE THE LOOP-ON-ERROR EXIT
3482                                    ;HERE TO CHECK THE OCCURANCE OF XMC 'WCLK'
3483
3484  485B  CD  B9  49   18.0  XMCDAT: CALL    CKSCLK          ;CLOCK THE SYSTEM TWO MORE TIMES
3485  485E  CD  B9  49   18.0          CALL    CKSCLK          ;TO 'LOCK' THE DATA IN THE PORT BOARD
3486  4861  DB  40       10.0          IN      TSTS            ;GET THE ACTUAL DATA FROM PORT BOARD
3487  4863  47           4.0          MOV     B,A             ;TEMP SAVE IT
3488  4864                               ROUT    ADATA           ;SAVE IN 'CAS' FOR ERROR REPORTING
 (1)  4864  D3  94       10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4866  7F           4.0                    MOV     A,A             ;RETRY LINK
3489
3490                                    ;SEE IF THIS TRANSITION IS TO BE CHECKED - MUST BE PAST THE GCR
3491                                    ;PREAMBLE AND THE MARK 1 BURST & END MARK BURST BUT NOT INTO THE
3492                                    ;2ND SUBGROUP OF DATA
3493
3494  4867  3A  D3  48   13.0          LDA     TRANSCT         ;GET THE CURRENT TRANSITION COUNT
3495  486A  3C           4.0          INR     A               ;COUNT THIS OCCURANCE OF 'WCLK'
3496  486B  32  D3  48   13.0          STA     TRANSCT         ;SAVE THE UPDATED COUNT
3497  486E                               ROUT    R05L            ;SAVE THE COUNT IN CAS FOR ERROR REPORT
 (1)  486E  D3  8A       10.0                   OUT     R05L            ;WRITE AC INTO R05L
 (1)  4870  7F           4.0                    MOV     A,A             ;RETRY LINK
3498  4871  FE  5B        7.0          CPI     @133            ;INTO THE DATA SUBGROUP YET?
3499  4873  DA  32  48   10.0          JC      CLK1            ;NO - KEEP CLOCKING THE SYSTEM
3500
3501                                    ;CHECK THE DATA FROM THE 'EXPECTED' DATA TABLE
3502
3503  4876  2A  DA  48   16.0          LHLD    EXPADR          ;GET THE ADDRESS OF THE DATA TABLE
```

```
3504  4879  7E              7.0          MOV     A,M             ;GET THE EXPECTED DATA BYTE
3505  487A                               ROUT    EDATA           ;SAVE IT IN THE 'CAS' FOR ERROR REPORTING
 (1)  487A  D3  95           10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  487C  7F               4.0                  MOV     A,A             ;RETRY LINK
3506  487D  B8               4.0          CMP     B               ;ACTUAL = EXPECTED DATA??
3507  487E  C8              12.0          RZ                      ;TAKE THE 'OK' EXIT IF NO COMPARE ERROR
3508  487F  3A  D3  48       13.0          LDA     TRANSCT         ;ERROR DETECTED - GET THE TRANSITION COUNT
3509  4882  32  D4  48       13.0          STA     EXDLPC          ;SAVE IT FOR THE ERROR LOOPING CONTROL
3510  4885  3E  FF            7.0          MVI     A,$FF           ;SET A 'FLAG' FOR ERROR EXIT
3511  4887  A7               4.0          ANA     A
3512  4888  C9              10.0          RET                     ;ERROR EXIT
3513                                 ;HERE IS THE ERROR LOOP CONTROL
3514
3515  4889  3A  94  4F       13.0  XDLOOP: LDA     LPFLG           ;GET LOOP-ON-ERROR FLAG
3516  488C  A7               4.0          ANA     A
3517  488D  C2  96  48       10.0          JNZ     XDLP1           ;JUMP IF LOOP FLAG IS SET
3518  4890  32  D4  48       13.0          STA     EXDLPC          ;NO - CLEAR THE LOOP CONTROL
3519  4893  C3  E6  48       10.0          JMP     ERCADR          ;TAKE THE PROCEED EXIT
3520
3521  4896  3A  D3  48       13.0  XDLP1:  LDA     TRANSCT         ;GET THE CURRENT TRNASITION COUNT
3522  4899  47               4.0          MOV     B,A             ;TEMP SAVE
3523  489A  3A  D4  48       13.0          LDA     EXDLPC          ;GET THE LOOP CONTROL COUNT
3524  489D  B8               4.0          CMP     B               ;LOOP AT THIS TRANSITION COUNT?
3525  489E  CA  E3  48       10.0          JZ      LPCADR          ;TAKE THE ERROR LOOP ADDRESS EXIT
3526  48A1  C3  E6  48       10.0          JMP     ERCADR          ;NO MATCH...TAKE THE PROCEED EXIT
3527
3528                                 ;UPDATE THE DATA POINTER FOR THE NEXT 'WCLK' SIGNAL
3529
3530  48A4  2A  DA  48       16.0  NXTBYT: LHLD    EXPADR          ;GET THE EXPECTED DATA TABLE ADDRESS
3531  48A7  23               6.0          INX     H               ;UPDATE THE POINTER
3532  48A8  22  DA  48       16.0          SHLD    EXPADR          ;SAVE THE POINTER FOR NEXT PASS
3533  48AB  3A  D3  48       13.0          LDA     TRANSCT         ;GET THE CURRENT TRANSITION COUNT
3534  48AE  FE  5F            7.0          CPI     @137            ;IS THIS THE LAST BYTE IN  SUBGROUP?
3535  48B0  C8              12.0          RZ                      ;EXIT - 'Z' SET IF ALL DONE
3536
3537                                 ;CLOCK THE SYSTEM 14(8) MORE TIMES TO GET NEXT BYTE
3538
3539  48B1  06  0E            7.0  CLKNXT: MVI     B,14            ;SETUP THE LOOP COUNT
3540  48B3  CD  B9  49       18.0  NXTB1:  CALL    CKSCLK          ;CLOCK THE SYSTEM
3541  48B6  05               4.0          DCR     B               ;DECREMENT THE LOOP COUNT
3542  48B7  C2  B3  48       10.0          JNZ     NXTB1           ;CLOCK TILL THE COUNT =0
3543
3544                                 ;CHECK TO SEE IF 'WCLK' IS SET AFTER 14 CLOCKS - NEXT BYTE READY
3545
3546  48BA  DB  D0            10.0          IN      WMCSTA          ;GET THE WMC STATUS
3547  48BC  32  E2  48       13.0          STA     SWMCSTA         ;SAVE THIS STATUS
3548  48BF  E6  01            7.0          ANI     X.WCLK          ;IS THE WCLK BIT SET?
3549  48C1  C2  C9  48       10.0          JNZ     NOCLKC          ;YUP - OK TO PROCEED
```

```
3551  48C4                                    ERR     CLK1,NOCLKC,4
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  48C4  CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0013                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48C7  13                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48C8  04                                        .BYTE   4
 (1)  48C9  CD  15  28        18.0              NOCLKC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  48CC  DA  32  48        10.0                    JC      CLK1            ;LOOP ADDRESS IF LOOP SPECIFIED
3552                                    ;>TRANSLATOR WRITE CLOCK NOT SET AFTER 8 CLOCK CYCLES
3553                                    ;>IMPROPER TRANSITION RATE W _L OCCUR ON TAPE
3554                                    ;*XMC5 MICRO ERROR 23
3555                                    ;*TRANSLATOR WRITE CLOCK NOT SET AFTER 8 CLOCK CYCLES
3556                                    ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
3557                                    ;*TRANSITION COUNT = LLL
3558  48CF  3E  FF              7.0           MVI     A,$FF           ;SET A FLAG
3559  48D1  A7                  4.0           ANA     A               ;AND THE CONDITION BITS
3560  48D2  C9                 10.0           RET                     ;EXIT
3561                                  ;HERE IS THE VARIABLE DATA STORAGE AREA AND THE FLAG STORAGE AREA
3562
3563  48D3  00                              TRANSCT:        .BYTE   0       ;TRANSITION COUNTER
3564  48D4  00                              EXDLPC:         .BYTE   0       ;TRANSITION COUNT AT DETECTED ERROR
3565  48D5  00                              CCTLWD:         .BYTE   0       ;CLOCK CONTROL WORD - COPY
3566  48D6  00                              XFRNUM:         .BYTE   0       ;4 X 5 SUBGROUP NUMBER
3567  48D7  00                              UNITMP:         .BYTE   0       ;SELECTED PORT # FROM 'HOST' CPU
3568  48D8  00  00                          SUPADR:         .ADDR   0       ;ADDRESS OF THE 'SUPPLY' DATA TABLE
3569  48DA  00  00                          EXPADR:         .ADDR   0       ;ADDRESS OF THE 'EXPECTED' DATA TABLE
3570
3571  48DC  00                              SDDRAT:         .BYTE   0       ;DATA TO DDR REG A
3572  48DD  00                              SDDRA:          .BYTE   0       ;SHIFTED DDR REG A DATA (BITS 5-0,17-16)
3573  48DE  C0                              SDDRB:          .BYTE   0       ;   "       "   "  B   " (BITS 13-6)
3574
3575  48DF  00                              CLKCNT:         .BYTE   0       ;COUNT OF 'SCLK' OCCURANCES
3576  48E0  00                              CLKON:          .BYTE   0       ;SOFTWARE FLAG FOR 'SCLK' ON
3577  48E1  00                              DPNUM:          .BYTE   0       ;DATA BYTE INDEX NUMBER FOR SUPPLY DATA
3578  48E2  00                              SWMCSTA:        .BYTE   0       ;SOFTWARE COPY OF WMCSTA WORD
3579
3580                                  ;STORAGE FOR THE ERROR LOOP CONTROL
3581
3582  48E3  C3  18  28        10.0  LPCADR:         JMP     TSTEND  ;FAKE FATAL EXIT - NORMALLY SET BY TEST
3583  48E6  C3  18  28        10.0  ERCADR:         JMP     TSTEND
```

```
3585                                    ;POINTER TABLE FOR 'SUPPLY' AND 'EXPECTED' DATA TABLE ADDRESSES
3586
3587  48E9  29  49                     XFRTBL: .ADDR   SUP00   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 00
3588  48EB  69  49                             .ADDR   EXP00   ;EXPECTED DATA TABLE FOR SUBGROUP 00
3589  48ED  2D  49                             .ADDR   SUP01   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 01
3590  48EF  6E  49                             .ADDR   EXP01   ;EXPECTED DATA TABLE FOR SUBGROUP 01
3591  48F1  31  49                             .ADDR   SUP02   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 02
3592  48F3  73  49                             .ADDR   EXP02   ;EXPECTED DATA TABLE FOR SUBGROUP 02
3593  48F5  35  49                             .ADDR   SUP03   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 03
3594  48F7  78  49                             .ADDR   EXP03   ;EXPECTED DATA TABLE FOR SUBGROUP 03
3595  48F9  39  49                             .ADDR   SUP04   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 04
3596  48FB  7D  49                             .ADDR   EXP04   ;EXPECTED DATA TABLE FOR SUBGROUP 04
3597  48FD  3D  49                             .ADDR   SUP05   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 05
3598  48FF  82  49                             .ADDR   EXP05   ;EXPECTED DATA TABLE FOR SUBGROUP 05
3599  4901  41  49                             .ADDR   SUP06   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 06
3600  4903  87  49                             .ADDR   EXP06   ;EXPECTED DATA TABLE FOR SUBGROUP 06
3601  4905  45  49                             .ADDR   SUP07   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 07
3602  4907  8C  49                             .ADDR   EXP07   ;EXPECTED DATA TABLE FOR SUBGROUP 07
3603  4909  49  49                             .ADDR   SUP10   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 10
3604  490B  91  49                             .ADDR   EXP10   ;EXPECTED DATA TABLE FOR SUBGROUP 10
3605  490D  4D  49                             .ADDR   SUP11   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 11
3606  490F  96  49                             .ADDR   EXP11   ;EXPECTED DATA TABLE FOR SUBGROUP 11
3607  4911  51  49                             .ADDR   SUP12   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 12
3608  4913  9B  49                             .ADDR   EXP12   ;EXPECTED DATA TABLE FOR SUBGROUP 12
3609  4915  55  49                             .ADDR   SUP13   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 13
3610  4917  A0  49                             .ADDR   EXP13   ;EXPECTED DATA TABLE FOR SUBGROUP 13
3611  4919  59  49                             .ADDR   SUP14   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 14
3612  491B  A5  49                             .ADDR   EXP14   ;EXPECTED DATA TABLE FOR SUBGROUP 14
3613  491D  5D  49                             .ADDR   SUP15   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 15
3614  491F  AA  49                             .ADDR   EXP15   ;EXPECTED DATA TABLE FOR SUBGROUP 15
3615  4921  61  49                             .ADDR   SUP16   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 16
3616  4923  AF  49                             .ADDR   EXP16   ;EXPECTED DATA TABLE FOR SUBGROUP 16
3617  4925  65  49                             .ADDR   SUP17   ;SUPPLY DATA TABLE FOR SUBGROUP TEST 17
3618  4927  B4  49                             .ADDR   EXP17   ;EXPECTED DATA TABLE FOR SUBGROUP 17
3619                                    ;HERE IS THE 'SUPPLY' DATA TABLES FOR DATA INJECTION INTO THE WMC
3620                                    ;WHEN 'SCLK' OCCURES
3621
3622                                    ;SUBGROUP 00 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3623
3624  4929  00                        SUP00:  .BYTE   0
3625  492A  00                                .BYTE   0
3626  492B  00                                .BYTE   0
3627  492C  00                                .BYTE   0
3628
3629                                    ;SUBGROUP 01 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3630
3631  492D  00                        SUP01:  .BYTE   0
3632  492E  00                                .BYTE   0
3633  492F  00                                .BYTE   0
3634  4930  FF                                .BYTE   $FF
3635
3636                                    ;SUBGROUP 02 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3637
3638  4931  00                        SUP02:  .BYTE   0
```

```
3639  4932  00                        .BYTE   0
3640  4933  FF                        .BYTE   $FF
3641  4934  00                        .BYTE   0
3642
3643                                  ;SUBGROUP 03 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3644
3645  4935  00                        SUP03:  .BYTE   0
3646  4936  00                                .BYTE   0
3647  4937  FF                                .BYTE   $FF
3648  4938  FF                                .BYTE   $FF
3649
3650                                  ;SUBGROUP 04 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3651
3652  4939  00                        SUP04:  .BYTE   0
3653  493A  FF                                .BYTE   $FF
3654  493B  00                                .BYTE   0
3655  493C  00                                .BYTE   0
3656
3657                                  ;SUBGROUP 05 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3658
3659  493D  00                        SUP05:  .BYTE   0
3660  493E  FF                                .BYTE   $FF
3661  493F  00                                .BYTE   0
3662  4940  FF                                .BYTE   $FF
3663                                  ·SUBGROUP 06 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3664
3665  4941  00                        SUP06:  .BYTE   0
3666  4942  FF                                .BYTE   $FF
3667  4943  FF                                .BYTE   $FF
3668  4944  00                                .BYTE   0
3669
3670                                  ;SUBGROUP 07 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3671
3672  4945  00                        SUP07:  .BYTE   0
3673  4946  FF                                .BYTE   $FF
3674  4947  FF                                .BYTE   $FF
3675  4948  FF                                .BYTE   $FF
3676
3677                                  ;SUBGROUP 10 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3678
3679  4949  FF                        SUP10:  .BYTE   $FF
3680  494A  00                                .BYTE   0
3681  494B  00                                .BYTE   0
3682  494C  00                                .BYTE   0
3683
3684                                  ;SUBGROUP 11 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3685
3686  494D  FF                        SUP11:  .BYTE   $FF
3687  494E  00                                .BYTE   0
3688  494F  00                                .BYTE   0
3689  4950  FF                                .BYTE   $FF
3690
3691                                  ;SUBGROUP 12 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3692
```

```
3693  4951  FF                  SUP12:  .BYTE   $FF
3694  4952  00                          .BYTE   0
3695  4953  FF                          .BYTE   $FF
3696  4954  00                          .BYTE   0
3697
3698                            ;SUBGROUP 13 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3699
3700  4955  FF                  SUP13:  .BYTE   $FF
3701  4956  00                          .BYTE   0
3702  4957  FF                          .BYTE   $FF
3703  4958  FF                          .BYTE   $FF
3704
3705                            ;SUBGROUP 14 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3706
3707  4959  FF                  SUP14:  .BYTE   $FF
3708  495A  FF                          .BYTE   $FF
3709  495B  00                          .BYTE   0
3710  495C  00                          .BYTE   0
3711                            ;SUBGROUP 15 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3712
3713  495D  FF                  SUP15:  .BYTE   $FF
3714  495E  FF                          .BYTE   $FF
3715  495F  00                          .BYTE   0
3716  4960  FF                          .BYTE   $FF
3717
3718                            ;SUBGROUP 16 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3719
3720  4961  FF                  SUP16:  .BYTE   $FF
3721  4962  FF                          .BYTE   $FF
3722  4963  FF                          .BYTE   $FF
3723  4964  00                          .BYTE   0
3724
3725                            ;SUBGROUP 17 DATA FOR INJECTION TO WMC VIA DDR REGISTERS
3726
3727  4965  FF                  SUP17:  .BYTE   $FF
3728  4966  FF                          .BYTE   $FF
3729  4967  FF                          .BYTE   $FF
3730  4968  FF                          .BYTE   $FF
3731
3732  4969                      S
 (1)                          ; ************************************************************************
3733                            ;HERE IS THE EXPECTED DATA TABLES FOR THE 4 X 5 SUBGROUP TRANSLATIONS
3734                            ;ALL EXPECTED DATA STARTS AT TRANSITION COUNT 133(8) AND ENDS AT 137(8)
3735  4969                      S
 (1)                          ; ************************************************************************
3736
3737                            ;EXPECTED SUBGROUP 00 DATA (11001 FOR ALL BITS)
3738
3739  4969  00                  EXP00:  .BYTE   0
3740  496A  FF                          .BYTE   $FF
3741  496B  FF                          .BYTE   $FF
3742  496C  FF                          .BYTE   $FF
3743  496D  00                          .BYTE   0
3744
```

```
3745                                ;EXPECTED SUBGROUP 01 DATA (11011 FOR ALL BITS)
3746
3747  496E  00                     EXP01:  .BYTE   0
3748  496F  FF                             .BYTE   $FF
3749  4970  FF                             .BYTE   $FF
3750  4971  00                             .BYTE   0
3751  4972  FF                             .BYTE   $FF
3752
3753                                ;EXPECTED SUBGROUP 02 DATA (10010 FOR ALL BITS)
3754
3755  4973  00                     EXP02:  .BYTE   0
3756  4974  00                             .BYTE   0
3757  4975  00                             .BYTE   0
3758  4976  FF                             .BYTE   $FF
3759  4977  FF                             .BYTE   $FF
3760                                ;EXPECTED SUBGROUP 03 DATA (10011 FOR ALL BITS)
3761
3762  4978  00                     EXP03:  .BYTE   0
3763  4979  00                             .BYTE   0
3764  497A  00                             .BYTE   0
3765  497B  FF                             .BYTE   $FF
3766  497C  00                             .BYTE   0
3767
3768                                ;EXPECTED SUBGROUP 04 DATA (11101 FOR ALL BITS)
3769
3770  497D  00                     EXP04:  .BYTE   0
3771  497E  FF                             .BYTE   $FF
3772  497F  00                             .BYTE   0
3773  4980  00                             .BYTE   0
3774  4981  FF                             .BYTE   $FF
3775
3776                                ;EXPECTED SUBGROUP 05 DATA (10101 FOR ALL BITS)
3777
3778  4982  00                     EXP05:  .BYTE   0
3779  4983  00                             .BYTE   0
3780  4984  FF                             .BYTE   $FF
3781  4985  FF                             .BYTE   $FF
3782  4986  00                             .BYTE   0
3783
3784                                ;EXPECTED SUBGROUP 06 DATA (10110 FOR ALL BITS)
3785
3786  4987  00                     EXP06:  .BYTE   0
3787  4988  00                             .BYTE   0
3788  4989  FF                             .BYTE   $FF
3789  498A  00                             .BYTE   0
3790  498B  00                             .BYTE   0
3791
3792                                ;EXPECTED SUBGROUP 07 DATA (10111 FOR ALL BITS)
3793
3794  498C  00                     EXP07:  .BYTE   0
3795  498D  00                             .BYTE   0
3796  498E  FF                             .BYTE   $FF
3797  498F  00                             .BYTE   0
3798  4990  FF                             .BYTE   $FF
```

```
3799
3800                                    ;EXPECTED SUBGROUP 10 DATA (11010 FOR ALL BITS)
3801
3802  4991  00                          EXP10:  .BYTE   0
3803  4992  FF                                  .BYTE   $FF
3804  4993  FF                                  .BYTE   $FF
3805  4994  00                                  .BYTE   0
3806  4995  00                                  .BYTE   0
3807                                    ;EXPECTED SUBGROUP 11 DATA (01001 FOR ALL BITS)
3808
3809  4996  FF                          EXP11:  .BYTE   $FF
3810  4997  00                                  .BYTE   0
3811  4998  00                                  .BYTE   0
3812  4999  00                                  .BYTE   0
3813  499A  FF                                  .BYTE   $FF
3814
3815                                    ;EXPECTED SUBGROUP 12 DATA (01010 FOR ALL BITS)
3816
3817  499B  FF                          EXP12:  .BYTE   $FF
3818  499C  00                                  .BYTE   0
3819  499D  00                                  .BYTE   0
3820  499E  FF                                  .BYTE   $FF
3821  499F  FF                                  .BYTE   $FF
3822
3823                                    ·EXPECTED SUBGROUP 13 DATA (01011 FOR ALL BITS)
3824
3825  49A0  FF                          EXP13:  .BYTE   $FF
3826  49A1  00                                  .BYTE   0
3827  49A2  00                                  .BYTE   0
3828  49A3  FF                                  .BYTE   $FF
3829  49A4  00                                  .BYTE   0
3830
3831                                    ;EXPECTED SUBGROUP 14 DATA (11110 FOR ALL BITS)
3832
3833  49A5  00                          EXP14:  .BYTE   0
3834  49A6  FF                                  .BYTE   $FF
3835  49A7  00                                  .BYTE   0
3836  49A8  FF                                  .BYTE   $FF
3837  49A9  FF                                  .BYTE   $FF
3838
3839                                    ;EXPECTED SUBGROUP 15 DATA (01101 FOR ALL BITS)
3840
3841  49AA  FF                          EXP15:  .BYTE   $FF
3842  49AB  00                                  .BYTE   0
3843  49AC  FF                                  .BYTE   $FF
3844  49AD  FF                                  .BYTE   $FF
3845  49AE  00                                  .BYTE   0
3846                                    ;EXPECTED SUBGROUP 16 DATA (01110 FOR ALL BITS)
3847
3848  49AF  FF                          EXP16:  .BYTE   $FF
3849  49B0  00                                  .BYTE   0
3850  49B1  FF                                  .BYTE   $FF
3851  49B2  00                                  .BYTE   0
3852  49B3  00                                  .BYTE   0
```

```
                                        ;EXPECTED SUBGROUP 17 DATA (01111 FOR ALL BITS)

49B4  FF                        EXP17:  .BYTE   $FF
49B5  00                                .BYTE   0
49B6  FF                                .BYTE   $FF
49B7  00                                .BYTE   0
49B8  FF                                .BYTE   $FF
49B9                            $
 (1)                            ; ******************************************************************************
                                ;CKSCLK -- ROUTINE TO CLOCK THE SYSTEM ONCE AND THEN CHECK FOR PRESENCE
                                ;       OF 'SCLK' FROM THE WMC. IF 'SCLK' IS ON, THEN LOAD NEW DATA INTO
                                ;       THE DDR.
                                ;
49B9                            $
 (1)                            ; ******************************************************************************

49B9  CD  16  4A      18.0      CKSCLK: CALL    CLKSYS          ;CLOCK SYSTEM ONCE
49BC  DB  C0          10.0              IN      DBUSSTA         ;GET STATUS OF DATA BUS
49BE  E6  02           7.0              ANI     T.SCLK          ;SAVE ONLY THE 'SCLK' BIT
49C0  C2  C8  49      10.0              JNZ     SCLKBIT         ;JUMP IF SCLK IS ON
49C3  AF               4.0              XRA     A               ;NO SCLK - CLEAR CLOCK ON FLAG
49C4  32  E0  48      13.0              STA     CLKON
49C7  C9              10.0              RET                     ;EXIT

49C8                            $
 (1)                            ; ******************************************************************************
                                ;SCLKBIT -- HERE WHEN A CLOCK CYCLE RESULTS IN 'SCLK' SETTING IN DBUS
                                ;       CONTROL WORD.  LOAD NEXT DATA SET (18 BITS) INTO DDR REGS.
49C8                            $
 (1)                            ; ******************************************************************************

49C8  3A  E0  48      13.0      SCLKBIT: LDA    CLKON           ;SEE IF 'SCLK' WAS ON
49CB  A7               4.0              ANA     A               ;SET CONDITION BITS
49CC  C0              12.0              RNZ                     ;EXIT IF CLOCK WAS SET - HAVE DATA ALREADY

49CD  CD  E1  49      18.0              CALL    SETDDR          ;LOAD THE DATA INTO THE DDR REGS
49D0  CD  FD  49      18.0              CALL    GENDAT          ;AND GENERATE NEW DATA FOR NEXT DDR LOAD

49D3  06  7F           7.0      SCKB1:  MVI     B,@177          ;GET A NEW WATCHDOG COUNT
49D5  78               4.0              MOV     A,B             ;COPY IT TO REG A
49D6  32  E0  48      13.0              STA     CLKON           ;SET THE SCLK ON FLAG
49D9  3A  DF  48      13.0              LDA     CLKCNT          ;GET THE CURRENT SCLK COUNT
49DC  3C               4.0              INR     A               ;UPDATE IT
49DD  32  DF  48      13.0              STA     CLKCNT          ;SAVE THE NEW COUNT
49E0  C9              10.0              RET                     ;EXIT
49E1                            $
 (1)                            ; ******************************************************************************
                                ;SETDDR -- ROUTINE TO TRANSLATE THE DESIRED DATA FROM A 1-FOR-1 BIT
                                ;       POSITION TO THE DDR TO BE USED AS DATA TO THE TM78.
                                ;
                                ;       THE DATA HAS TO BE SHIFTED TO COMPENSATE FOR THE 'LEFT' FLAG
                                ;       CONTROL IN THE DDR REGISTERS. THE PACKING MODE USED IS 'IMAGE'.
49E1                            $
 (1)                            ; ******************************************************************************
```

```
3901
3902                                    ;HERE TO TRANSLATE THE DATA INTO AN ACCEPTABLE FORMAT FOR DDR INJECTION
3903
3904  49E1  3A DC 48      13.0  SETDDR: LDA     SDDRAT          ;GET DDRA TEMP
3905  49E4  E6 3F          7.0          ANI     @77             ;STRIP 2 BITS OFF
3906  49E6  07              4.0          RLC
3907  49E7  07              4.0          RLC                     ;LEFT JUSTIFY
3908  49E8  32 DD 48      13.0          STA     SDDRA           ;SAVE DDRA DESIRED
3909  49EB  D3 D8         10.0          OUT     DDRA            ;LOAD THE REAL DDRA
3910
3911  49ED  3A DC 48      13.0          LDA     SDDRAT          ;GET DDRA AGAIN
3912  49F0  E6 C0          7.0          ANI     @300            ;SAVE 2 BITS
3913  49F2  07              4.0          RLC
3914  49F3  07              4.0          RLC
3915  49F4  32 DE 48      13.0          STA     SDDRB           ;SAVE DDRB DATA
3916  49F7  D3 D9         10.0          OUT     DDRB            ;LOAD THE REAL DDRB
3917
3918  49F9  AF              4.0          XRA     A
3919  49FA  D3 DA         10.0          OUT     DDRC
3920  49FC  C9            10.0          RET                     ;EXIT
3921  49FD                        S
 (1)                        ; **********************************************************************
3922                         ;GENDAT -- ROUTINE TO GENERATE DDR REG DATA FOR NEXT "SCLK" CYCLE.
3923                         ;       DATA IS LOADED IN DDR REGS AND A COPY IS STORED IN 'SDDRAT'
3924                         ;       THRU 'SDDRCT' FOR REFERENCE.
3925  49FD                        S
 (1)                        ; **********************************************************************
3926
3927  49FD  F5            12.0  GENDAT: PUSH    PSW             ;SAVE PSW + A
3928  49FE  C5            12.0          PUSH    B               ;SAVE B + C
3929  49FF  E5            12.0          PUSH    H               ;SAVE H + L
3930  4A00  3A E1 48      13.0          LDA     DPNUM           ;GET THE LAST PATTERN # USED
3931  4A03  3C              4.0          INR     A               ;POINT TO NEXT
3932  4A04  32 E1 48      13.0          STA     DPNUM           ;SAVE NUMBER
3933  4A07  4F              4.0          MOV     C,A
3934  4A08  06 00           7.0          MVI     B,0             ;B + C HAVE OFFSET
3935  4A0A  2A D8 48      16.0          LHLD    SUPADR          ;GET THE BASE ADDRESS OF THE SUPPLY DATA
3936  4A0D  09            10.0          DAD     B               ;ADD IN OFFSET TO H + L
3937
3938  4A0E  7E              7.0          MOV     A,M             ;GET DATA BYTE FOR DDR 'A' (BITS 7-0)
3939  4A0F  32 DC 48      13.0          STA     SDDRAT          ;SAVE DDR 'A' IMAGE
3940
3941  4A12  E1            10.0  GENOUT: POP     H               ;RESTORE REGS
3942  4A13  C1            10.0          POP     B
3943  4A14  F1            10.0          POP     PSW
3944  4A15  C9            10.0          RET
3945
3946  4A16                        S
 (1)                        ; **********************************************************************
3947                         ;CLKSYS -- ROUTINE TO CLOCK THE SYSTEM 1 TIME BY WRITING CLOCK CONTROL
3948                         ;       'CLK' BIT TO A '1' THEN A '0'.
3949  4A16                        S
 (1)                        ; **********************************************************************
3950
```

```
3951  4A16  3A  D5  48      13.0  CLKSYS: LDA   CCTLWD          ;GET SOFTWARE CLOCK CONTROL IMAGE
3952  4A19  F6  40            7.0          ORI   SSCLK          ;ADD IN 'CLK' BIT
3953  4A1B  D3  F0           10.0          OUT   CLKCTL         ;LOAD CLOCK CONTROL
3954  4A1D  E6  3F            7.0          ANI   @77            ;STRIP OFF CLOCK BIT
3955  4A1F  D3  F0           10.0          OUT   CLKCTL         ;LOAD CLOCK CONTROL WORD
3956  4A21  C9               10.0          RET                  ;EXIT - CLOCK CYCLE COMPLETE
```

```
                                        .SB  L  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ; ***************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ; ***************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ; ***************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ; ***************************************************************************
4A22  F5              12.0  CLEAR:  PUSH   PSW          ;SAVE THE SELECTED PORT #
4A23  C5              12.0          PUSH   B
4A24  06  00           7.0          MVI    B,0         ;START TO CLEAR AT PORT #0
4A26  DB  E0          10.0  CLRLP:  IN     INTSTA      ;GET MB SELECT INFO
4A28  E6  80           7.0          ANI    BIT7        ;SAVE ONLY THE MASSBUS SELECT BIT
4A2A  B0               4.0          ORA    B           ;ADD IN THE SELECTED PORT #
4A2B  D3  E0          10.0          OUT    MBSEL       ;RESET TO THIS PORT
4A2D  3E  80           7.0          MVI    A,@200      ;LOAD MTA REGISTER #0 SELECT CODE
4A2F  D3  40          10.0          OUT    TCMD        ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
4A31  AF               4.0          XRA    A           ;CLEAR TU COMMAND A
4A32  D3  40          10.0          OUT    TCMD        ;
4A34  3E  81           7.0          MVI    A,@201      ;LOAD MTA REGISTER #1 SELECT CODE
4A36  D3  40          10.0          OUT    TCMD        ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
4A38  3E  00           7.0          MVI    A,SELCLR    ;LOAD TU "CLEAR SELECT" COMMAND
4A3A  D3  40          10.0          OUT    TCMD        ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
4A3C  AF               4.0          XRA    A
4A3D  D3  44          10.0          OUT    TAMT        ;CLEAR AMTIE WORD
4A3F  D3  48          10.0          OUT    PDIAG       ;CLEAR DIAG CONTROL WORD
4A41  D3  4C          10.0          OUT    PENAB       ;CLEAR PORT ENABLE WORD
4A43  04               4.0          INR    B           ;POINT TO THE NEXT PORT TO CLEAR
4A44  78               4.0          MOV    A,B
4A45  FE  04           7.0          CPI    4           ;DONE?
4A47  C2  26  4A      10.0          JNZ    CLRLP       ;NO - CLEAR THIS PORT ALSO
4A4A  C1              10.0          POP    B           ;RESET B & C
4A4B  F1              10.0          POP    PSW         ;ALL DONE
4A4C  C9              10.0          RET                ;EXIT
      0000                    .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE    4F9B          ATTCD   4F97          AXNUM   4F91
B      =%0000          BADST  = 0090          BIT0   = 0001          BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008          BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080          BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A          BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00           BSAVE   4F9C          BYTCNT= 00D4          BYTEH   4F24
BYTEL   4F23           C      =%0001          CASCT   4F21          CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1
CBYTH  = 008B          CBYTL  = 008A          CCTLWD  48D5          CDG1H  = 0087
CDG1L  = 0086          CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095
CDG3L  = 0094          CDVTH  = 008D          CDVTL  = 008C          CHPTIE= 0028
CHOTIE= 0020           CH1TIE= 0021           CH2TIE= 0022           CH3TIE= 0023
CH4TIE= 0024           CH5TIE= 0025           CH6TIE= 0026           CH7TIE= 0027
CKLOP  = 2815          CKSCLK  49B9          CLEAR   4A22          CLK     4834
CLKCNT  48DF           CLKCTL= 00F0          CLKERO  484D  G       CLKNXT  48B1
CLKON   48E0           CLKRS   4823          CLKSYS  4A16          CLK1    4832
CLOCK   4F26           CLRLP   4A26          CMC0H  = 0099          CMC0L  = 0098
CMC1H  = 009B          CMC1L  = 009A          CMC2H  = 009D          CMC2L  = 009C
CMC3H  = 009F          CMC3L  = 009E          CMINH  = 0097          CMINL  = 0096
CNTCTL= 00D7           CRCWRD= 0018           CSAVE   4F9D          CSRLH  = 0091
CSRLL  = 0090          CTCH   = 0085          CTCL   = 0084          CTSTH  = 008F
CTSTL  = 008E          CXCTH  = 0081          CXCTL  = 0080          CXINH  = 0083
CXINL  = 0082          C.     = 0001          C.AVAI= 0080          C.DP   = 0008
C.DSE  = 0010          C.DTU  = 0003          C.DVA  = 0008          C.FAIL= 00FC
C.FMT  = 0070          C.FNCT= 003E          C.GO   = 0001          C.INTC= 00FE
C.MAIN= 0020           C.NSA  = 0080          C.RCT  = 00FC          C.SER  = 0080
C.SHR  = 0040          C.SKPC= 000F          C.TAPE= 0040          C.WCS  = 0002
D      =%0002          DATACT= 00D0          DBUS    4F28          DBUSCT= 00C0
DBUSST= 00C0           DDRA   = 00D8          DDRAIN= 0010          DDRB   = 00D9
DDRBIN= 0002           DDRC   = 00DA          DDRCIN= 0001          DDRCO  = 0088
DDRCTL= 00DB           DIAFLG  4F22          DIAGPG= 4300          DIAGRM= 4F90
DIARD  = 000B          DONE1  = 0045          DONINT= 0010          DPNUM   48E1
DSAVE   4F9E           DSE    = 0006          D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010           D.LAGC= 0020           D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008           D.WR4  = 0080          E      =%0003          ECCBAD= 0042
ECCCOR= 0019           ECCOK  = 0041          ECCSTA= 001A          ECCTST= 000E
EDATA  = 0095          EOTCLR= 0003           ERCADR  48E6          ERFLG   4F93
ERLP   = 2809          ERLPA  = 280F          ERLPB  = 2812          ERLPE  = 280C
ERNUM   4F90           ERRCNT= 00D6           ESAVE   4F9F          EXDLPC  48D4
EXPADR  48DA           EXP00   4969           EXP01   496E          EXP02   4973
EXP03   4978           EXP04   497D           EXP05   4982          EXP06   4987
EXP07   498C           EXP10   4991           EXP11   4996          EXP12   499B
EXP13   49A0           EXP14   49A5           EXP15   49AA          EXP16   49AF
EXP17   49B4           E.ACRC= 0010           E.AMT  = 0020          E.CDP  = 0080
E.CRC  = 0080          E.PNTR= 0008           E.RPE  = 0040          E.STEC= 0001
E.TTEC= 0002           E.UNC  = 0004          FEXIT   4856          FIFORD- 006A
FINDIT  4752           FORMAT  4F25          FOUND   4774          FPORT   4794
FWDTST= 0061           GCRID  = 0089          GCRSET= 0002          GENDAT  49FD
GENOUT  4A12           GOODTM= 0092           H      =%0004          HLSAVE  4FA0
IE     = 0008          INTSTA= 00E0           ITERA   4F9A          I.PWR  = 0020
I.RMPE= 0040           I5.5   = 0010          I6.5   = 0020          I7.5   = 0040
KCALL  = 005F          KCLR   = 007B          KDEP   = 003F          KENAB   0078
KEXAM  = 003E          KEYBRD= 00C8           KEY1   = 0078          KEY10   006D
```

```
KEY11 = 006E            KEY12 = 006F            KEY13 = 005C            KEY14 = 005D
KEY15 = 005E            KEY16 = 005F            KEY17 = 003C            KEY18 = 003D
KEY19 = 003E            KEY2  = 0079            KEY20 = 003F            KEY3  = 007A
KEY4  = 007B            KEY5  = 0074            KEY6  = 0075            KEY7  = 0076
KEY8  = 0077            KEY9  = 006C            KINTA = 006F            KLDAD = 003D
KN0   = 003C            KN1   = 005C            KN2   = 005D            KN3   = 005E
KN4   = 006C            KN5   = 006D            KN6   = 006E            KN7   = 0074
KN8   = 0075            KN9   = 0076            KU2   = 0079            KU3   = 007A
KU8   = 0077            L     =%0005            LBLANK= 000F            LCE   = 000B
LCH   = 000C            LCL   = 000D            LCP   = 000E            LC0   = 0000
LC1   = 0001            LC2   = 0002            LC3   = 0003            LC4   = 0004
LC5   = 0005            LC6   = 0006            LC7   = 0007            LC8   = 0008
LC9   = 0009            LDLEDA= 00CA            LDLEDB= 00CB            LDLEDC= 00CC
LDLEDD= 00CD            LDLEDE= 00CE            LDLEDF= 00CF            LKDIAG= 2800
LKKBD = 004C            LKKEY   0049            LKLWMG= 0058            LKLWMP= 0055
LKLWPG= 0052            LKLWPP= 004F            LKMOD7= 0046            LKOPR = 0046
LPCADR  48E3            LPFLG   4F94            LPNUM   4F92            M     =%0006
MBSEL = 00E0            MB.A  = 0008            MB.B  = 0004            MEMTOP= 4FFF
MINUS = 000A            MM    = 8000            MSE   = 0008            MSGN  = 0013
MTACLR= 0000            MT.ARA= 0020            MT.CPE= 0080            MT.DSE= 0001
MT.FWD= 0040            MT.INH= 0008            MT.LWR= 0004            MT.MOT= 0002
MT.NWT= 0080            MT.PEC= 0040            MT.PSB= 0004            MT.PS0= 0001
MT.PS1= 0002            MT.REV= 0020            MT.WRT= 0010            MT.Z  = 0008
M.ATA = 0080            M.CAPE= 0020            M.CONT= 0080            M.DEM = 0020
M.EBL = 0010            M.EXC = 0008            M.FAIL= 0008            M.ILR = 0010
M.INIT= 0010            M.OCC = 0020            M.ONLI= 0001            M.PE  = 0040
M.PORT= 0080            M.RDEN= 0002            M.RDPE= 0008            M.RUN = 0004
M.SCLK= 0001            M.TRA = 0040            M.UNIT= 0007            M.WCLK= 0040
M.WCLN= 0080            M.WREN= 0080            M5.5  = 0001            M6.5  = 0002
M7.5  = 0004            NOCLKC  48C9  G         NOTCAP= 0088            NOUNIT  476B  G
NXTBYT  48A4            NXTB1   48B3            OKAY  = 00FF            OPRRAM= 4300
OPSTRT= 0058            OPVER = 0040            PADCNT= 00D5            PADCRC= 0080
PDIAG = 0048            PEID  = 008A            PENAB = 004C            PESET = 0001
PL    = 00B1            PRDD  = 004C            PRENF = 009C            PS    = 00B2
PSTAT = 0048            PSW   =%0009            P.AMTP= 0001            P.BCTC= 004C
P.CMDP= 0020            P.INTE= 0080            P.LCS = 0040            P.LWR = 0020
P.RDP = 0002            P.RPEN= 0004            P.RPST= 0002            P.RPOE= 0020
P.RP1E= 0010            P.RP2E= 0020            P.RP3E= 0010            P.SING= 0080
P.STAT= 0002            P.STPE= 0080            P.TACH= 0008            P.TUPR= 0010
P.WCSP= 0004            P.WDS = 0040            P.WFLP= 0001            P.WPEN= 0010
P.WPOE= 0008            P.WP1E= 0004            P.WP2E= 0008            P.WP3E= 0004
P.SVOK= 0002            QUE   = 281B            QUEM  = 281E            RAMT  = 0010
RARA  = 0006            RARAI = 0004            RCHBDO= 0048            RCHBD1= 0047
RCHOK = 0046            RCHTST= 000C            RCLRT = 000D            RCMD  = 000B
RCMLP = 0003            RCONT = 0080            RDATA = 0017            RDCLK = 0010
RDON  = 0011            READG = 0007            REND  = 0014            REQST = 2806
RESCHR= 00D1            REVTST= 0064            REWIND= 0004            RFIFOL= 0008
RGCLK = 0002            RGCRI = 0003            RIBG  = 0001            RILL  = 0012
RINST = 000C            RMCTST= 0008            RMK2  = 0013            RNOP  = 0000
RPATH = 0001            RPBAD = 0044            RPCHI = 0001            RPCLK = 0003
RPCTL = 0009            RPEI  = 0002            RPFAIL= 0000            RPF1  = 009D
RPF2  = 009E            RPOK  = 0043            RPOSTN= 0016            RPSTA = 0015
RRCMT = 000A            RSTAT = 0002            RTIEB = 000A            RTIER = 0030
RTM   = 0005            RUNKI = 0009            RUPTST= 005E            RWDUNL= 0005
```

```
R.AMT  = 0001    R.BOP  = 0008    R.DATA= 0040    R.DON  = 0002
R.DRDY= 0010    R.END  = 0010    R.ILL  = 0004    R.JVOK= 0004
R.MK2  = 0008    R.PLOD= 0008    R.PL00= 001C    R.PL01= 0020
R.POST= 0020    R.STNM= 0002    R.STOP= 0004    R.STPC= 0001
R.TBJN= 0080    R.TSTD= 0040    R.VOK  = 0080    R00H   = 0081
R00L   = 0080    R01H   = 0083    R01L   = 0082    R02H   = 0085
R02L   = 0084    R03H   = 0087    R03L   = 0086    R04H   = 0089
R04L   = 0088    R05H   = 008B    R05L   = 008A    R06H   = 008D
R06L   = 008C    R07H   = 008F    R07L   = 008E    R10H   = 0091
R10L   = 0090    R11H   = 0093    R11L   = 0092    R12H   = 0095
R12L   = 0094    R13H   = 0097    R13L   = 0096    R14H   = 0099
R14L   = 0098    R15H   = 009B    R15L   = 009A    R16H   = 009D
R16L   = 009C    R17H   = 009F    R17L   = 009E    R7.5   = 0010
SCKB1    49D3    SCLKBI   49C8    SDDRA    48DD    SDDRAT   48DC
SDDRB    48DE    SELCLR= 0000    SETATA= 00A1    SETDDR   49E1
SID    = 0080    SOD    = 0080    SOE    = 0040    SP     =%0008
SSCLK  = 0040    SSTEP  = 0005    STACK  = 4FFF    STATRM=  4F20
STPCT    4F20    STRSP  = 5000    SUPADR   48D8    SUP00    4929
SUP01    492D    SUP02    4931    SUP03    4935    SUP04    4939
SUP05    493D    SUP06    4941    SUP07    4945    SUP10    4949
SUP11    494D    SUP12    4951    SUP13    4955    SUP14    4959
SUP15    495D    SUP16    4961    SUP17    4965    SWMCST   48E2
TADR00= 0080    TADR01= 0081    TADR02= 0082    TADR03= 0083
TADR04= 0084    TADR05= 0085    TADR06= 0086    TADR07= 0087
TADR10= 0088    TADR11= 0089    TADR12= 008A    TADR13= 008B
TAMT   = 0044    TASEL  = 0080    TCMD   = 0040    TC.FWD= 0040
TC.INH= 0008    TC.LWR= 0004    TC.REV= 0020    TC.WRT= 0010
TEMP     4F99    TEST1    4300    TEST10   44E2    TEST11   4527
TEST12   456C    TEST13   45B1    TEST14   45F6    TEST15   463B
TEST16   4680    TEST17   46C5    TEST2    4344    TEST20   470A
TEST3    4389    TEST4    43CE    TEST5    4413    TEST6    4458
TEST7    449D    TMF    = 0099    TMRDY  = 0040    TRANSC   48D3
TRKENA= 00D2    TSET   = 2803    TSTEND= 2818    TSTRST   47E8
TSTS   = 0040    TUSEL0= 00D1    TUSEL1= 00D2    TU78   = 0010
T.ATH0= 0001    T.ATH1= 0002    T.BOT  = 0004    T.EOT  = 0002
T.FPT  = 0001    T.NTHR= 0004    T.ONL  = 0020    T.PES  = 0008
T.PSBJ= 0020    T.PS0J= 0008    T.PS1J= 0010    T.RDY  = 0080
T.RDY0= 0040    T.RWD  = 0010    T.SCLK= 0002    T1CONT   432A
T1DE0    4324  G  T1LOOP   4319    T1RPT    4305    T10CON   450D
T10DE0   4507  G  T10L00   44FC    T10RPT   44E7    T11CON   4552
T11DE0   454C  G  T11L00   4541    T11RPT   452C    T12CON   4597
T12DE0   4591  G  T12L00   4586    T12RPT   4571    T13CON   45DC
T13DE0   45D6  G  T13L00   45CB    T13RPT   45B6    T14CON   4621
T14DE0   461B  G  T14L00   4610    T14RPT   45FB    T15CON   4666
T15DE0   4660  G  T15L00   4655    T15RPT   4640    T16CON   46AB
T16DE0   46A5  G  T16L00   469A    T16RPT   4685    T17CON   46F0
T17DE0   46EA  G  T17L00   46DF    T17RPT   46CA    T2CONT   436F
T2DE0    4369  G  T2LOOP   435E    T2RPT    4349    T20CON   4735
T20DE0   472F  G  T20L00   4724    T20RPT   470F    T3CONT   43B4
T3DE0    43AE  G  T3LOOP   43A3    T3RPT    438E    T4CONT   43F9
T4DE0    43F3  G  T4LOOP   43E8    T4RPT    43D3    T5CONT   443E
T5DE0    4438  G  T5LOOP   442D    T5RPT    4418    T6CONT   4483
T6DE0    447D  G  T6LOOP   4472    T6RPT    445D    T7CONT   44C8
T7DE0    44C2  G  T7LOOP   44B7    T7RPT    44A2    UIBG   = 00A1
```

```
UNITMP  48D7          VALFC   4F98          VALTB   4F95          VELTST= 005B
WDR.P = 0010          WMCCTL= 00D3          WMCERR= 00DA          WMCSTA= 00D0
WRTCLK= 0000          WRTDAT= 00D3          W.ACRC= 0004          W.CRC = 0008
W.DIAG= 0002          W.DONN= 0040          W.ECC = 0010          W.ENAB= 0080
W.ERR = 0020          W.FMT = 0070          W.GCR = 0010          W.LEFT= 0004
W.ONES= 0020          W.RESI= 0002          W.REV = 0004          W.ROME= 0010
W.RST = 0001          W.SKIP= 000F          W.WRIT= 0008          W.XFER= 0020
X     =%000A          XDLOOP  4889          XDLP1   4896          XFRNUM  48D6
XFRSUB  482F          XFRTBL  48E9          XMCDAT  485B          X.DONN= 0080
X.ENAB- 0040          X.PEPE= 0002          X.ROME= 0001          X.WCLK= 0001
Y     -%000B          .     = 4A4D


 ERRORS DETECTED:  0

 *XMC5.A78/PTP,XMC5=NLIST,PARAM,MACRO,LIST,XMC5
 RUN-TIME: 6 11 0 SECONDS
 CORE USED:  10K
```

     3   'MEMORY MAP'' EXTENDED ADDRESS BIT
    19   READ MICRO CONTROLLER CONTROL & STATUS WORDS
   139   TRANSPORT/PORT COMMAND + STATUS WORDS
   380   COMMON ADDRESS SPACE REGISTERS
   444   MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
   485   MASSBUS CONTROL WORD DEFINITIONS
   557   WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
   709   KEYBOARD/DISPLAY CONTROL + STATUS WORD
   797   MISCELLANEOUS VARIABLE DEFINITIONS
   801   RAM ADDRESS DEFINITIONS
   819   LINKAGE TABLE ADDRESSES
   842   BIT DEFINITIONS
   856   RIM AND SIM INSTRUCTION BIT DEFINITIONS
   877   INTERRUPTABLE INSTRUCTION MACROS
   924   DIAGNOSTIC VARIABLES
   945   COMMON VARIABLE STORAGE
   959   MACRO CALL DEFINITIONS
  1047   TM78 MICRO-DIAGNOSTIC MACRO CALLS
  1101   8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
  1333   TEST 1 - XMC - RESYNC BURST TEST
  1823   SUBROUTINE CLKSYS
  1837   SUBROUTINE CLEAR ALL TU PORTS
  2381   TABLE EXPECTED PARITY
  2483   PROGRAM VARIABLES

```
                ; ************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;                "BYTE/SCLK COUNT NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                MM = DATA FORMAT FROM CAS REGISTER 2
                ;                NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;                BYTE-SCLK COUNT = LLL
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                LLL = AS ABOVE
                ;                MM  = AS ABOVE
                ;                NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;                TRANSITION COUNT = LLL
                ;                WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;                EXPECTED 18 BITS =      E EEEEEE
                ;                ACTUAL 18 BITS =        A AAAAAA
                ;
                ;                WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;                BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;                OF CAS REG 15 LOW BYTE.
                ;
                ;                ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;                AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;                17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;                TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;                AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;                "SUBGROUP NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       10 -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;.......................................................................
```

```
1016                    ;*****************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN FRROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ;*****************************************************************
1035                    ;*****************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ;*****************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  XMC6 - TRANSLATOR MICRO CONTROLLER PART #6
                                .SBTTL  TEST 1 - XMC - RESYNC BURST TEST
                                ;ID     XMC6-TRANSLATOR MICRO CONTROLLER PART #6
 4300                           ST
  (1)                           ; ************************************************************************
  (1)                           ;*TEST TITLE
  (1)                           ;*---- -----
                                ;*XMC - RESYNC BURST TEST
 4300                           SD
  (1)                           ; ************************************************************************
  (1)                           ;*DESCRIPTION
  (1)                           ;*-----------
                                ;THIS TEST COMMANDS THE WRITE PATH TO WRITE A GCR RECORD OF 1113(10) BYTES
                                ;AND THEN SINGLE STEPS THE WRITE PATH (WMC, XMC) WHILE MONITORING THE
                                ;XMC "XL WCLK" BIT.  THE OUTPUT OF THE TRANSLATOR SHOULD INDICATE A PRE-
                                ;AMBLE, 158 DATA GROUPS, RESYNC BURST, DATA GROUP, END MARK, RESIDUAL GROUP,
                                ;CRC GROUP, AND POSTAMBLE.  THIS OUTPUT IS CHECKED FOR SEQUENCE, CONTENT,
                                ;*DATA STATES AND FLUX REVERSAL RATE.
 4300                           SP
  (1)                           ; ************************************************************************
  (1)                           ;*PROCEDURE
  (1)                           ;*---------
                                ;*BGNTST
                                ;*      INITIALIZE THE BYTE COUNTER, PAD COUNTER, AND ERROR COUNTER
                                ;*      CLEAR THE ERROR COUNTER
                                ;*      REQUEST A TU PORT FROM THE HOST
                                ;*      CALL SUBROUTINE CLEAR
                                ;*      IF NO UNITS SELECTED
                                ;*      :       THEN-ERROR-ABORT TEST
                                ;*      :       ELSE-CONTINUE
                                ;*      ENDIF
                                ;*      SET LOOP COMMAND/STATUS ON SELECTED PORT
                                ;*      CLEAR THE COMMAND BYTE FOR SELECTED PORT
                                ;*      SET WRITE PATH ENABLE
                                ;*      INITIALIZE THE EXPECTED RESULTS TABLE POINTER
                                ;*      SET WMC/XMC RESTART
                                ;*      SELECT THE SINGLE STEP CLOCK
                                ;*      CLEAR WMC/XMC RESTART
                                ;*      LOAD THE BYTE COUNT WITH 1113 (BITS 15:0)
                                ;*      SET DDR CONTROL TO "IN"
                                ;*      SELECT THE IMAGE MODE OF DATA FORMATTING
                                ;*      LOAD THE RESIDUAL CHARACTER WITH 6
                                ;*      LOAD THE PAD COUNT WITH 6 (BITS 7:0)
                                ;*      CLEAR THE PAD COUNT (BITS 15:8)
                                ;*      ENABLE ALL TRACKS LOAD TRACK ENABLE WITH ONES
                                ;*      SET THE CLOCK COUNT TO 180(10)
                                ;*      BGNDO
                                ;*      :       CALL SUBROUTINE CLKSYS
                                ;*      :       DECREMENT THE CLOCK COUNT
                                ;*      :       DO UNTIL CLOCK COUNT=0
                                ;*      ENDDO
                                ;*      SET WRITE MICRO TO WRITE MICRO ENABLE, TRANSLATOR ENABLE, GCR, AND WRITE
                                ;*      SET THE CLOCK COUNT TO 256(10)
                                ;*      BGNDO
```

```
1377                    ;*      :    CALL SUBROUTINE CLKSYS
1378                    ;*      :    DECREMENT THE CLOCK COUNT
1379                    ;*      :    DO UNTIL CLOCK COUNT=0 OR XL WRITE CLOCK=1
1380                    ;*    ENDDO
1381                    ;*    IF CLOCK COUNT=0
1382                    ;*      :    THEN-FATAL ERROR-ABORT TEST
1383                    ;*      :    ELSE-CONTINUE
1384                    ;*    ENDIF
1385                    ;*    CLEAR THE GROUP COUNTER
1386                    ;*    BGNDO
1387                    ;*      :    CALL SUBROUTINE CLKSYS
1388                    ;*      :    CALL SUBROUTINE CLKSYS
1389                    ;*      :    INPUT THE TRANSLATOR OUTPUT DATA
1390                    ;*      :    IF THE ACTUAL TRANSLATOR OUTPUT=EXPECTED TRANSLATOR OUTPUT
1391                    ;*      :    :    THEN-CONTINUE
1392                    ;*      :    :    ELSE-ERROR
1393                    ;*      :    ENDIF
1394                    ;*      :    INPUT THE TRANSLATOR OUTPUT DATA PARITY
1395                    ;*      :    IF DATA OUTPUT DURING DATA GROUP
1396                    ;*      :    :    THEN-EXPECTED PARITY=ODD PARITY FOR OUTPUT DATA
1397                    ;*      :    :    ELSE-EXPECTED PARITY=1
1398                    ;*      :    ENDIF
1399                    ;*      :    IF INPUT PARITY=EXPECTED PARITY
1400                    ;*      :    :    THEN-CONTINUE
1401                    ;*      :    :    ELSE-ERROR
1402                    ;*      :    ENDIF
1403                    ;*      :    SET THE LOOP COUNT TO 14(8)
1404                    ;*      :    BGNDO
1405                    ;*      :    :    CALL SUBROUTINE CLKSYS
1406                    ;*      :    :    DECREMENT THE LOOP COUNT
1407                    ;*      :    :    DO UNTIL THE LOOP COUNT = 0
1408                    ;*      :    ENDDO
1409                    ;*      :    IF XL WRITE CLOCK = 1
1410                    ;*      :    :    THEN - CONTINUE
1411                    ;*      :    :    ELSE - ERROR - IMPROPER TRANSITION RATE FOR GCR
1412                    ;*      :    ENDIF
1413                    ;*      :    INCREMENT THE EXPECTED RESULTS TABLE POINTER
1414                    ;*      :    INCREMENT THE GROUP COUNTER
1415                    ;*      :    IF GROUP COUNTER < 158
1416                    ;*      :    :  THEN-DECREMENT THE EXPECTED RESULTS TABLE POINTER BY ONE GROUP.
1417                    ;*      :    :  ELSE-CONTINUE
1418                    ;*      :    ENDIF
1419                    ;*      :    DO UNTIL THE EXPECTED RESULTS POINTER=END OF TABLE
1420                    ;*    ENDDO
1421                    ;*    SET THE CLOCK COUNT TO 256(10)
1422                    ;*    BGNDO
1423                    ;*      :    CALL SUBROUTINE CLKSYS
1424                    ;*      :    DECREMENT THE CLOCK COUNT
1425                    ;*      :    DO UNTIL CLOCK COUNT=0 OR TRANSLATOR OR WRITE MICRO DONE=0
1426                    ;*    ENDDO
1427                    ;*    IF CLOCK COUNT=0
1428                    ;*      :    THEN-ERROR-TIMEOUT WAITING FOR WRITE MICRO DONE
1429                    ;*      :    ELSE-CONTINUE
1430                    ;*    ENDIF
```

```
1431                                    ;*    IF TRANSLATOR DONE
1432                                    ;*    :    THEN-ERROR-TRANSLATOR DONE BEFORE WRITE MICRO
1433                                    ;*    :    ELSE-SET THE CLOCK COUNT TO 256(10)
1434                                    ;*    :    BGNDO
1435                                    ;*    :    :    CALL SUBROUTINE CLKSYS
1436                                    ;*    :    :    DECREMENT THE CLOCK COUNT
1437                                    ;*    :    :    DO UNTIL TRANSLATOR DONE OR CLOCK COUNT=0
1438                                    ;*    :    ENDDO
1439                                    ;*    :    IF CLOCK COUNT=0
1440                                    ;*    :    :    THEN-ERROR-TIMEOUT WAITING FOR TRANSLATOR DONE
1441                                    ;*    :    :    ELSE-CONTINUE
1442                                    ;*    :    ENDIF
1443                                    ;*    ENDIF
1444                                    ;*ENDTST
1445  4300                              S
 (1)                                    ; ***************************************************************************
1446
```

```
1448  4300                              TEST1:  TESTX   @1
 (1)  4300  3E  01              7.0                      MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28         18.0                      CALL    TSET            ;SETUP THE TEST
1449                                    ;%GCR - PREAMBLE/POSTAMBLE TEST - 1 DATA GROUP
1450                                    ;&M8958, M8959, M8955'S
1451
1452  4305  3E  01              7.0  XLLP1:  MVI     A,W.RST         ;CLEAR THE XMC ENABLE BIT
1453  4307  D3  D3             10.0          OUT     WMCCTL          ;DISABLE THE XMC
1454  4309  3E  00              7.0          MVI     A,0
1455  430B  D3  F0             10.0          OUT     CLKCTL          ;SET CLOCK CONTROL TO NORMAL OPERATION
1456  430D  AF                  4.0          XRA     A
1457  430E  D3  D3             10.0          OUT     WMCCTL          ;FINISH THE RESTART OF WMC/XMC
1458                                                                  ;XMC SHOULD START SELF TEST
1459  4310  3E  FF              7.0          MVI     A,@377          ;LOAD A COUNT VALUE
1460  4312  00                  4.0  1$:     NOP                     ;NOP
1461  4313  3D                  4.0          DCR     A               ;DECREMENT COUNT
1462  4314  C2  12  43         10.0          JNZ     1$              ;CONTINUE UNTIL COUNT = 0
1463                                                                  ;
1464
1465  4317  AF                  4.0          XRA     A               ;CLEAR UNIT MAP STORAGE
1466  4318  32  25  47         13.0          STA     UNITMP          ;1ST TIME HERE
1467  431B  32  29  47         13.0          STA     EXDLPC          ;CLEAR THE ERROR COUNTER
1468
1469  431E  3E  30              7.0  T1LOOP: MVI     A,@60           ;INIT THE BYTE COUNTER
1470  4320  D3  D7             10.0          OUT     CNTCTL          ;TO NORMAL MODE
1471  4322  3E  70              7.0          MVI     A,@160          ;INIT THE PAD COUNTER
1472  4324  D3  D7             10.0          OUT     CNTCTL          ;TO NORMAL MODE
1473  4326  3E  B0              7.0          MVI     A,@260          ;INIT THE ERROR COUNTER
1474  4328  D3  D7             10.0          OUT     CNTCTL          ;TO NORMAL MODE
1475  432A                                  CLRECT
 (1)  432A  AF                  4.0                      XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  432B  D3  D6             10.0                      OUT     ERRCNT          ;CLEAR BIT, 7-0
 (1)  432D  D3  D6             10.0                      OUT     ERRCNT          ;CLEAR BITS 15-8
```

```
1477                                    ;REQUEST THE HOST FOR THE PORTS TO BE USED IN THIS TEST...VALUE WILL
1478                                    ;BE IN CAS REG 12 LOW BYTE ON RETURN FROM HOST REQUEST.
1479
1480  432F  3A  25  47  13.0            LDA     UNITMP          ;GET THE UNIT MAP STORAGE
1481  4332  A7           4.0            ANA     A
1482  4333  C2  4C  43  10.0            JNZ     T1LPA           ;JUMP OVER IF ALREADY HAVE IT
1483  4336                              REQ     @7,0,0,0
 (1)  4336  CD  06  28  18.0                            CALL    REQST
 (1)  4339  00                                          .BYTE   0               ;DATA PATTERN NUMBER
 (1)  433A  00  00                                      .WORD   0               ;SYSTEM "0" COUNT
 (1)  433C  00  00                                      .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  433E  00                                          .BYTE                   ;DATA COMPARE FLAG IF =1
 (1)  433F  07                                          .BYTE   @7              ;REQUEST CODE
1484  4340                              RIN     R12L
 (1)  4340  DB  94          10.0                IN      R12L            ;READ R12L INTO AC
 (1)  4342  7F               4.0                MOV     A,A             ;RETRY LINK
1485  4343  32  25  47  13.0            STA     UNITMP          ;SAVE THE UNIT MAP INFO FROM HOST
1486  4346  AF           4.0            XRA     A
1487  4347  CD  C6  45  18.0            CALL    CLEAR
1488  434A  06  00        7.0            MVI     B,0
1489  434C  3A  25  47  13.0  T1LPA:     LDA     UNITMP          ;GET SELECTED UNIT MAP
1490  434F  E6  01        7.0            ANI     $01             ;UNIT 0 PRESENT?
1491  4351  C2  7D  43  10.0            JNZ     FOUND           ;YES-USE IT
1492  4354  04           4.0            INR     B
1493  4355  3A  25  47  13.0            LDA     UNITMP          ;
1494  4358  E6  02        7.0            ANI     $02             ;UNIT 1 PRESENT?
1495  435A  C2  7D  43  10.0            JNZ     FOUND           ;YES-USE IT
1496  435D  04           4.0            INR     B
1497  435E  3A  25  47  13.0            LDA     UNITMP          ;
1498  4361  E6  04        7.0            ANI     $04             ;UNIT 2 PRESENT?
1499  4363  C2  7D  43  10.0            JNZ     FOUND           ;YES-USE IT
1500  4366  04           4.0            INR     B
1501  4367  3A  25  47  13.0            LDA     UNITMP
1502  436A  E6  08        7.0            ANI     $08             ;UNIT 3 PRESENT
1503  436C  C2  7D  43  10.0            JNZ     FOUND           ;YES-USE IT
1504                                                            ;ELSE - OPERATOR ERROR
1505  436F                              ERR     EXIT,DUMMY
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  436F  CD  09  28  18.0                            CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4372  01                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4373  00                                          .BYTE
 (1)  4374  CD  15  28  18.0                    DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4377  DA  B7  45  10.0                            JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1506                                    ;>OPERATOR ERROR NO TM78 UNITS SELECTED
1507                                    ;>FATAL ERROR - TEST ABORTED
1508                                    ;*XMC6 MICRO ERROR 01
1509                                    ;*OPERATOR ERROR NO TM78 UNITS SELECTED
1510                                    ;*FATAL ERROR - TEST ABORTED
1511  437A  C3  B7  45  10.0            JMP     EXIT
```

```
1513                                        ;SET UP FOR THE DATA TRANSFER
1514  437D  DB  E0          10.0  FOUND:  IN      INTSTA
1515  437F  E6  80           7.0          ANI     BIT7
1516  4381  B0                4.0          ORA     B
1517  4382  D3  E0          10.0          OUT     MBSEL
1518  4384  3E  40           7.0          MVI     A,P.LCS         ;KILL THE MTA BOARD
1519  4386  D3  48          10.0          OUT     PDIAG
1520  4388  AF                4.0          XRA     A
1521  4389  D3  40          10.0          OUT     TCMD
1522  438B  3E  10           7.0          MVI     A,P.WPEN        ;SET WRITE PATH ENABLE
1523  438D  D3  4C          10.0          OUT     PENAB
1524  438F  21  F1  45       10.0          LXI     H,GXPTBL        ;LOAD ADDRESS OF EXPECTED
1525  4392  22  2B  47      16.0          SHLD    POINTER         ;RESULTS TABLE INTO POINTER
1526  4395  3E  01           7.0          MVI     A,W.RST         ;LOAD THE WMC/XMC RESTART CODE
1527  4397  D3  D3          10.0          OUT     WMCCTL          ;ISSUE THE COMMAND
1528  4399  3E  05           7.0          MVI     A,SSTEP         ;SELECT SINGLE STEP WRITE CLOCK
1529  439B  D3  F0          10.0          OUT     CLKCTL          ;INITIALIZE THE CLOCK
1530  439D  32  23  47      13.0          STA     CCTLWD          ;SAVE SOFTWARE IMAGE
1531  43A0  AF                4.0          XRA     A               ;CLEAR THE ACC.
1532  43A1  D3  D3          10.0          OUT     WMCCTL          ;CLEAR THE WMC/XMC RESTART BIT
1533  43A3  21  59  04       10.0          LXI     H,1113          ;GET THE BYTE COUNT - 1 DATA GROUP
1534  43A6  23                6.0          INX     H               ;CORRECT FOR INTEL 8253 CHIP
1535  43A7  7D                4.0          MOV     A,L
1536  43A8  D3  D4          10.0          OUT     BYTCNT          ;LOAD BITS 7-0
1537  43AA  7C                4.0          MOV     A,H
1538  43AB  D3  D4          10.0          OUT     BYTCNT          ;CLEAR THE BYTE COUNTER BITS 15-8
1539  43AD  AF                4.0          XRA     A
1540  43AE  32  2D  47      13.0          STA     TRANSCNT        ;INITIALIZE THE TRANSITION COUNT
1541  43B1  32  2E  47      13.0          STA     TRANSCNT+1
1542  43B4  32  27  47      13.0          STA     DPCNT           ;CLEAR THE PARITY TEST COUNTER
1543  43B7  32  28  47      13.0          STA     DPCNT+1
1544  43BA  32  24  47      13.0          STA     GRPCNT
1545  43BD  3E  88           7.0          MVI     A,@210
1546  43BF  D3  DB          10.0          OUT     DDRCTL          ;SET THE DDR CONTROL TO 'IN'
1547  43C1  3E  50           7.0          MVI     A,$50           ;LOAD THE IMAGE MODE COMMAND
1548  43C3  D3  D0          10.0          OUT     DATACTL         ;SET THE DATA CONTROL MODE
1549  43C5  3E  18           7.0          MVI     A,24            ;LOAD THE RESIDUAL CHARACTER VALUE
1550  43C7  D3  D1          10.0          OUT     RESCHR          ;SET THE RESIDUAL CHARACTER
1551  43C9  3E  06           7.0          MVI     A,6             ;LOAD PAD COUNT VALUE
1552  43CB  3C                4.0          INR     A               ;CORRECT FOR INTEL 8253 CHIP
1553  43CC  D3  D5          10.0          OUT     PADCNT          ;WRITE TO WMC BITS 7-0
1554  43CE  AF                4.0          XRA     A               ;LOAD HI VALUE
1555  43CF  D3  D5          10.0          OUT     PADCNT          ;WRITE TO WMC
1556  43D1  D3  D2          10.0          OUT     TRKENA          ;CLEAR THE TRACK ENABLE BITS 1ST
1557  43D3  D3  D2          10.0          OUT     TRKENA
1558  43D5  3E  FF           7.0          MVI     A,$FF           ;LOAD ACC WITH ALL ONES
1559  43D7  D3  D2          10.0          OUT     TRKENA          ;ENABLE DATA TRACKS
1560  43D9  D3  D2          10.0          OUT     TRKENA          ;ENABLE PARITY TRACK
1561  43DB  06  B4           7.0          MVI     B,180           ;LOAD THE CLOCK COUNT
```

```
1563  43DD  CD  BA  45      18.0  CLKRS:  CALL   CLKSYS              ;CLOCK IT
1564  43E0  05                4.0          DCR    B                  ;
1565  43E1  C2  DD  43      10.0          JNZ    CLKRS              ;CONTINUE WITH RESTART CLOCKING
1566
1567  43E4  3E  D8            7.0          MVI    A,W.ENAB.X.ENAB!W.GCR!W.WRITE
1568  43E6  D3  D3           10.0          OUT    WMCCTL             ;START THE OPERATION
1569  43E8  06  FF            7.0  CLK1:   MVI    B,$FF              ;SET TIMEOUT COUNT TO 256
1570                                                                 ;
1571  43EA  CD  BA  45      18.0  CLK:    CALL   CLKSYS              ;SINGLE STEP THE WRITE PATH
1572                                                                 ;
1573  43ED  DB  D0           10.0          IN     WMCSTA             ;GET THE WMC STATUS
1574  43EF  E6  01            7.0          ANI    X.WCLK             ;X.WCLK SET?
1575  43F1  C2  06  44       10.0          JNZ    BGNPRE             ;YES-BEGINNING OF PREAMBLE
1576  43F4  05                4.0          DCR    B                  ;NO-DECREMENT CLOCK
1577  43F5  C2  EA  43       10.0          JNZ    CLK                ;CONTINUE UNTIL ZERO
1578  43F8                                 ERR    TSTEND,ABOCNO
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  43F8  CD  09  28       18.0                 CALL   ERLP               ,PROCESS ERROR - DO 2.3
 (1)        0002                               MSGN   =      MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43FB  02                                 .BYTE  MSGN               ;MESSAGE NUMBER ID
 (1)  43FC  00                                 .BYTE
 (1)  43FD  CD  15  28       18.0          ABOCNO::  CALL   CKLOP               ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4400  DA  18  28       10.0                    JC     TSTEND             ;LOOP ADDRESS IF LOOP SPECIFIED
1579                                 ;>WRITE PATH TIMEOUT WHILE WAITING FOR "XL CLK" TO SET
1580                                 ;>FATAL ERROR - TEST ABORTED
1581                                 ;*XMC6 MICRO ERROR 02
1582                                 ;*WRITE PATH TIMEOUT WHILE WAITING FOR "XL CLK" TO SET
1583                                 ;*FATAL ERROR - TEST ABORTED
1584
1585  4403  C3  18  28       10.0          JMP    TSTEND
```

```
1587  4406  CD  BA  45   18.0  BGNPRE: CALL   CLKSYS          ;CLOCK THE SYSTEM
1588  4409  CD  BA  45   18.0          CALL   CLKSYS          ;CLOCK AGAIN
1589  440C  DB  40        10.0          IN     TSTS            ;GET THE DATA READ
1590  440E  47             4.0          MOV    B,A             ;SAVE IT IN B=ADATA
1591  440F                               ROUT   ADATA           ;STORE ACTUAL DATA IN CAS
 (1)  440F  D3  94        10.0                 OUT    ADATA           ;WRITE AC INTO ADATA
 (1)  4411  7F             4.0                 MOV    A,A             ;RETRY LINK
1592                                            ;
1593  4412  2A  2D  47   16.0          LHLD   TRANSCNT        ;GET THE TRANSITION COUNT
1594  4415  23             6.0          INX    H               ;UPDATE
1595  4416  22  2D  47   16.0          SHLD   TRANSCNT        ;RESTORE THE TRANSITION COUNT
1596  4419  7D             4.0          MOV    A,L             ;GET TRANSCOUNT LOW
1597  441A                               ROUT   R05L            ;WRITE TO CAS
 (1)  441A  D3  8A        10.0                 OUT    R05L            ;WRITE AC INTO R05L
 (1)  441C  7F             4.0                 MOV    A,A             ;RETRY LINK
1598  441D  7C             4.0          MOV    A,H             ;GET TRANSCOUNT HIGH
1599  441E                               ROUT   R05H            ;WRITE TO CAS
 (1)  441E  D3  8B        10.0                 OUT    R05H            ;WRITE AC INTO R05H
 (1)  4420  7F             4.0                 MOV    A,A             ;RETRY LINK
1600  4421  2A  28  47   16.0          LHLD   POINTER         ;GET POINTER TO EXPECTED DATA
1601  4424  7E             7.0          MOV    A,M             ;GET EXPECTED DATA BYTE
1602  4425  32  26  47   13.0          STA    XDATA           ;SAVE FOR LATER
1603  4428  B8             4.0          CMP    B               ;COMPARE WITH ACTUAL
1604  4429  CA  41  44   10.0          JZ     PRECN1          ;CONTINUE IF EQUAL
1605                                            ;ELSE-ERROR
1606  442C                               ROUT   EDATA           ;STORE EXPECTED DATA IN CAS
 (1)  442C  D3  95        10.0                 OUT    EDATA           ;WRITE AC INTO EDATA
 (1)  442E  7F             4.0                 MOV    A,A             ;RETRY LINK
1607  442F  3A  29  47   13.0  XDLPC1: LDA    EXDLPC          ;GET THE CURRENT COUNT AT ERROR
1608  4432  A7             4.0          ANA    A
1609  4433  C2  3C  44   10.0          JNZ    XDLPC2          ;STAY AT THIS COUNT IF NOT ZERO
1610  4436  2A  2D  47   16.0          LHLD   TRANSCNT        ;GET THE COUNT AT THE ERROR
1611  4439  22  29  47   16.0          SHLD   EXDLPC          ;SAVE IT
1612  443C                       XDLPC2: ERRB   XDLOOP,PRECN1,@4
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  443C  CD  12  28   18.0                       CALL   ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0003                                    MSGN   =      MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  443F  03                                      .BYTE  MSGN            ;MESSAGE NUMBER ID
 (1)  4440  04                                      .BYTE  @4              ;PRINT ROUTINE NUMBER
 (1)  4441  CD  15  28   18.0                PRECN1::       CALL   CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4444  DA  4A  44   10.0                       JC     XDLOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1613                                    ;>DATA STATE RECEIVED FROM THE XMC INCORRECT
1614                                    ;*XMC6 MICRO ERROR 03
1615                                    ;*DATA STATE RECEIVED FROM THE XMC INCORRECT
1616                                    ;*ACTUAL = NNNN
1617                                    ;*EXPECTED = NNNN
1618  4447  C3  6C  44   10.0          JMP    XDLPC           ;SKIP OVER ERROR LOOP CONTROL
1619
```

```
1621
1622                                        ;HERE IS THE ERROR LOOP CONTROL
1623
1624  444A  3A  94  4F      13.0  XDLOOP:  LDA   LPFLG        ;GET THE LOOP ON ERROR FLAG
1625  444D  A7                    4.0           ANA   A
1626  444E  C2  5A  44        10.0           JNZ   XD_P1       ;JUMP IF FLAG IS SET
1627  4451  21  00  00        10.0           LXI   H,0         ;CLEAR H&L
1628  4454  22  29  47        16.0           SHLD  EXDLPC      ;CLEAR THE ERROR POINTER
1629  4457  C3  6C  44        10.0           JMP   XDLPC       ;AND CONTINUE TESTING
1630
1631  445A  2A  2D  47        16.0  XDLP1:   LHLD  TRANSCNT    ;GET THE CURRENT XLWRITE CLOCK COUNT
1632  445D  EB                    4.0           XCHG            ;PUT H&L IN D&E
1633  445E  2A  29  47        16.0           LHLD  EXDLPC      ;GET THE FAILING COUNT
1634  4461  7A                    4.0           MOV   A,D
1635  4462  2F                    4.0           CMA
1636  4463  57                    4.0           MOV   D,A
1637  4464  7B                    4.0           MOV   A,E
1638  4465  2F                    4.0           CMA
1639  4466  5F                    4.0           MOV   E,A
1640  4467  13                    6.0           INX   D
1641  4468  19                   10.0           DAD   D           ;CURRENT COUNT - FAILING COUNT?
1642  4469  DA  1E  43           10.0           JC    T1LOOP      ;YES - LOOP NOW
```

```
1644  446C  DB  48        10.0  XDLPC:  IN      PSTAT           ;GET THE PORT PARITY BIT
1645  446E  0F              4.0          RRC                     ;RIGHT JUSTIFY
1646  446F  0F              4.0          RRC                     ;
1647  4470  E6  01          7.0          ANI     $01             ;REMOVE OTHER BITS
1648  4472  57              4.0          MOV     D,A             ;STORE ACTUAL PARITY IN REG D
1649
1650  4473                               ROUT    ADATA           ;STORE ACTUAL PARITY IN CAS
 (1)  4473  D3  94          10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4475  7F              4.0                  MOV     A,A             ;RETRY LINK
1651  4476  2A  2B  47      16.0         LHLD    POINTER
1652  4479  01  46  46      10.0         LXI     B,GXPDAT        ;LOAD BEGINNING ADDRESS OF EXPECTED DATA
1653  447C  78              4.0          MOV     A,B             ;MAKE THE TWOS COMPLEMENT OF REGISTER
1654  447D  2F              4.0          CMA                     ;PAIR B & C
1655  447E  47              4.0          MOV     B,A
1656  447F  79              4.0          MOV     A,C
1657  4480  2F              4.0          CMA
1658  4481  4F              4.0          MOV     C,A
1659  4482  03              6.0          INX     B
1660  4483  09             10.0          DAD     B               ;ADD POINTER AND BEGINNING OF DATA TABLE
1661  4484  D2  AB  44      10.0          JNC     PPREAM          ;CHECK PREAMBLE PARITY TRACK
1662
1663  4487  2A  2B  47      16.0         LHLD    POINTER         ;RESET H & L REGS
1664  448A  01  8C  46      10.0         LXI     B,GDATEN        ;GET THE ENDING ADDRESS OF EXPECTED DATA
1665  448D  78              4.0          MOV     A,B             ;MAKE THE TWOS COMPLEMENT OF REGISTER
1666  448E  2F              4.0          CMA                     ;PAIR B & C
1667  448F  47              4.0          MOV     B,A
1668  4490  79              4.0          MOV     A,C
1669  4491  2F              4.0          CMA
1670  4492  4F              4.0          MOV     C,A
1671  4493  03              6.0          INX     B
1672  4494  09             10.0          DAD     B               ;ADD POINTER AND END OF DATA TABLE
1673  4495  DA  AB  44      10.0          JC      PPOST           ;CHECK POSTAMBLE PARITY TRACK
1674
1675
1676
1677  4498  01  DC  46      10.0  DPTST:  LXI     B,PXPTBL        ;POINT TO EXPECTED PARITY DATA TABLE
1678  449B  2A  27  47      16.0          LHLD    DPCNT           ;POINT TO THE TABLE OFFSET NUMBER
1679  449E  23              6.0          INX     H               ;UPDATE IT
1680  449F  22  27  47      16.0          SHLD    DPCNT           ;SAVE NEW COUNT
1681  44A2  09             10.0          DAD     B               ;MAKE REAL POINTER
1682  44A3  7E              7.0          MOV     A,M             ;GET EXPECTED PARITY STATE
1683  44A4  A7              4.0          ANA     A               ;SET CONDITION BITS
1684  44A5  FA  F8  44      10.0          JM      PTYC            ;JMP IF NO COMPARE ON THIS DATA
1685  44A8  C3  B0  44      10.0          JMP     PTY             ;TEST THE PARITY BIT
1686
1687                                ;HERE TO CHECK THE PREAMBLE/POSTAMBLE PARITY TRACK BIT
1688
1689  44AB                         PPOST:
1690  44AB  3A  26  47      13.0  PPREAM: LDA     XDATA           ;GET THE DATA EXPECTED
1691  44AE  E6  01          7.0          ANI     1               ;PARITY TRACK IS SAME
```

```
1693  44B0  BA                4.0  PTY:    CMP     D                ;SEE IF EXPECTED=ACTUAL
1694  44B1  CA  CD  44         10.0         JZ      PTSTOK           ;JMP IF OK
1695  44B4                            ROUT    EDATA            ;SAVE THE EXPECTED DATA
 (1)  44B4  D3  95              10.0                 OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  44B6  7F                   4.0                 MOV     A,A              ;RETRY LINK
1696  44B7  7A                   4.0         MOV     A,D              ;ERROR - GET THE ACTUAL DATA
1697  44B8                            ROUT    ADATA            ;SAVE FOR PRINTOUT
 (1)  44B8  D3  94              10.0                 OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  44BA  7F                   4.0                 MOV     A,A              ;RETRY LINK
1698
1699  44BB  3A  29  47          13.0  PTY1:   LDA     EXDLPC           ;GET THE COUNT AT THE ERROR
1700  44BE  A7                   4.0         ANA     A
1701  44BF  C2  C8  44          10.0         JNZ     PTY2             ;JUMP IF AT AN ERROR NOW
1702  44C2  2A  2D  47          16.0         LHLD    TRANSCNT         ;GET THE COUNT AT ERROR
1703  44C5  22  29  47          16.0         SHLD    EXDLPC           ;SAVE IT
1704  44C8                       PTY2:    ERRB    XPLOOP,PTSTOK,@4
 (1)                             ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44C8  CD  12  28          18.0                 CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)       0004                                    MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44CB  04                                    .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  44CC  04                                    .BYTE   @4               ;PRINT ROUTINE NUMBER
 (1)  44CD  CD  15  28          18.0         PTSTOK::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44D0  DA  D6  44          10.0                 JC      XPLOOP           ;LOOP ADDRESS IF LOOP SPECIFIED
1705                                ;>PARITY STATE RECEIVED FROM THE XMC INCORRECT
1706                                ;*XMC6 MICRO ERROR 04
1707                                ;*PARITY STATE RECEIVED FROM THE XMC INCORRECT
1708                                ;*ACTUAL = NNNN
1709                                ;*EXPECTED = NNNN
1710  44D3  C3  F8  44          10.0         JMP     PTYC             ;JMP OVER CODE
1711
1712
1713
1714  44D6  3A  94  4F          13.0  XPLOOP: LDA     LPFLG            ;GET THE LOOP ON ERROR FLAG
1715  44D9  A7                   4.0         ANA     A
1716  44DA  C2  E6  44          10.0         JNZ     XPLP1            ;JUMP IF FLAG IS SET
1717  44DD  21  00  00          10.0         LXI     H,0              ;CLEAR H&L
1718  44E0  22  29  47          16.0         SHLD    EXDLPC           ;CLEAR THE ERROR POINTER
1719  44E3  C3  F8  44          10.0         JMP     PTYC             ;AND CONTINUE TESTING
1720
1721  44E6  2A  2D  47          16.0  XPLP1:  LHLD    TRANSCNT         ;GET THE CURRENT XLWRITE CLOCK COUNT
1722  44E9  EB                   4.0         XCHG                     ;PUT H&L IN D&E
1723  44EA  2A  29  47          16.0         LHLD    EXDLPC           ;GET THE FAILING COUNT
1724  44ED  7A                   4.0         MOV     A,D
1725  44EE  2F                   4.0         CMA
1726  44EF  57                   4.0         MOV     D,A
1727  44F0  7B                   4.0         MOV     A,E
1728  44F1  2F                   4.0         CMA
1729  44F2  5F                   4.0         MOV     E,A
1730  44F3  13                   6.0         INX     D
1731  44F4  19                  10.0         DAD     D                ;CURRENT COUNT = FAILING COUNT?
1732  44F5  DA  1E  43          10.0         JC      T1LOOP           ;YES - LOOP NOW
```

```
1734  44F8  2A  2B  47   16.0  PTYC:  LHLD   POINTER          ;NO-UPDATE THE
1735  44FB  23                 6.0         INX    H              ;TABLE POINTER
1736  44FC  22  2B  47   16.0         SHLD   POINTER          ;ADDRESS
1737  44FF  2A  2B  47   16.0         LHLD   POINTER
1738  4502  01  50  46   10.0         LXI    B,RESYNC
1739  4505  78                 4.0         MOV    A,B
1740  4506  2F                 4.0         CMA
1741  4507  47                 4.0         MOV    B,A
1742  4508  79                 4.0         MOV    A,C
1743  4509  2F                 4.0         CMA
1744  450A  4F                 4.0         MOV    C,A
1745  450B  03                 6.0         INX    B
1746  450C  09                10.0         DAD    B              ;IS A GROUP DONE
1747  450D  D2  29  45   10.0         JNC    CONT           ;NO-CONTINUE
1748  4510  3A  24  47   13.0         LDA    GRPCNT         ;YES-GET GROUP COUNT
1749  4513  3C                 4.0         INR    A              ;GROUP COUNT
1750  4514  FE  9E                 7.0         CPI    158            ;END OF GROUP 158?
1751  4516  CA  29  45   10.0         JZ     CONT           ;YES-CONTINUE
1752  4519  32  24  47   13.0         STA    GRPCNT
1753  451C  21  46  46   10.0         LXI    H,GXPDAT
1754  451F  22  2B  47   16.0         SHLD   POINTER
1755  4522  AF                 4.0         XRA    A
1756  4523  32  27  47   13.0         STA    DPCNT
1757  4526  32  28  47   13.0         STA    DPCNT+1
1758  4529  06  0E                 7.0  CONT:  MVI    B,14           ;LOAD THE CLOCK COUNT
1759  452B  CD  BA  45   18.0  1$:    CALL   CLKSYS         ;CLOCK THE WRITE PATH
1760  452E  05                 4.0         DCR    B              ;DECREMENT THE COUNT
1761  452F  C2  2B  45   10.0         JNZ    1$             ;CONTINUE UNTIL 8 CLOCKS
1762  4532  DB  D0                10.0         IN     WMCSTA
1763  4534  E6  01                 7.0         ANI    X.WCLK
1764  4536  C2  3E  45   10.0         JNZ    PRECN3
1765  4539                       ERR    T1LOOP,PRECN3,@4
 (1)                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4539  CD  09  28   18.0                CALL   ERLP           ;PROCESS ERROR - DO 2.3
 (1)        0005                      MSGN   =      MSGN+1         ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  453C  05                       .BYTE  MSGN           ;MESSAGE NUMBER ID
 (1)  453D  04                       .BYTE  @4
 (1)  453E  CD  15  28   18.0  PRECN3::      CALL   CKLOP          ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4541  DA  1E  43   10.0                JC     T1LOOP         ;LOOP ADDRESS IF LOOP SPECIFIED
1766                       ;>TRANSLATOR "XL WCLK" NOT SET AFTER EIGHT CLOCK CYCLES
1767                       ;>IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
1768                       ;*XMC6 MICRO ERROR 05
1769                       ;*TRANSLATOR "XL WCLK" NOT SET AFTER EIGHT CLOCK CYCLES
1770                       ;*IMPROPER TRANSITION RATE WILL OCCUR ON TAPE
```

```
 1772  4544  2A 2B 47      16.0          LHLD    POINTER          ;GET THE TABLF POINTER
 1773  4547  01 DB 46      10.0          LXI     B,GTBLEN         ;GET END OF TABLE ADDRESS
 1774  454A  78              4.0          MOV     A,B              ;FROM THE
 1775  454B  2F              4.0          CMA                      ;TWO'S COMPLIMENT
 1776  454C  47              4.0          MOV     B,A              ;OF THE TABLE END
 1777  454D  79              4.0          MOV     A,C              ;ADDRESS
 1778  454E  2F              4.0          CMA
 1779  454F  4F              4.0          MOV     C,A
 1780  4550  03              6.0          INX     B
 1781  4551  09             10.0          DAD     B                ;END OF TABLE?
 1782  4552  DA 58 45       10.0          JC      RECEND           ;YES
 1783  4555  C3 06 44       10.0          JMP     BGNPRE
 1784
 1785  4558  06 FF           7.0  RECEND: MVI     B,$FF            ;LOAD THE TIMEOUT COUNT
 1786  455A  CD BA 45       18.0  1$:     CALL    CLKSYS           ;CLOCK THE SYSTEM
 1787  455D  DB D0          10.0          IN      WMCSTA           ;INPUT THE WMC STATUS
 1788  455F  E6 40           7.0          ANI     W.DONN           ;WMC DONE?
 1789  4561  CA 74 45       10.0          JZ      RECCN1           ;YES-EXIT
 1790  4564  DB D0          10.0          IN      WMCSTA           ;GET WMC STATUS
 1791  4566  E6 80           7.0          ANI     X.DONN           ;XMC DONE
 1792  4568  CA 98 45       10.0          JZ      SEQERR           ;YES-ERROR
 1793  456B  05              4.0          DCR     B                ;NO-DECREMENT THE COUNT
 1794  456C  C2 5A 45       10.0          JNZ     1$               ;CONTINUE IF NOT ZERO
 1795  456F                               ERR     RECEND,RECCN1
  (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  456F  CD 09 28       18.0                  CALL    ERLP             ;PROCESS ERROR - DO 2.3
  (1)       0006                           MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4572  06                            .BYTE   MSGN             ;MESSAGE NUMBER ID
  (1)  4573  00                            .BYTE
  (1)  4574  CD 15 28       18.0  RECCN1::         CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
  (1)  4577  DA 58 45       10.0                  JC      RECEND           ;LOOP ADDRESS IF LOOP SPECIFIED
 1796                               ;>'WMC NOT DONE" TIMEOUT WAITING FOR IT TO GO TO ZERO
 1797                               ;*XMC6 MICRO ERROR 06
 1798                               ;*'WMC NOT DONE" TIMEOUT WAITING FOR IT TO GO TO ZERO
 1799
 1800  457A  06 FF           7.0          MVI     B,$FF            ;LOAD THE TIMEOUT COUNT
 1801  457C  CD BA 45       18.0  3$:     CALL    CLKSYS           ;CLOCK THE SYSTEM
 1802  457F  DB D0          10.0          IN      WMCSTA           ;INPUT THE WMC STATUS
 1803  4581  E6 80           7.0          ANI     X.DONN           ;XMC-DONE?
 1804  4583  CA 8F 45       10.0          JZ      RECCN3           ;YES-EXIT
 1805  4586  05              4.0          DCR     B                ;NO-DECREMENT THE COUNT
 1806  4587  C2 7C 45       10.0          JNZ     3$               ;CONTINUE IF NOT ZERO
 1807  458A                               ERR     T1LOOP,RECCN3
  (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  458A  CD 09 28       18.0                  CALL    ERLP             ;PROCESS ERROR - DO 2.3
  (1)       0007                           MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  458D  07                            .BYTE   MSGN             ;MESSAGE NUMBER ID
  (1)  458E  00                            .BYTE
  (1)  458F  CD 15 28       18.0  RECCN3::         CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
  (1)  4592  DA 1E 43       10.0                  JC      T1LOOP           ;LOOP ADDRESS IF LOOP SPECIFIED
 1808                               ;>"XMC NOT DONE" TIMEOUT WAITING FOR IT TO GO TO ZERO
 1809                               ;*XMC6 MICRO ERROR 07
 1810                               ;*"XMC NOT DONE" TIMEOUT WAITING FOR IT TO GO TO ZERO
 1811  4595  C3 A3 45       10.0          JMP     XMEND
```

```
1813  4598                                  SEQERR: ERR     T1LOOP,RECCN2
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4598  CD  09  28        18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0008                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  459B  08                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  459C  00                                         .BYTE
 (1)  459D  CD  15  28        18.0              RECCN2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45A0  DA  1E  43        10.0                        JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1814                                           ;>''XMC NOT DONE'' RESET BEFORE ''WMC NOT DONE'' RESET
1815                                           ;*XMC6 MICRO ERROR 10
1816                                           ;*''XMC NOT DONE'' RESET BEFORE ''WMC NOT DONE'' RESET
1817
1818  45A3                                  XMEND:  ENDTST  XLLP1
 (1)                                           ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  45A3                                             REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  45A3  CD  06  28        18.0                        CALL    REQST
 (2)  45A6  00                                         .BYTE                   ;DATA PATTERN NUMBER
 (2)  45A7  00  00                                     .WORD                   ;SYSTEM '"' COUNT
 (2)  45A9  00  00                                     .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  45AB  00                                         .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  45AC  07                                         .BYTE   7               ;REQUEST CODE
 (1)  45AD  3A  9A  4F        13.0                        LDA     ITERA           ;GET ITERATION COUNT
 (1)  45B0  3D                 4.0                        DCR     A               ;DOWNCOUNT
 (1)  45B1  32  9A  4F        13.0                        STA     ITERA           ;SAVE COUNT
 (1)  45B4  F2  05  43        10.0                        JP      XLLP1           ;DO TEST UNTIL TILL = 0
1819
1820  45B7  C3  18  28        10.0  EXIT:   JMP     TSTEND
1821
```

```
                                        .SBTTL  SUBROUTINE CLKSYS

45BA                                    SD
 (1)                                    ; ************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS SUBROUTINE WILL GENERATE ONE SINGLE STEP USING THE CONTENTS OF
                                        ;*CCTLWD AS THE CLOCK CONTROL BYTE.
45BA                                    S
 (1)                                    ; ************************************************************************

45BA  3A 23 47        13.0  CLKSYS: LDA     CCTLWD          ;GET THE SOFTWARE CLOCK CONTROL IMAGE
45BD  F6 40            7.0          ORI     SSCLK           ;ADD IN 'CLK' BIT
45BF  D3 F0           10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL
45C1  E6 BF            7.0          ANI     @277            ;STRIP OFF CLOCK BIT
45C3  D3 F0           10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL WORD
45C5  C9              10.0          RET                     ;EXIT-CLOCK CYCLE
```

```
                                .SBTTL  SUBROUTINE CLEAR ALL TU PORTS

                                SSUB
                                ; ************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*CLEAR ALL TU PORTS
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                ;*AND LOOP MODES.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                ;*    CLEAR PORT SELECT FOR TRANSPORT
                                ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                ;*    CLEAR PORT AMTIE WORD
                                ;*ENDSUB
                                S
                                ; ************************************************************************

 1856  45C6  F5          12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
 1857  45C7  C5          12.0          PUSH    B
 1858  45C8  06  00       7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
```

```
1860  45CA  DB  E0          10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
1861  45CC  E6  80           7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
1862  45CE  B0               4.0          ORA     B               ;ADD IN THE SELECTED PORT #
1863  45CF  D3  E0          10.0          OUT     MBSEL           ;RESET TO THIS PORT
1864
1865  45D1  3E  80           7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
1866  45D3  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
1867  45D5  AF               4.0          XRA     A               ;CLEAR TU COMMAND A
1868  45D6  D3  40          10.0          OUT     TCMD            ;
1869  45D8  3E  81           7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
1870  45DA  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
1871  45DC  3E  00           7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
1872  45DE  D3  40          10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
1873  45E0  AF               4.0          XRA     A
1874  45E1  D3  44          10.0          OUT     TAMT            ;CLEAR AMTIE WORD
1875  45E3  D3  48          10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
1876  45E5  D3  4C          10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
1877
1878  45E7  04               4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
1879  45E8  78               4.0          MOV     A,B
1880  45E9  FE  04           7.0          CPI     4               ;DONE?
1881  45EB  C2  CA  45      10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
1882
1883  45EE  C1              10.0          POP     B               ;RESET B & C
1884  45EF  F1              10.0          POP     PSW             ;ALL DONE
1885  45F0  C9              10.0          RET                     ;EXIT
1886
```

```
1888
1889  45F1  FF                GXPTBL: .BYTE   $FF          ;PREAMBLE TERM GROUP (10101)
1890                                                       ;CHARACTER 1 = ALL ONES
1891  45F2  FF                        .BYTE   $FF          ;PREAMBLE TERM GROUP
1892                                                       ;CHAPACTER 2 = ALL ZEROS
1893  45F3  00                        .BYTE   $00          ;PREAMBLE TERM GROUP
1894                                                       ;CHARACTER 3 = ALL ONES
1895  45F4  00                        .BYTE   $00          ;PREAMBLE TERM GROUP
1896                                                       ;CHARACTER 4 = ALL ZEROS
1897  45F5  FF                        .BYTE   $FF          ;PREAMBLE TERM GROUP
1898                                                       ;CHARACTER 5 = ALL ONES
1899
1900  45F6  FF                        .BYTE   $FF          ;PREAMBLE SECOND GROUP (01111)
1901                                                       ;CHARACTER 1 = ALL ZEROS
1902  45F7  00                        .BYTE   $00          ;PREAMBLE SECOND GROUP
1903                                                       ;CHARACTER 2 = ALL ONES
1904  45F8  FF                        .BYTE   $FF          ;PREAMBLE SECOND GROUP
1905                                                       ;CHARACTER 3 = ALL ONES
1906  45F9  00                        .BYTE   $00          ;PREAMBLE SECOND GROUP
1907                                                       ;CHARACTER 4 = ALL ONES
1908  45FA  FF                        .BYTE   $FF          ;PREAMBLE SECOND GROUP
1909                                                       ;CHARACTER 5 = ALL ONES
1910                                                       ;
1911  45FB  00                        .BYTE   $00          ;PREAMBLE SYNC GROUP 1 (14 GRPS OF 11111)
1912                                                       ;CHARACTER 1 = ALL ONES
1913  45FC  FF                        .BYTE   $FF          ;PREAMBLE SYNC GROUP 1
1914                                                       ;CHARACTER 2 = ALL ONES
1915  45FD  00                        .BYTE   $00          ;PREAMBLE SYNC GROUP 1
1916                                                       ;CHARACTER 3 = ALL ONES
1917  45FE  FF                        .BYTE   $FF          ;PREAMBLE SYNC GROUP 1
1918                                                       ;CHARACTER 4 - ALL ONES
1919  45FF  00                        .BYTE   $00          ;PREAMBLE SYNC GROUP 1
1920                                                       ;CHARACTER 5 = ALL ONES
1921                                                       ;
1922  4600  FF                        .BYTE   $FF          ;PREAMBLE SYNC GROUP 2
1923                                                       ;CHARACTER 1 = ALL ONES
1924  4601  00                        .BYTE   $00          ;PREAMBLE SYNC GROUP 2
1925                                                       ;CHARACTER 2 = ALL ONES
1926  4602  FF                        .BYTE   $FF          ;PREAMBLE SYNC GROUP 2
1927                                                       ;CHARACTER 3 = ALL ONES
1928  4603  00                        .BYTE   $00          ;PREAMBLE SYNC GROUP 2
1929                                                       ;CHARACTER 4 = ALL ONES
1930  4604  FF                        .BYTE   $FF          ;PREAMBLE SYNC GROUP 2
1931                                                       ;CHARACTER 5 = ALL ONES
1932                                                       ;
1933  4605  00                        .BYTE   $00          ;PREAMBLE SYNC GROUP 3
1934                                                       ;CHARACTER 1 = ALL ONES
1935  4606  FF                        .BYTE   $FF          ;PREAMBLE SYNC GROUP 3
1936                                                       ;CHARACTER 2 = ALL ONES
1937  4607  00                        .BYTE   $00          ;PREAMBLE SYNC GROUP 3
1938                                                       ;CHARACTER 3 = ALL ONES
1939  4608  FF                        .BYTF   $FF          ;PREAMBLE SYNC GROUP 3
1940                                                       ;CHARACTFR 4 - ALL ONES
1941  4609  00                        .BYTE   $00          ;PREAMBLE SYNC GROUP 3
```

```
1942                                                    ;CHARACTER 5 = ALL ONES
1943                                                    ;
1944  460A  FF                 .BYTE   $FF             ;PREAMBLE SYNC GROUP 4
1945                                                    ;CHARACTER 1 = ALL ONES
1946  460B  00                 .BYTE   $00             ;PREAMBLE SYNC GROUP 4
1947                                                    ;CHARACTER 2 = ALL ONES
1948  460C  FF                 .BYTE   $FF             ;PREAMBLE SYNC GROUP 4
1949                                                    ;CHARACTER 3 = ALL ONES
1950  460D  00                 .BYTE   $00             ;PREAMBLE SYNC GROUP 4
1951                                                    ;CHARACTER 4 = ALL ONES
1952  460E  FF                 .BYTE   $FF             ;PREAMBLE SYNC GROUP 4
1953                                                    ;CHARACTER 5 = ALL ONES
1954                                                    ;
1955  460F  00                 .BYTE   $00             ;PREAMBLE SYNC GROUP 5
1956                                                    ;CHARACTER 1 = ALL ONES
1957  4610  FF                 .BYTE   $FF             ;PREAMBLE SYNC GROUP 5
1958                                                    ;CHARACTER 2 = ALL ONES
1959  4611  00                 .BYTE   $00             ;PREAMBLE SYNC GROUP 5
1960                                                    ;CHARACTER 3 = ALL ONES
1961  4612  FF                 .BYTE   $FF             ;PREAMBLE SYNC GROUP 5
1962                                                    ;CHARACTER 4 - ALL ONES
1963  4613  00                 .BYTE   $00             ;PREAMBLE SYNC GROUP 5
1964                                                    ;CHARACTER 5 = ALL ONES
1965                                                    ;
1966  4614  FF                 .BYTE   $FF             ;PREAMBLE SYNC GROUP 6
1967                                                    ;CHARACTER 1 = ALL ONES
1968  4615  00                 .BYTE   $00             ;PREAMBLE SYNC GROUP 6
1969                                                    ;CHARACTER 2 = ALL ONES
1970  4616  FF                 .BYTE   $FF             ;PREAMBLE SYNC GROUP 6
1971                                                    ;CHARACTER 3 = ALL ONES
1972  4617  00                 .BYTE   $00             ;PREAMBLE SYNC GROUP 6
1973                                                    ;CHARACTER 4 = ALL ONES
1974  4618  FF                 .BYTE   $FF             ;PREAMBLE SYNC GROUP 6
1975                                                    ;CHARACTER 5 = ALL ONES
1976                                                    ;
1977  4619  00                 .BYTE   $00             ;PREAMBLE SYNC GROUP 7
1978                                                    ;CHARACTER 1 = ALL ONES
1979  461A  FF                 .BYTE   $FF             ;PREAMBLE SYNC GROUP 7
1980                                                    ;CHARACTER 2 = ALL ONES
1981  461B  00                 .BYTE   $00             ;PREAMBLE SYNC GROUP 7
1982                                                    ;CHARACTER 3 = ALL ONES
1983  461C  FF                 .BYTE   $FF             ;PREAMBLE SYNC GROUP 7
1984                                                    ;CHARACTER 4 - ALL ONES
1985  461D  00                 .BYTE   $00             ;PREAMBLE SYNC GROUP 7
1986                                                    ;CHARACTER 5 = ALL ONES
1987                                                    ;
1988  461E  FF                 .BYTE   $FF             ;PREAMBLE SYNC GROUP 8
1989                                                    ;CHARACTER 1 = ALL ONES
1990  461F  00                 .BYTE   $00             ;PREAMBLE SYNC GROUP 8
1991                                                    ;CHARACTER 2 = ALL ONES
1992  4620  FF                 .BYTE   $FF             ;PREAMBLE SYNC GROUP 8
1993                                                    ;CHARACTER 3 = ALL ONES
1994  4621  00                 .BYTE   $00             ;PREAMBLE SYNC GROUP 8
1995                                                    ;CHARACTER 4 = ALL ONES
```

```
1996  4622  FF                    .BYTE   $FF             ;PREAMBLE SYNC GROUP 8
1997                                                      ;CHARACTER 5 = ALL ONES
1998                                                      ;
1999  4623  00                    .BYTE   $00             ;PREAMBLE SYNC GROUP 9
2000                                                      ;CHARACTER 1 = ALL ONES
2001  4624  FF                    .BYTE   $FF             ;PREAMBLE SYNC GROUP 9
2002                                                      ;CHARACTER 2 = ALL ONES
2003  4625  00                    .BYTE   $00             ;PREAMBLE SYNC GROUP 9
2004                                                      ;CHARACTER 3 = ALL ONES
2005  4626  FF                    .BYTE   $FF             ;PREAMBLE SYNC GROUP 9
2006                                                      ;CHARACTER 4 - ALL ONES
2007  4627  00                    .BYTE   $00             ;PREAMBLE SYNC GROUP 9
2008                                                      ;CHARACTER 5 = ALL ONES
2009                                                      ;
2010  4628  FF                    .BYTE   $FF             ;PREAMBLE SYNC GROUP 10
2011                                                      ;CHARACTER 1 = ALL ONES
2012  4629  00                    .BYTE   $00             ;PREAMBLE SYNC GROUP 10
2013                                                      ;CHARACTER 2 = ALL ONES
2014  462A  FF                    .BYTE   $FF             ;PREAMBLE SYNC GROUP 10
2015                                                      ;CHARACTER 3 = ALL ONES
2016  462B  00                    .BYTE   $00             ;PREAMBLE SYNC GROUP 10
2017                                                      ;CHARACTER 4 = ALL ONES
2018  462C  FF                    .BYTE   $FF             ;PREAMBLE SYNC GROUP 10
2019                                                      ;CHARACTER 5 = ALL ONES
2020                                                      ;
2021  462D  00                    .BYTE   $00             ;PREAMBLE SYNC GROUP 11
2022                                                      ;CHARACTER 1 = ALL ONES
2023  462E  FF                    .BYTE   $FF             ;PREAMBLE SYNC GROUP 11
2024                                                      ;CHARACTER 2 = ALL ONES
2025  462F  00                    .BYTE   $00             ;PREAMBLE SYNC GROUP 11
2026                                                      ;CHARACTER 3 = ALL ONES
2027  4630  FF                    .BYTE   $FF             ;PREAMBLE SYNC GROUP 11
2028                                                      ;CHARACTER 4 - ALL ONES
2029  4631  00                    .BYTE   $00             ;PREAMBLE SYNC GROUP 11
2030                                                      ;CHARACTER 5 = ALL ONES
2031                                                      ;
2032  4632  FF                    .BYTE   $FF             ;PREAMBLE SYNC GROUP 12
2033                                                      ;CHARACTER 1 = ALL ONES
2034  4633  00                    .BYTE   $00             ;PREAMBLE SYNC GROUP 12
2035                                                      ;CHARACTER 2 = ALL ONES
2036  4634  FF                    .BYTE   $FF             ;PREAMBLE SYNC GROUP 12
2037                                                      ;CHARACTER 3 = ALL ONES
2038  4635  00                    .BYTE   $00             ;PREAMBLE SYNC GROUP 12
2039                                                      ;CHARACTER 4 = ALL ONES
2040  4636  FF                    .BYTE   $FF             ;PREAMBLE SYNC GROUP 12
2041                                                      ;CHARACTER 5 = ALL ONES
2042                                                      ;
2043  4637  00                    .BYTE   $00             ;PREAMBLE SYNC GROUP 13
2044                                                      ;CHARACTER 1 = ALL ONES
2045  4638  FF                    .BYTE   $FF             ;PREAMBLE SYNC GROUP 13
2046                                                      ;CHARACTER 2 = ALL ONES
2047  4639  00                    .BYTE   $00             ;PREAMBLE SYNC GROUP 13
2048                                                      ;CHARACTER 3 = ALL ONES
2049  463A  FF                    .BYTE   $FF             ;PREAMBLE SYNC GROUP 13
```

```
2050
2051  463B  00                          .BYTE   $00             ;CHARACTER 4 - ALL ONES
2052                                                            ;PREAMBLE SYNC GROUP 13
2053                                                            ;CHARACTER 5 = ALL ONES
2054  463C  FF                          .BYTE   $FF             ;
2055                                                            ;PREAMBLE SYNC GROUP 14
2056  463D  00                          .BYTE   $00             ;CHARACTER 1 = ALL ONES
2057                                                            ;PREAMBLE SYNC GROUP 14
2058  463E  FF                          .BYTE   $FF             ;CHARACTER 2 = ALL ONES
2059                                                            ;PREAMBLE SYNC GROUP 14
2060  463F  00                          .BYTE   $00             ;CHARACTER 3 = ALL ONES
2061                                                            ;PREAMBLE SYNC GROUP 14
2062  4640  FF                          .BYTE   $FF             ;CHARACTER 4 = ALL ONES
2063                                                            ;PREAMBLE SYNC GROUP 14
2064                                                            ;CHARACTER 5 = ALL ONES
2065  4641  FF                          .BYTE   $FF             ;
2066                                                            ;MARK ONE GROUP
2067  4642  FF                          .BYTE   $FF             ;CHARACTER 1 = ALL ZEROS
2068                                                            ;MARK ONE GROUP
2069  4643  00                          .BYTE   $00             ;CHARACTER 2 = ALL ZEROS
2070                                                            ;MARK ONE GROUP
2071  4644  FF                          .BYTE   $FF             ;CHARACTER 3 = ALL ONES
2072                                                            ;MARK ONE GROUP
2073  4645  00                          .BYTE   $00             ;CHARACTER 4 = ALL ONES
2074                                                            ;MARK ONE GROUP
2075                                                            ;CHARACTER 5 = ALL ONES
2076                                                            ;
2077                          ;HERE IS THE DATA GROUP - 7 BYTES OF ALL 0'S (BYTE #86.)
2078                          ;ECC FOR DATA GRP = 0.
2079  4646  FF                GXPDAT: .BYTE   $FF             ;1ST CHAR IN 1ST HALF OF DATA GROUP - P   0
2080  4647  00                        .BYTE   0               ;2ND CHAR - P = 1
2081  4648  00                        .BYTE   0               ;3RD CHAR - P = 1
2082  4649  00                        .BYTE   0               ;4TH CHAR - P = 1
2083  464A  FF                        .BYTE   $FF             ;5TH CHAR - P = 1
2084
2085  464B  00                        .BYTE   0               ;1ST CHAR IN 2ND HALF OF DATA GROUP - P - 0
2086  464C  FF                        .BYTE   $FF             ;2ND CHAR - P = 1
2087  464D  FF                        .BYTE   $FF             ;3RD CHAR - P = 1
2088  464E  FF                        .BYTE   $FF             ;4TH CHAR - P = 1
2089  464F  00                        .BYTE   0               ;5TH CHAR (ECC CHAR =0) - P - 1
2090
2091  4650  FF                RESYNC: .BYTE   $FF             ;MARK TWO GROUP
2092  4651  00                        .BYTE   00
2093  4652  FF                        .BYTE   $FF
2094  4653  FF                        .BYTE   $FF
2095  4654  FF                        .BYTE   $FF
2096
2097  4655  00                        .BYTE   00              ;SYNC GROUP
2098  4656  FF                        .BYTE   $FF
2099  4657  00                        .BYTE   00
2100  4658  FF                        .BYTE   $FF
2101  4659  00                        .BYTE   00
2102
2103  465A  FF                        .BYTE   $FF             ;SYNC GROUP
```

```
2104  465B  00                        .BYTE   00
2105  465C  FF                        .BYTE   $FF
2106  465D  00                        .BYTE   00
2107  465E  FF                        .BYTE   $FF
2108
2109  465F  FF                        .BYTE   $FF             ;MARK ONE GROUP
2110  4660  FF                        .BYTE   $FF
2111  4661  00                        .BYTE   00
2112  4662  FF                        .BYTE   $FF
2113  4663  00                        .BYTE   00
2114
2115  4664  FF                        .BYTE   $FF             ;DATA SUB GROUP 1
2116  4665  00                        .BYTE   00
2117  4666  00                        .BYTE   00
2118  4667  00                        .BYTE   00
2119  4668  FF                        .BYTE   $FF
2120
2121  4669  00                        .BYTE   00              ;DATA SUB GROUP 2
2122  466A  FF                        .BYTE   $FF
2123  466B  FF                        .BYTE   $FF
2124  466C  FF                        .BYTE   $FF
2125  466D  00                        .BYTE   00
2126
2127                          ;END MARK = 11111 (ALL 1'S)
2128
2129  466E  FF                        .BYTE   $FF             ;END MARK GROUP
2130                                                          ;CHARACTER 1 = ALL ONES
2131  466F  00                        .BYTE   $00             ;END MARK GROUP
2132                                                          ;CHARACTER 2 = ALL ONES
2133  4670  FF                        .BYTE   $FF             ;END MARK GROUP
2134                                                          ;CHARACTER 3 = ALL ONES
2135  4671  00                        .BYTE   $C0             ;END MARK GROUP
2136                                                          ;CHARACTER 4 = ALL ONES
2137  4672  FF                        .BYTE   $FF             ;END MARK GROUP
2138                                                          ;CHARACTER 5 = ALL ONES
2139                                                          ;
```

```
2141
2142                                    ;4 PAD BYTES
2143
2144  4673  00                                .BYTE   @0              ;RESIDUAL GROUP - AFTER TRANSLATION - 1ST HALF
2145                                                                  ;CHARACTER 1
2146  4674  FF                                .BYTE   $FF             ;RESIDUAL GROUP - AFTER TRANSLATION
2147                                                                  ;CHARACTER 2
2148  4675  FF                                .BYTE   $FF             ;RESIDUAL GROUP - AFTER TRANSLATION
2149                                                                  ;CHARACTER 3
2150  4676  FF                                .BYTE   $FF             ;RESIDUAL GROUP - AFTER TRANSLATION
2151                                                                  ;CHARACTER 4
2152  4677  00                                .BYTE   @0              ;RESIDUAL GROUP - AFTER TRANSLATION
2153                                                                  ;CHARACTER 5
2154                                    ;2 PAD BYTES
2155                                    ;ACRC
2156                                    ;ECC
2157  4678  FF                                .BYTE   @377            ;RESIDUAL GROUP - AFTER TRANSLATION - 2ND HALF
2158                                                                  ;CHARACTER 1
2159  4679  82                                .BYTE   @202            ;RESIDUAL GROUP - AFTER TRANSLATION
2160                                                                  ;CHARACTER 2
2161  467A  82                                .BYTE   @202            ;RESIDUAL GROUP - AFTER TRANSLATION
2162                                                                  ;CHARACTER 3
2163  467B  24                                .BYTE   @044            ;RESIDUAL GROUP - AFTER TRANSLATION
2164                                                                  ;CHARACTER 4
2165  467C  59                                .BYTE   @131            ;RESIDUAL GROUP - AFTER TRANSLATION
2166                                                                  ;CHARACTER 5
2167                                    ;4 CRC BYTES
2168  467D  49                                .BYTE   @111            ;CRC GROUP - AFTER TRANSLATION - 1ST HALF
2169                                                                  ;CHARACTER 1
2170  467E  B6                                .BYTE   @266            ;CRC GROUP - AFTER TRANSLATION
2171                                                                  ;CHARACTER 2
2172  467F  59                                .BYTE   @131            ;CRC GROUP - AFTER TRANSLATION
2173                                                                  ;CHARACTER 3
2174  4680  B6                                .BYTE   @266            ;CRC GROUP - AFTER TRANSLATION
2175                                                                  ;CHARACTER 4
2176  4681  49                                .BYTE   @111            ;CRC GROUP - AFTER TRANSLATION
2177                                                                  ;CHARACTER 5
2178                                    ;2 CRC
2179                                    ;RESID
2180                                    ;ECC
2181  4682  3E                                .BYTE   @076            ;CRC GROUP - AFTER TRANSLATION - 2ND HALF
2182                                                                  ;CHARACTER 6
2183  4683  D1                                .BYTE   @321            ;CRC GROUP - AFTER TRANSLATION
2184                                                                  ;CHARACTER 7
2185  4684  3E                                .BYTE   @076            ;CRC GROUP - AFTER TRANSLATION
2186                                                                  ;CHARACTER 8
2187  4685  41                                .BYTE   @101            ;CRC GROUP - AFTER TRANSLATION
2188                                                                  ;CHARACTER 9
2189  4686  C9                                .BYTE   @311            ;CRC GROUP - AFTER TRANSLATION
2190                                                                  ;CHARACTER 10
2191                                                                  ;
```

```
2193
2194  4687  36                          .BYTE   @066        ;MARK TWO
2195                                                        ;CHARACTER 1 = ALL ONES
2196  4688  C9                          .BYTE   @311        ;MARK TWO
2197                                                        ;CHARACTER 2 = ALL ONES
2198  4689  36                          .BYTE   @066        ;MARK TWO
2199                                                        ;CHARACTER 3 = ALL ONES
2200  468A  36                          .BYTE   @066        ;MARK TWO
2201                                                        ;CHARACTER 4 = ALL ZEROS
2202  468B  36                          .BYTE   @066        ;MARK TWO
2203                                                        ;CHARACTER 5 = ALL ZEROS
2204                                                        ;
2205  468C  C9                GDATEN:   .BYTE   @311        ;POSTAMBLE SYNC GROUP 1
2206                                                        ;CHARACTER 1 = ALL ONES
2207  468D  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 1
2208                                                        ;CHARACTER 2 - ALL ONES
2209  468E  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 1
2210                                                        ;CHARACTER 3 = ALL ONES
2211  468F  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 1
2212                                                        ;CHARACTER 4 = ALL ONES
2213  4690  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 1
2214                                                        ;CHARACTER 5 = ALL ONES
2215                                                        ;
2216  4691  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 2
2217                                                        ;CHARACTER 1 = ALL ONES
2218  4692  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 2
2219                                                        ;CHARACTER 2 - ALL ONES
2220  4693  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 2
2221                                                        ;CHARACTER 3 = ALL ONES
2222  4694  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 2
2223                                                        ;CHARACTER 4 = ALL ONES
2224  4695  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 2
2225                                                        ;CHARACTER 5 = ALL ONES
2226                                                        ;
2227  4696  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 3
2228                                                        ;CHARACTER 1 = ALL ONES
2229  4697  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 3
2230                                                        ;CHARACTER 2 - ALL ONES
2231  4698  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 3
2232                                                        ;CHARACTER 3 = ALL ONES
2233  4699  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 3
2234                                                        ;CHARACTER 4 = ALL ONES
2235  469A  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 3
2236                                                        ;CHARACTER 5 = ALL ONES
2237                                                        ;
2238  469B  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 4
2239                                                        ;CHARACTER 1 = ALL ONES
2240  469C  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 4
2241                                                        ;CHARACTER 2 - ALL ONES
2242  469D  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 4
2243                                                        ;CHARACTER 3 = ALL ONES
2244  469E  C9                          .BYTE   @311        ;POSTAMBLE SYNC GROUP 4
2245                                                        ;CHARACTER 4 = ALL ONES
2246  469F  36                          .BYTE   @066        ;POSTAMBLE SYNC GROUP 4
```

```
2247                                                    ;CHARACTER 5 = ALL ONES
2248                                                    ;
2249  46A0  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 5
2250                                                    ;CHARACTER 1 = ALL ONES
2251  46A1  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 5
2252                                                    ;CHARACTER 2 - ALL ONES
2253  46A2  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 5
2254                                                    ;CHARACTER 3 = ALL ONES
2255  46A3  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 5
2256                                                    ;CHARACTER 4 = ALL ONES
2257  46A4  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 5
2258                                                    ;CHARACTER 5 = ALL ONES
2259                                                    ;
2260  46A5  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 6
2261                                                    ;CHARACTER 1 = ALL ONES
2262  46A6  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 6
2263                                                    ;CHARACTER 2 - ALL ONES
2264  46A7  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 6
2265                                                    ;CHARACTER 3 = ALL ONES
2266  46A8  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 6
2267                                                    ;CHARACTER 4 = ALL ONES
2268  46A9  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 6
2269                                                    ;CHARACTER 5 = ALL ONES
2270                                                    ;
2271  46AA  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 7
2272                                                    ;CHARACTER 1 = ALL ONES
2273  46AB  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 7
2274                                                    ;CHARACTER 2 - ALL ONES
2275  46AC  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 7
2276                                                    ;CHARACTER 3 = ALL ONES
2277  46AD  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 7
2278                                                    ;CHARACTER 4 = ALL ONES
2279  46AE  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 7
2280                                                    ;CHARACTER 5 = ALL ONES
2281                                                    ;
2282  46AF  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 8
2283                                                    ;CHARACTER 1 = ALL ONES
2284  46B0  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 8
2285                                                    ;CHARACTER 2 - ALL ONES
2286  46B1  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 8
2287                                                    ;CHARACTER 3 = ALL ONES
2288  46B2  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 8
2289                                                    ;CHARACTER 4 = ALL ONES
2290  46B3  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 8
2291                                                    ;CHARACTER 5 = ALL ONES
2292                                                    ;
2293  46B4  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 9
2294                                                    ;CHARACTER 1 = ALL ONES
2295  46B5  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 9
2296                                                    ;CHARACTER 2 - ALL ONES
2297  46B6  C9                  .BYTE   @311           ;POSTAMBLE SYNC GROUP 9
2298                                                    ;CHARACTER 3 = ALL ONES
2299  46B7  36                  .BYTE   @066           ;POSTAMBLE SYNC GROUP 9
2300                                                    ;CHARACTER 4 = ALL ONES
```

```
2301  46B8  C9                .BYTE   @311          ;POSTAMBLE SYNC GROUP 9
2302                                                ;CHARACTER 5 = ALL ONES
2303                                                ;
2304  46B9  36                .BYTE   @066          ;POSTAMBLE SYNC GROUP 10
2305                                                ;CHARACTER 1 = ALL ONES
2306  46BA  C9                .BYTE   @311          ;POSTAMBLE SYNC GROUP 10
2307                                                ;CHARACTER 2 - ALL ONES
2308  46BB  36                .BYTE   @066          ;POSTAMBLE SYNC GROUP 10
2309                                                ;CHARACTER 3 = ALL ONES
2310  46BC  C9                .BYTE   @311          ;POSTAMBLE SYNC GROUP 10
2311                                                ;CHARACTER 4 = ALL ONES
2312  46BD  36                .BYTE   @066          ;POSTAMBLE SYNC GROUP 10
2313                                                ;CHARACTER 5 = ALL ONES
2314                                                ;
2315  46BE  C9                .BYTE   @311          ;POSTAMBLE SYNC GROUP 11
2316                                                ;CHARACTER 1 = ALL ONES
2317  46BF  36                .BYTE   @066          ;POSTAMBLE SYNC GROUP 11
2318                                                ;CHARACTER 2 - ALL ONES
2319  46C0  C9                .BYTE   @311          ;POSTAMBLE SYNC GROUP 11
2320                                                ;CHARACTER 3 = ALL ONES
2321  46C1  36                .BYTE   @066          ;POSTAMBLE SYNC GROUP 11
2322                                                ;CHARACTER 4 = ALL ONES
2323  46C2  C9                .BYTE   @311          ;POSTAMBLE SYNC GROUP 11
2324                                                ;CHARACTER 5 = ALL ONES
2325                                                ;
2326  46C3  36                .BYTE   @066          ;POSTAMBLE SYNC GROUP 12
2327                                                ;CHARACTER 1 = ALL ONES
2328  46C4  C9                .BYTE   @311          ;POSTAMBLE SYNC GROUP 12
2329                                                ;CHARACTER 2 - ALL ONES
2330  46C5  36                .BYTE   @066          ;POSTAMBLE SYNC GROUP 12
2331                                                ;CHARACTER 3 = ALL ONES
2332  46C6  C9                .BYTE   @311          ;POSTAMBLE SYNC GROUP 12
2333                                                ;CHARACTER 4 = ALL ONES
2334  46C7  36                .BYTE   @066          ;POSTAMBLE SYNC GROUP 12
2335                                                ;CHARACTER 5 = ALL ONES
2336                                                ;
2337  46C8  C9                .BYTE   @311          ;POSTAMBLE SYNC GROUP 13
2338                                                ;CHARACTER 1 = ALL ONES
2339  46C9  36                .BYTE   @066          ;POSTAMBLE SYNC GROUP 13
2340                                                ;CHARACTER 2 - ALL ONES
2341  46CA  C9                .BYTE   @311          ;POSTAMBLE SYNC GROUP 13
2342                                                ;CHARACTER 3 = ALL ONES
2343  46CB  36                .BYTE   @066          ;POSTAMBLE SYNC GROUP 13
2344                                                ;CHARACTER 4 = ALL ONES
2345  46CC  C9                .BYTE   @311          ;POSTAMBLE SYNC GROUP 13
2346                                                ;CHARACTER 5 = ALL ONES
2347                                                ;
2348  46CD  36                .BYTE   @066          ;POSTAMBLE SYNC GROUP 14
2349                                                ;CHARACTER 1 = ALL ONES
2350  46CE  C9                .BYTE   @311          ;POSTAMBLE SYNC GROUP 14
2351                                                ;CHARACTER 2 - ALL ONES
2352  46CF  36                .BYTE   @066          ;POSTAMBLE SYNC GROUP 14
2353                                                ;CHARACTER 3 = ALL ONES
2354  46D0  C9                .BYTE   @311          ;POSTAMBLE SYNC GROUP 14
```

```
2355
2356  46D1  36                          .BYTE   @066          ;CHARACTER 4 = ALL ONES
2357                                                          ;POSTAMBLE SYNC GROUP 14
2358                                                          ;CHARACTER 5 = ALL ONES
2359  46D2  C9                          .BYTE   @311          ;
2360                                                          ;POSTAMBLE SECOND GROUP
2361  46D3  36                          .BYTE   @066          ;CHARACTER 1 = ALL ONES
2362                                                          ;POSTAMBLE SECOND GROUP
2363  46D4  C9                          .BYTE   @311          ;CHARACTER 2 = ALL ONES
2364                                                          ;POSTAMBLE SECOND GROUP
2365  46D5  36                          .BYTE   @066          ;CHARACTER 3 = ALL ONES
2366                                                          ;POSTAMBLE SECOND GROUP
2367  46D6  36                          .BYTE   @066          ;CHARACTER 4 = ALL ONES
2368                                                          ;POSTAMBLE SECOND GROUP
2369                                                          ;CHARACTER 5 = ALL ZEROS
2370  46D7  C9                          .BYTE   @311          ;
2371                                                          ;POSTAMBLE TERM GROUP
2372  46D8  C9                          .BYTE   @311          ;CHARACTER 1 = ALL ONES
2373                                                          ;POSTAMBLE TERM GROUP
2374  46D9  36                          .BYTE   @066          ;CHARACTER 2 = ALL ZEROS
2375                                                          ;POSTAMBLE TERM GROUP
2376  46DA  36                          .BYTE   @066          ;CHARACTER 3 = ALL ONES
2377                                                          ;POSTAMBLE TERM GROUP
2378  46DB  36                GTBLEN:   .BYTE   @066          ;CHARACTER 4 = ALL ZEROS
2379                                                          ;POSTAMBLE TERM GROUP
                                                              ;CHARACTER 5 = 0'S TO RESET TRACKS
```

```
                                .SBTTL  TABLE EXPECTED PARITY

                                ;HERE IS THE TABLE OF EXPECTED PARITY STATES FOR DATA & RESIDUAL GROUPS
                                ;ALL BYTES IN DATA GRP=0 (EVEN ECC) SO 'P' TRACK 4 X 5 IS 0001 = 01111

46DC  00                        PXPTBL: .BYTE   0               ;1ST ENTRY IS A DUMMY
46DD  00                                .BYTE   0               ;'P' TRANSLATION ACTS SAME AS OTHER
46DE  01                                .BYTE   1               ;TRACKS
46DF  00                                .BYTE   0
46E0  01                                .BYTE   1
46E1  00                                .BYTE   0
46E2  00                                .BYTE   0
46E3  01                                .BYTE   1
46E4  00                                .BYTE   0
46E5  01                                .BYTE   1
46E6  00                                .BYTE   0

46E7  FF                                .BYTE   @377            ;SKIP OVER MARK TWO GROUP
46E8  FF                                .BYTE   @377
46E9  FF                                .BYTE   @377
46EA  FF                                .BYTE   @377
46EB  FF                                .BYTE   @377

46EC  FF                                .BYTE   @377            ;SKIP OVER SYNC GROUP
46ED  FF                                .BYTE   @377
46EE  FF                                .BYTE   @377
46EF  FF                                .BYTE   @377
46F0  FF                                .BYTE   @377

46F1  FF                                .BYTE   @377            ;SKIP OVER SYNC GROUP
46F2  FF                                .BYTE   @377
46F3  FF                                .BYTE   @377
46F4  FF                                .BYTE   @377
46F5  FF                                .BYTE   @377

46F6  FF                                .BYTE   @377            ;SKIP OVER MARK ONE GROUP
46F7  FF                                .BYTE   @377
46F8  FF                                .BYTE   @377
46F9  FF                                .BYTE   @377
46FA  FF                                .BYTE   @377

46FB  00                                .BYTE   0               ;PTRACK FOR DATA GROUP
46FC  01                                .BYTE   1
46FD  00                                .BYTE   0
46FE  01                                .BYTE   1
46FF  00                                .BYTE   0
4700  00                                .BYTE   0
4701  01                                .BYTE   1
4702  00                                .BYTE   0
4703  01                                .BYTE   1
4704  00                                .BYTE   0

                                ;SKIP OVER END MARK GRP
```

```
2435  4705  FF                          .BYTE   @377
2436  4706  FF                          .BYTE   @377
2437  4707  FF                          .BYTE   @377
2438  4708  FF                          .BYTE   @377
2439  4709  FF                          .BYTE   @377
2440
2441                            ;RESIDUAL GRP
2442                            ;1ST GRP IS 4 PADS
2443
2444  470A  01                          .BYTE   1
2445  470B  00                          .BYTE   0
2446  470C  01                          .BYTE   1
2447  470D  00                          .BYTE   0
2448  470E  01                          .BYTE   1
```

```
                                        ;2ND GRP IS 2 PAD BYTES + ACRC OF 323 + ECC OF 225

470F  01                                        .BYTE   1
4710  00                                        .BYTE   0
4711  01                                        .BYTE   1
4712  00                                        .BYTE   0
4713  01                                        .BYTE   1

                                        ;CRC GRP 1ST HALF IS 4 CRC BYTES OF 136

4714  00                                        .BYTE   0
4715  01                                        .BYTE   1
4716  01                                        .BYTE   1
4717  01                                        .BYTE   1
4718  00                                        .BYTE   0

                                        ;2ND HALF OF CRC GRP IS 2 CRC OF 136 + RESID BYTE OF 006 + ECC BYTE OF 352

4719  01                                        .BYTE   1
471A  01                                        .BYTE   1
471B  01                                        .BYTE   1
471C  00                                        .BYTE   0
471D  01                                        .BYTE   1

                                        ;HERE IS THE MARK-2 PARITY TABLE

471E  00                                        .BYTE   0               ;MARK 2 IS 11100
471F  01                                        .BYTE   1
4720  00                                        .BYTE   0
4721  00                                        .BYTE   0
4722  00                                        .BYTE   0

```

```
                                .SBTTL  PROGRAM VARIABLES

4723  00                        CCTLWD: .BYTE   0               ;CLOCK CONTROL WORD
4724  00                        GRPCNT: .BYTE   0               ;GROUP COUNTER
4725  00                        UNITMP: .BYTE   0               ;UNIT MAP
4726  00                        XDATA:  .BYTE   0               ;COPY OF EXPECTED DATA
4727  00    00                  DPCNT:  .WORD   0               ;PARITY TEST COUNTER
4729  00    00                  EXDLPC: .WORD   0               ;TRANSITION COUNT AT ERROR
472B  00    00                  POINTER: .WORD  0               ;POINTER TO EXPECTED DATA TABLE
472D  00    00                  TRANSCNT: .WORD 0               ;TRANSITION COUNT

                                .END

0000
```

```
A     =%0001          ABOCNO  43FD  G     ADATA = 0094          AMTIEP= 0001
AMTIE7= 0002          ARAIDF= 0098        ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91          B     =%0000        BADST = 0090          BGNPRE  4406
BIT0  = 0001          BIT1  = 0002        BIT15 = 8000          BIT2  = 0004
BIT3  = 0008          BIT4  = 0010        BIT5  = 0020          BIT6  = 0040
BIT7  = 0080          BIT8  = 0100        BIT9  = 0200          BRKPBC= 4F0A
BRKRAM= 4F10          BRKSTR= 4E60        BRKXCT= 4F00          BSAVE   4F9C
BYTCNT= 00D4          BYTEH   4F24        BYTEL   4F23          C     =%0001
CASCT   4F21          CASCTL= 00A0        CASSTA= 00A0          CATTH = 0089
CATTL = 0088          CBUSST= 00A1        CBYTH = 008B          CBYTL = 008A
CCTLWD  4723          CDG1H = 0087        CDG1L = 0086          CDG2H = 0093
CDG2L = 0092          CDG3H = 0095        CDG3L = 0094          CDVTH = 008D
CDVTL = 008C          CHPTIE= 0028        CH0TIE= 0020          CH1TIE= 0021
CH2TIE= 0022          CH3TIE= 0023        CH4TIE= 0024          CH5TIE= 0025
CH6TIE= 0026          CH7TIE= 0027        CKLOP = 2815          CLEAR   45C6
CLK     43EA          CLKCTL= 00F0        CLKRS   43DD          CLKSYS  45BA
CLK1    43E8          CLOCK   4F26        CLRLP   45CA          CMC0H = 0099
CMC0L = 0098          CMC1H = 009B        CMC1L = 009A          CMC2H = 009D
CMC2L = 009C          CMC3H = 009F        CMC3L = 009E          CMINH = 0097
CMINL = 0096          CNTCTL= 00D7        CONT    4529          CRCWRD= 0018
CSAVE   4F9D          CSRLH = 0091        CSRLL = 0090          CTCH  = 0085
CTCL  = 0084          CTSTH = 008F        CTSTL = 008E          CXCTH = 0081
CXCTL = 0080          CXINH = 0083        CXINL = 0082          C.    = 0001
C.AVAI= 0080          C.DP  = 0008        C.DSE = 0010          C.DTU = 0003
C.DVA = 0008          C.FAIL= 00FC        C.FMT = 0070          C.FNCT= 003E
C.GO  = 0001          C.INTC= 00FE        C.MAIN= 0020          C.NSA = 0080
C.RCT = 00FC          C.SER = 0080        C.SHR = 0040          C.SKPC= 000F
C.TAPE= 0040          C.WCS = 0002        D     =%0002          DATACT= 00D0
DBUS    4F28          DBUSCT= 00C0        DBUSST= 00C0          DDRA  = 00D8
DDRAIN= 0010          DDRB  = 00D9        DDRBIN= 0002          DDRC  = 00DA
DDRCIN= 0001          DDRCO = 0088        DDRCTL= 00DB          DIAFLG  4F22
DIAGPG= 4300          DIAGRM= 4F90        DIARD = 000B          DONE1 = 0045
DONINT= 0010          DPCNT   4727        DPTST   4498          DSAVE   4F9E
DSE   = 0006          DUMMY   4374  G     D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010          D.LAGC= 0020        D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008          D.WR4 = 0080        E     =%0003          ECCBAD= 0042
ECCCOR= 0019          ECCOK = 0041        ECCSTA= 001A          ECCTST= 000E
EDATA = 0095          EOTCLR= 0003        ERFLG   4F93          ERLP  = 2809
ERLPA = 280F          ERLPB = 2812        ERLPE = 280C          ERNUM   4F90
ERRCNT= 00D6          ESAVE   4F9F        EXDLPC  4729          EXIT    45B7
E.ACRC= 0010          E.AMT = 0020        E.CDP = 0080          E.CRC = 0080
E.PNTR= 0008          E.RPE = 0040        E.STEC= 0001          E.TTEC= 0002
E.UNC = 0004          FIFORD= 006A        FORMAT  4F25          FOUND   437D
FWDTST= 0061          GCRID = 0089        GCRSET= 0002          GDATEN  468C
GOODTM= 0092          GRPCNT  4724        GTBLEN  46DB          GXPDAT  4646
GXPTBL  45F1          H     =%0004        HLSAVE  4FA0          IE    = 0008
INTSTA= 00E0          ITERA   4F9A        I.PWR = 0020          I.RMPE= 0040
I5.5  = 0010          I6.5  = 0020        I7.5  = 0040          KCALL = 005F
KCLR  = 007B          KDEP  = 003F        KENAB = 0078          KEXAM = 003E
KEYBRD= 00C8          KEY1  = 0078        KEY10 = 006D          KEY11 = 006E
KEY12 = 006F          KEY13 = 005C        KEY14 = 005D          KEY15 = 005E
KEY16 = 005F          KEY17 = 003C        KEY18 = 003D          KEY19 = 003E
KEY2  = 0079          KEY20 = 003F        KEY3  = 007A          KEY4  = 007B
KEY5  = 0074          KEY6  = 0075        KEY7  = 0076          KEY8  = 0077
```

```
KEY9   = 006C          KINTA = 006F          KLDAD = 003D          KN0    = 003C
KN1    = 005C          KN2   = 005D          KN3   = 005E          KN4    = 006C
KN5    = 006D          KN6   = 006E          KN7   = 0074          KN8    = 0075
KN9    = 0076          KU2   = 0079          KU3   = 007A          KU8    = 0077
L      =%0005          LBLANK= 000F          LCE   = 000B          LCH    = 000C
LCL    = 000D          LCP   = 000E          LC0   = 0000          LC1    = 0001
LC2    = 0002          LC3   = 0003          LC4   = 0004          LC5    = 0005
LC6    = 0006          LC7   = 0007          LC8   = 0008          LC9    = 0009
LDLEDA= 00CA           LDLEDB= 00CB          LDLEDC= 00CC          LDLEDD= 00CD
LDLEDE= 00CE           LDLEDF= 00CF          LKDIAG= 2800          LKKBD  = 004C
LKKEY  = 0049          LKLWMG= 0058          LKLWMP= 0055          LKLWPG= 0052
LKLWPP= 004F           LKMOD7= 0046          LKOPR  = 0046         LPFLG    4F94
LPNUM    4F92          M      =%0006         MBSEL  = 00E0         MB.A   = 0008
MB.B   = 0004          MEMTOP= 4FFF          MINUS  = 000A         MM     = 8000
MSF    = 0008          MSGN   = 0008         MTACLR= 0000          MT.ARA= 0020
MT.CPE= 0080           MT.DSE= 0001          MT.FWD= 0040          MT.INH= 0008
MT.LWR= 0004           MT.MOT= 0002          MT.NWT= 0080          MT.PEC= 0040
MT.PSB= 0004           MT.PS0= 0001          MT.PS1= 0002          MT.REV= 0020
MT.WRT= 0010           MT.Z   = 0008         M.ATA  = 0080         M.CAPE= 0020
M.CONT= 0080           M.DEM  = 0020         M.EBL  = 0010         M.EXC  = 0008
M.FAIL= 0008           M.ILR  = 0010         M.INIT= 0010          M.OCC  = 0020
M.ONLI= 0001           M.PE   = 0040         M.PORT= 0080          M.RDEN= 0002
M.RDPE= 0008           M.RUN  = 0004         M.SCLK= 0001          M.TRA  = 0040
M.UNIT= 0007           M.WCLK= 0040          M.WCLN= 0080          M.WREN= 0080
M5.5   = 0001          M6.5   = 0002         M7.5   = 0004         NOTCAP= 0088
OKAY   = 00FF          OPRRAM= 4300          OPSTRT= 0058          OPVER  = 0040
PADCNT= 00D5           PADCRC= 0080          PDIAG  = 0048         PEID   = 008A
PENAB  = 004C          PESET  = 0001         PL     = 00B1         POINTE   472B
PPOST    44AB          PPREAM   44AB         PRDD   = 004C         PRECN1   4441  G
PRECN3   453E  G       PRENF  = 009C         PS     = 00B2         PSTAT  = 0048
PSW    =%0009          PTSTOK   44CD  G      PTY      44B0         PTYC     44F8
PTY1     44BB          PTY2     44C8         PXPTBL   46DC         P.AMTP-  0001
P.BCIC= 0040           P.CMDP= 0020          P.INTE= 0080          P.LCS  = 0040
P.LWR  = 0020          P.RDP  = 0002         P.RPEN= 0004          P.RPST= 0002
P.RPOE= 0020           P.RP1E= 0010          P.RP2E= 0020          P.RP3E= 0010
P.SING= 0080           P.STAT= 0002          P.STPE= 0080          P.TACH= 0008
P.TUPR= 0010           P.WCSP= 0004          P.WDS  = 0040         P.WFLP= 0001
P.WPEN= 0010           P.WPOE= 0008          P.WP1E= 0004          P.WP2E= 0008
P.WP3E= 0004           P.5VOK= 0002          QUE    = 281B         QUEM   = 281E
RAMT   = 0010          RARA   = 0006         RARAI  = 0004         RCHBD0= 0048
RCHBD1= 0047           RCHOK  = 0046         RCHTST= 000C          RCLRT  = 000D
RCMD   = 000B          RCMLP  = 0003         RCONT  = 0080         RDATA  = 0017
RDCLK  = 0010          RDON   = 0011         READG  = 0007         RECCN1   4574  G
RECCN2   459D  G       RECCN3   458F  G      RECEND   4558         REND   = 0014
REQST  = 2806          RESCHR= 00D1          RESYNC   4650         REVTST= 0064
REWIND= 0004           RFIFOL= 0008          RGCLK  = 0002         RGCRI  = 0003
RIBG   = 0001          RILL   = 0012         RINST  = 000C         RMCTST= 0008
RMK2   = 0013          RNOP   = 0000         RPATH  = 0001         RPBAD  = 0044
RPCHI  = 0001          RPCLK  = 0003         RPCTL  = 0009         RPEI   = 0002
RPFAIL= 0000           RPF1   = 009D         RPF2   = 009E         RPOK   = 0043
RPOSTN= 0016           RPSTA  = 0015         RRCMT  = 000A         RSTAT  = 0002
RTIEB  = 000A          RTIER  = 0030         RTM    = 0005         RUNKI  = 0009
RUPTST= 005E           RWDUNL= 0005          R.AMT  = 0001         R.BOP  = 0008
R.DATA= 0040           R.DON  = 0002         R.DRDY= 0010          R.END  = 0010
```

```
R.ILL = 0004            R.JVOK= 0004            R.MK2 = 0008            R.PLOD= 0008
R.PLOO= 0010            R.PLO1= 0020            R.POST= 002C            R.STNM= 0002
R.STOP= 0004            R.STPC= 0001            R.TBJN= 0080            R.TSTD= 0040
R.VOK = 0080            R00H  = 0081            R00L  = 0080            R01H  = 0083
R01L  = 0082            R02H  = 0085            R02L  = 0084            R03H  = 0087
R03L  = 0086            R04H  = 0089            R04L  = 0088            R05H  = 008B
R05L  = 008A            R06H  = 008D            R06L  = 008C            R07H  = 008F
R07L  = 008E            R10H  = 0091            R10L  = 0090            R11H  = 0093
R11L  = 0092            R12H  = 0095            R12L  = 0094            R13H  = 0097
R13L  = 0096            R14H  = 0099            R14L  = 0098            R15H  = 009B
R15L  = 009A            R16H  = 009D            R16L  = 009C            R17H  = 009F
R17L  = 009E            R7.5  = 0010            SELCLR= 0000            SEQERR  4598
SETATA= 00A1            SID   = 0080            SOD   = 0080            SOE   = 0040
SP    =%0008            SSCLK = 0040            SSTEP = 0005            STACK = 4FFF
STATRM= 4F20            STPCT   4F20            STRSP = 5000            TADR00= 0080
TADR01= 0081            TADR02= 0082            TADR03= 0083            TADR04= 0084
TADR05= 0085            TADR06= 0086            TADR07= 0087            TADR10= 0088
TADR11= 0089            TADR12= 008A            TADR13= 008B            TAMT  = 0044
TASEL = 0080            TCMD  = 0040            TC.FWD= 0040            TC.INH= 0008
TC.LWR= 0004            TC.REV= 0020            TC.WRT= 0010            TEMP    4F99
TEST1   4300            TMF   = 0099            TMRDY = 0040            TRANSC  472D
TRKENA= 00D2            TSET  = 2803            TSTEND= 2818            TSTS  = 0040
TUSEL0= 00D1            TUSEL1= 00D2            TU78  = 0010            T.ATH0= 0001
T.ATH1= 0002            T.BOT = 0004            T.EOT = 0002            T.FPT = 0001
T.NTHR= 0004            T.ONL = 0020            T.PES = 0008            T.PSBJ= 0020
T.PSOJ= 0008            T.PS1J= 0010            T.RDY = 0080            T.RDYO= 0040
T.RWD = 0010            T.SCLK= 0002            T1LOOP  431E            T1LPA   434C
UIBG  = 00A1            UNITMP  4725            VALFC   4F98            VALTB   4F95
VELTST= 005B            WDR.P = 0010            WMCCTL= 00D3            WMCERR= 00DA
WMCSTA= 00D0            WRTCLK= 0000            WRTDAT= 00D5            W.ACRC= 0004
W.CRC = 0008            W.DIAG= 0002            W.DONN= 0040            W.ECC = 0010
W.ENAB= 0080            W.ERR = 0020            W.FMT = 0070            W.GCR = 0010
W.LEFT= 0004            W.ONES= 0020            W.RESI= 0002            W.REV = 0004
W.ROME= 0010            W.RST = 0001            W.SKIP= 000F            W.WRIT= 0008
W.XFER= 0020            X     =%000A            XDATA   4726            XDLOOP  444A
XDLPC   446C            XDLPC1  442F            XDLPC2  443C            XDLP1   445A
XLLP1   4305            XMEND   45A3            XPLOOP  44D6            XPLP1   44E6
X.DONN= 0080            X.ENAB= 0040            X.PEPE= 0002            X.ROME= 0001
X.WCLK= 0001            Y     =%000B            .     = 472F


 ERRORS DETECTED:  0

 *XMC6.A78/PTP,XMC6=NLIST,PARAM,MACRO,LIST,XMC6
 RUN-TIME: 4 6 0 SECONDS
```

     3   "MEMORY MAP" EXTENDED ADDRESS BIT
    19   READ MICRO CONTROLLER CONTROL & STATUS WORDS
   139   TRANSPORT/PORT COMMAND + STATUS WORDS
   380   COMMON ADDRESS SPACE REGISTERS
   444   MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
   485   MASSBUS CONTROL WORD DEFINITIONS
   557   WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
   709   KEYBOARD/DISPLAY CONTROL + STATUS WORD
   797   MISCELLANEOUS VARIABLE DEFINITIONS
   801   RAM ADDRESS DEFINITIONS
   819   LINKAGE TABLE ADDRESSES
   842   BIT DEFINITIONS
   856   RIM AND SIM INSTRUCTION BIT DEFINITIONS
   877   INTERRUPTABLE INSTRUCTION MACROS
   924   DIAGNOSTIC VARIABLES
   945   COMMON VARIABLE STORAGE
   959   MACRO CALL DEFINITIONS
  1047   TM78 MICRO-DIAGNOSTIC MACRO CALLS
  1101   8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
  1333   TEST 01 - TIE BUS INTEGRITY TEST
  1400   TEST 02 - LOOP COMMAND WRITE TO COMMAND READ
  1467   TEST 03 - STATUS REGISTER TEST
  1601   TEST 04 - READ PATH SELF TEST
  1729   TEST 05 - READ CHANNEL 0 SELF TEST
  1804   TEST 06 - READ CHANNEL 1 SELF TEST
  1879   TEST 07 - READ CHANNEL 2 SELF TEST
  1954   TEST 10 - READ CHANNEL 3 SELF TEST
  2029   TEST 11 - READ CHANNEL 4 SELF TEST
  2104   TEST 12 - READ CHANNEL 5 SELF TEST
  2179   TEST 13 - READ CHANNEL 6 SELF TEST
  2254   TEST 14 - READ CHANNEL 7 SELF TEST
  2329   TEST 15 - READ CHANNEL P SELF TEST
  2404   TEST 16 - READ CHANNEL AMTIE TEST - FROM TU PORT #0
  2525   TEST 17 - READ CHANNEL AMTIE TEST - FROM TU PORT #1
  2626   TEST 20 - READ CHANNEL AMTIE TEST - FROM TU PORT #2
  2728   TEST 21 - READ CHANNEL AMTIE TEST - FROM TU PORT #3
  2829   TEST 22 - ECC SELF TEST
  3028   SUBROUTINE CHECK AMTIE
  3167   SUBROUTINE CLEAR ALL TU PORTS
  3211   SUBROUTINE RDCTL
  3283   SUBROUTINE RCCLR
  3312   SUBROUTINE RCSTRT
  3341   SUBROUTINE RCCK
  3474   PROGRAM VARIABLES

```
961                     ;*************************************************************************
962                     ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
963                     ;ERROR MACRO CALLS
964                     ;
965                     ;       1  -    REQUEST HOST CPU TO PRINT:
966                     ;               "BYTE/SCLK COUNT NUMBER = LLL"
967                     ;               WHERE:
968                     ;               LLL = THE VALUE STORED IN CAS REGISTER 5
969                     ;                     (THE BYTE COUNT REGISTER 16 BITS).
970                     ;
971                     ;       2  -    REQUEST HOST CPU TO PRINT:
972                     ;               DATA FORMAT = MM
973                     ;               SKIP COUNT  = NN
974                     ;               WHERE:
975                     ;               MM = DATA FORMAT FROM CAS REGISTER 2
976                     ;               NN = SKIP COUNT FROM CAS REGISTER 2
977                     ;
978                     ;       3  -    REQUEST HOST CPU TO PRINT:
979                     ;               BYTE-SCLK COUNT = LLL
980                     ;               DATA FORMAT = MM
981                     ;               SKIP COUNT  = NN
982                     ;               WHERE:
983                     ;               LLL = AS ABOVE
984                     ;               MM  = AS ABOVE
985                     ;               NN  = AS ABOVE
986                     ;
987                     ;       4  -    REQUEST HOST TO PRINT:
988                     ;               TRANSITION COUNT = LLL
989                     ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
990                     ;
991                     ;       5  -    REQUEST HOST CPU TO PRINT:
992                     ;               EXPECTED 18 BITS =      E EEEEEE
993                     ;               ACTUAL 18 BITS =        A AAAAAA
994                     ;
995                     ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
996                     ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
997                     ;               OF CAS REG 15 LOW BYTE.
998                     ;
999                     ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
1000                    ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
1001                    ;               17 LOW BYTE SIGN BIT.
1002                    ;
1003                    ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
1004                    ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
1005                    ;               AND/OR ACTUAL DATA.
1006                    ;
1007                    ;       7  -    REQUEST HOST CPU TO PRINT:
1008                    ;               "SUBGROUP NUMBER = LLL"
1009                    ;               WHERE:
1010                    ;               LLL = THE VALUE STORED IN CAS REGISTER 5
1011                    ;                     (THE BYTE COUNT REGISTER 16 BITS).
1012                    ;
1013                    ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
1014                    ;*************************************************************************
```

```
1016                    ; ************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ; ************************************************************************
1035                    ; ************************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ; ************************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  RPM1 - READ PATH MICROCONTROLLER PART #1
                                .SBTTL  TEST 01 - TIE BUS INTEGRITY TEST
                                ;ID     RPM1-READ PATH MICRO CONTROLLER PART #1

4300                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*RPM - TIE BUS INTEGRITY TEST
4300                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS FOR STUCK BITS ON THE TIE BUS BY WRITING 16 (4 BIT)
                                ;*DATA PATTERNS (0-17) TO THE TIE BUS AND READING THEM FROM THE TIE BUS.
4300                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  SET NORMAL READ PATH CLOCKS
                                ;*  CLEAR TIE BUS VALUE
                                ;*  SET THE JAM TIE BUS BIT
                                ;*  BGNDO
                                ;*  :  OUTPUT TIE BUS VALUE TO THE TIE BUS
                                ;*  :  WAIT (NOP)
                                ;*  :  INPUT THE TIE BUS VALUE
                                ;*  :  IF TIE BUS OUTPUT=TIE BUS INPUT
                                ;*  :  :  THEN-CONTINUE
                                ;*  :  :  ELSE-ERROR
                                ;*  :  ENDIF
                                ;*  :  INCREMENT THE COMMAND VALUE
                                ;*  :  DO UNTIL THE COMMAND VALUE=20(8)
                                ;*  ENDDO
                                ;*ENDTST
4300                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*RPM1 MICRO TEST 01
                                ;*RPM1 MICRO ERROR 01
                                ;*RPM1-READ PATH-TIE BUS INTEGRITY TEST
                                ;*M8953, M8950, M8951
                                ;*TIE DATA WRITTEN NOT=READ
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
4300                            S
 (1)                            ; ************************************************************************
4300                            TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1) 4300  3E  01          7.0                 MVI     A,@1        ;DEFINE THE TEST NUMBER
 (1) 4302  CD  03  28     18.0                 CALL    TSET        ;SETUP THE TEST
                                ;%RPM1-READ PATH-TIE BUS INTEGRITY TEST
                                ;&M8953, M8950, M8951, M8952
```

```
1371  4305  3E  10        7.0  TST01X: MVI   A,RDCLK          ;SET NORMAL READ PATH CLOCKS
1372  4307  D3  F0       10.0          OUT   CLKCTL
1373
1374  4309  AF            4.0          XRA   A                ;CLEAR TIE BUS JAM VALUE
1375  430A  32  49  4B   13.0          STA   JAMDAT
1376
1377  430D  3E  80        7.0          MVI   A,R.TBJN                 ;SET JAM TIE BUS BIT
1378  430F  D3  09       10.0          OUT   RPCTL
1379
1380  4311  3A  49  4B   13.0  TIEL0:  LDA   JAMDAT          ;OUTPUT TIE BUS DATA TO TIE BUS
1381  4314  D3  0A       10.0          OUT   RTIEB
1382  4316                             ROUT  EDATA
 (1)  4316  D3  95       10.0                OUT     EDATA          ;WRITE AC INTO EDATA
 (1)  4318  7F            4.0                MOV     A,A            ;RETRY LINK
1383  4319  47            4.0          MOV   B,A              ;SAVE TIE BUS WRITTEN IN B
1384  431A  00            4.0          NOP                    ;WAIT
1385  431B  DB  30        10.0          IN    RTIER            ;INPUT THE TIE BUS VALUE
1386  431D  B8            4.0          CMP   B                ;IF TIE BUS OUTPUT=TIE BUS INPUT
1387  431E  CA  29  43    10.0          JZ    TIECNO           ;THEN CONTINUE
1388  4321                             ROUT  ADATA            ;ELSE-ERROR
 (1)  4321  D3  94       10.0                OUT     ADATA          ;WRITE AC INTO ADATA
 (1)  4323  7F            4.0                MOV     A,A            ;RETRY LINK
1389  4324                             ERRB  TIEL0,TIECNO
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4324  CD  12  28    18.0                CALL    ERLPB          ;PROCESS ERROR - DO 2.3
 (1)  0001                            MSGN  =       MSGN+1                 ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4327  01                        MSGN  .BYTE   MSGN           ;MESSAGE NUMBER ID
 (1)  4328  00                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4329  CD  15  28    18.0  TIECNO::       CALL    CKLOP                  ;CHECK LOOP FUNCTION - DO 2.2
 (1)  432C  DA  11  43    10.0                JC      TIEL0          ;LOOP ADDRESS IF LOOP SPECIFIED
1390                             ;>TIE DATA WRITTEN NOT=READ
1391
1392  432F  3A  49  4B   13.0          LDA   JAMDAT
1393  4332  3C            4.0          INR   A
1394  4333  32  49  4B   13.0          STA   JAMDAT
1395  4336  FE  10        7.0          CPI   @20
1396  4338  C2  11  43    10.0          JNZ   TIEL0
1397  433B                             ENDTST  TST01X
 (1)                                 ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  433B                            REQ   7                ;FAKE CALL TO KEEP TEST ALIVE
 (2)  433B  CD  06  28    18.0                CALL    REQST
 (2)  433E  00                              .BYTE                   ;DATA PATTERN NUMBER
 (2)  433F  00  00                          .WORD                   ;SYSTEM "" COUNT
 (2)  4341  00  00                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4343  00                              .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4344  07                              .BYTE   7               ;REQUEST CODE
 (1)  4345  3A  9A  4F    13.0          LDA   ITERA            ;GET ITERATION COUNT
 (1)  4348  3D            4.0          DCR   A                ;DOWNCOUNT
 (1)  4349  32  9A  4F    13.0          STA   ITERA            ;SAVE COUNT
 (1)  434C  F2  05  43    10.0          JP    TST01X           ;DO TEST UNTIL TILL = 0
1398
```

```
1400                                    .SBTTL  TEST 02 - LOOP COMMAND WRITE TO COMMAND READ
1401
1402  434F                              ST
 (1)                                    ;*****************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
1403                                    ;*RPM - LOOP COMMAND WRITE TO COMMAND READ
1404  434F                              SD
 (1)                                    ;*****************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
1405                                    ;*THIS TEST CHECKS THE READ PATH COMMAND REGISTER BY WRITING 256 DATA
1406                                    ;*PATTERNS 000-377(8) TO THE READ PATH COMMAND REGISTER AND CHECKING
1407                                    ;*THE RESULTS AT THE LOOP AROUND COMMAND REGISTER.
1408  434F                              SP
 (1)                                    ;*****************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
1409                                    ;*BGNTST
1410                                    ;*    SET NORMAL READ PATH CLOCK
1411                                    ;*    CLEAR THE COMMAND VALUE TO ZERO
1412                                    ;*    ENABLE THE READ MICRO CONTROLLER
1413                                    ;*    BGNDO
1414                                    ;*      :   OUTPUT COMMAND VALUE TO READ PATH COMMAND REGISTER
1415                                    ;*      :   INPUT COMMAND FROM READ PATH COMMAND REGISTER
1416                                    ;*      :       IF COMMAND OUTPUT = COMMAND INPUT
1417                                    ;*      :           THEN-CONTINUE
1418                                    ;*      :           ELSE-ERROR
1419                                    ;*      :       ENDIF
1420                                    ;*      :   INCREMENT THE COMMAND VALUE
1421                                    ;*      :   DO UNTIL COMMAND VALUE = ZERO
1422                                    ;*    ENDDO
1423                                    ;*ENDTST
1424  434F                              SE
 (1)                                    ;*****************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1425                                    ;*RPM1 MICRO TEST 02
1426                                    ;*RPM1 MICRO ERROR 02
1427                                    ;*RPM1-READ PATH-LOOP COMMAND WRITE TO READ
1428                                    ;*M8953, M8960
1429                                    ;*DATA WRITTEN NOT = READ
1430                                    ;*ACTUAL = NNNN
1431                                    ;*EXPECTED = NNNN
1432  434F                              S
 (1)                                    ;*****************************************************************************
1433
1434  434F                              TEST2:  TESTX   @2              ;INITIALIZE THE TEST
 (1)  434F  3E  02            7.0                     MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4351  CD  03  28       18.0                     CALL    TSET            ;SETUP THE TEST
1435                                    ;%RPM1-READ PATH-LOOP COMMAND WRITE TO READ
1436                                    ;@M8953, M8960, M8951, M8952
1437
1438  4354  3E  10             7.0     TST02X: MVI     A,RDCLK
```

```
1439  4356  D3  F0          10.0          OUT     CLKCTL          ;SET NORMAL CLOCKS
1440                                      ;
1441  4358  AF               4.0          XRA     A
1442  4359  32  46  4B       13.0          STA     DATAE           ;CLEAR THE  OMMAND VALUE
1443                                      ;
1444  435C  3E  00           7.0          MVI     A,0             ;LOAD THE READ PATH ENABLE BIT
1445  435E  D3  09          10.0          OUT     RPCTL           ;ENABLE THE READ PATH
1446
1447  4360  21  46  4B       10.0  RMCLP0: LXI     H,DATAE         ;LOAD ADDRESS OF THE DATA BYTE
1448  4363  7E               7.0          MOV     A,M             ;GET THE DATA BYTE
1449  4364  D3  0B          10.0          OUT     RCMD            ;OUTPUT TO THE READ MICRO
1450                                      ;
1451  4366  DB  03          10.0          IN      RCMLP           ;INPUT FROM THE READ MICRO
1452  4368  BE               7.0          CMP     M               ;EQUAL?
1453  4369  CA  78  43       10.0          JZ      RMCCNO
1454  436C                                ROUT    ADATA           ;NO-STORE ACTUAL DATA
 (1)  436C  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  436E  7F               4.0                  MOV     A,A             ;RETRY LINK
1455  436F  7E               7.0          MOV     A,M             ;GET THE EXPECTED DATA
1456  4370                                ROUT    EDATA           ;STORE EXPECTED DATA
 (1)  4370  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4372  7F               4.0                  MOV     A,A             ;RETRY LINK
1457  4373                                ERRB    RMCLP0,RMCCNO
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4373  CD  12  28       18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0002                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4376  02                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4377  00                                 .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4378  CD  15  28       18.0          RMCCNO::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  437B  DA  60  43       10.0                  JC      RMCLP0          ;LOOP ADDRESS IF LOOP SPECIFIED
1458                                ;>DATA WRITTEN NOT = READ
1459  437E  3A  46  4B       13.0          LDA     DATAE           ;GET THE DATA BYTE
1460  4381  3C               4.0          INR     A               ;INCREMENT
1461  4382  32  46  4B       13.0          STA     DATAE           ;STORE THE DATA BYTE
1462  4385  A7               4.0          ANA     A               ;SET THE CONDITION CODE
1463  4386  C2  60  43       10.0          JNZ     RMCLP0          ;DONE?
1464  4389                                ENDTST  TST02X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4389                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4389  CD  06  28       18.0                  CALL    REQST
 (2)  438C  00                                 .BYTE                   ;DATA PATTERN NUMBER
 (2)  438D  00  00                             .WORD                   ;SYSTEM "" COUNT
 (2)  438F  00  00                             .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4391  00                                 .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  4392  07                                 .BYTE   7               ;REQUEST CODE
 (1)  4393  3A  9A  4F       13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4396  3D               4.0          DCR     A               ;DOWNCOUNT
 (1)  4397  32  9A  4F       13.0          STA     ITERA           ;SAVE COUNT
 (1)  439A  F2  54  43       10.0          JP      TST02X          ;DO TEST UNTIL TILL = 0
1465
```

```
                                .SBTTL  TEST 03 - STATUS REGISTER TEST

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM - STATUS REGISTER TEST
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS EACH BIT IN THE READ PATH MICRO CONTROLLER STATUS BYTE
                                ;*BY ISSUING THE COMMANDS 012 - 172 TO THE READ PATH. THESE TEST COMMANDS
                                ;*LOAD THE STATUS REGISTER AS FOLLOWS:
                                ;*
                                ;*      COMMAND         STATUS
                                ;*      -------         ------
                                ;*
                                ;*       012              001
                                ;*       032              002
                                ;*       052              004
                                ;*       072              010
                                ;*       112              020
                                ;*       132              040
                                ;*       152              100
                                ;*       172              200
                                ;*
                                ;*AFTER EACH COMMAND IS ISSUED THE RESULTING STATUS IS VERIFIED.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     SET NORMAL READ PATH CLOCK
                                ;*       BGNDO
                                ;*        ENABLE THE READ PATH
                                ;*        SET EXPECTED STATUS BYTE = 1
                                ;*        LOAD THE COMMAND BYTE WITH 12(8)
                                ;*        BGNDO
                                ;*        :    OUTPUT THE COMMAND BYTE
                                ;*        :    DELAY 5 MICRO SECONDS
                                ;*        :    INPUT THE READ PATH ACTUAL STATUS BYTE
                                ;*        :       IF INPUT STATUS BYTE = EXPECTED STATUS BYTE
                                ;*        :            THEN-CONTINUE
                                ;*        :            ELSE-ERROR
                                ;*        :       ENDIF
                                ;*        :     SHIFT THE EXPECTED STATUS LEFT ONE BIT
                                ;*        :     ADD 20(8) THE COMMAND BYTE
                                ;*        :     DO UNTIL THE COMMAND BYTE = 212(8)
                                ;*     ENDDO
                                ;*     DO UNTIL ALL CLOCKS TESTED
                                ;*    ENDDO
                                ;*ENDTST
                                SE
```

```
 (1)                                ; ***************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1512                                ;*RPM1 MICRO TEST 03
1513                                ;*RPM1 MICRO ERROR 03
1514                                ;*RPM1-READ PATH-STATUS BYTE TEST
1515                                ;*M8953, M8960
1516                                ;*ACTUAL STATUS NOT = EXPECTED STATUS
1517                                ;*ACTUAL = NNNN
1518                                ;*EXPECTED = NNNN
1519  439D                          S
 (1)                                ; ***************************************************************************
1520
1521  439D                          TEST3:  TESTX   @3              ;INITIALIZE THE TEST
 (1)  439D  3E  03          7.0                 MVI     A,@3            ;DEFINE THE TEST NUMBER
 (1)  439F  CD  03  28     18.0                 CALL    TSET            ;SETUP THE TEST
1522                                ;%RPM1-READ PATH-STATUS BYTE TEST
1523                                ;&M8953, M8960
1524
1525  43A2  A7                4.0  TST03X: ANA     A
1526  43A3  3E  10            7.0  TST03A: MVI     A,RDCLK
1527  43A5  D3  F0           10.0          OUT     CLKCTL          ;SET NORMAL CLOCKS
1528  43A7  CD  12  44       18.0          CALL    STATCK          ;GO DO STATUS BYTE TEST
1529  43AA  D2  B2  43       10.0          JNC     TST03B          ;NO ERRORS - GO USE NEXT CLOCK RATE
1530  43AD                          ERRB    TST03A,TST03B
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VAI ID
 (1)  43AD  CD  12  28     18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0003                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43B0  03                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43B1  00                             .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  43B2  CD  15  28     18.0      TST03B::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43B5  DA  A3  43     10.0                 JC      TST03A          ;LOOP ADDRESS IF LOOP SPECIFIED
1531                            ;>ACTUAL STATUS NOT = EXPECTED STATUS, WITH CLOCK = NORMAL(20.00 MHZ)
1532                                ;
1533  43B8  A7                4.0          ANA     A
1534  43B9  3E  00            7.0          MVI     A,0             ;
1535  43BB  D3  F0           10.0          OUT     CLKCTL          ;SET -10% CLOCK
1536  43BD  CD  12  44       18.0          CALL    STATCK          ;GO DO STATUS BYTE TEST
1537  43C0  D2  C8  43       10.0          JNC     TST03C          ;NO ERRORS - GO DO THE NEXT CLOCK RATE
1538  43C3                          ERRB    TST03B,TST03C
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  43C3  CD  12  28     18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0004                             MSGN    =       MSGN+1          ;UPDATE MFSSAGE NUMBER FOR THIS
 (1)  43C6  04                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43C7  00                             .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  43C8  CD  15  28     18.0      TST03C::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43CB  DA  B2  43     10.0                 JC      TST03B          ;LOOP ADDRESS IF LOOP SPECIFIED
1539                            ;>ACTUAL STATUS NOT = EXPECTED STATUS, WITH CLOCK = -10%(18.08 MHZ)
1540                                ;
1541  43CE  A7                4.0          ANA     A
1542  43CF  3E  08            7.0          MVI     A,@10           ;
1543  43D1  D3  F0           10.0          OUT     CLKCTL          ;SET +10% CLOCK
1544  43D3  CD  12  44       18.0          CALL    STATCK  ;GO DO STATUS BYTE TEST
1545  43D6  D2  DE  43       10.0          JNC     TST03D          ;NO ERRORS - GO USE NEXT CLOCK RATE
```

```
1546  43D9                                        ERRB    TST03C,TST03D
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  43D9   CD   12   28        18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0005                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43DC   05                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43DD   00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  43DE   CD   15   28        18.0            TST03D::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43E1   DA   C8   43        10.0                    JC      TST03C          ;LOOP ADDRESS IF LOOP SPECIFIED
1547                                    ;>ACTUAL STATUS NOT = EXPECTED STATUS, WITH CLOCK = +10%(21.84 MHZ)
1548                                                    ;
1549  43E4   A7                 4.0             ANA     A
1550  43E5   3E   18             7.0             MVI     A,@30           ;
1551  43E7   D3   F0            10.0             OUT     CLKCTL          ;SET -30% CLOCK
1552  43E9   CD   12   44        18.0             CALL    STATCK          ;GO DO STATUS BYTE TEST
1553  43E    D2   F4   43        10.0             JNC     TST03E          ;NO ERRORS - WERE DONE - GO HOME
1554  43EF                                    ERRB    TST03D,TST03E
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  43EF   CD   12   28        18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0006                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43F2   06                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43F3   00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  43F4   CD   15   28        18.0            TST03E::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43F7   DA   DE   43        10.0                    JC      TST03D          ;LOOP ADDRESS IF LOOP SPECIFIED
1555                                    ;>ACTUAL STATUS NOT = EXPECTED STATUS, WITH CLOCK = -30%(15.60 MHZ)
1556                                                    ;
1557  43FA   A7                 4.0             ANA     A
1558  43FB                                    ENDTST  TST03X
 (1)                                      ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43FB                                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43FB   CD   06   28        18.0                    CALL    REQST
 (2)  43FE   00                                          .BYTE                   ;DATA PATTERN NUMBER
 (2)  43FF   00   00                                     .WORD                   ;SYSTEM "" COUNT
 (2)  4401   00   00                                     .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4403   00                                          .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  4404   07                                          .BYTE   7               ;REQUEST CODE
 (1)  4405   3A   9A   4F        13.0                    LDA     ITERA           ;GET ITERATION COUNT
 (1)  4408   3D                  4.0                     DCR     A               ;DOWNCOUNT
 (1)  4409   32   9A   4F        13.0                    STA     ITERA           ;SAVE COUNT
 (1)  440C   F2   A2   43        10.0                    JP      TST03X          ;DO TEST UNTIL TILL = 0
1559                                                    ;
1560  440F   C3   55   44        10.0             JMP     TEST4           ;GO DO THE NEXT TEST
1561                                                    ;
1562  4412   3E   00             7.0     STATCK:  MVI     A,0             ;LOAD THE READ PATH ENABLE COMMAND
1563  4414   D3   09            10.0              OUT     RPCTL           ;ENABLE THE READ PATH
1564                                                    ;
1565  4416   3E   01             7.0              MVI     A,@001          ;LOAD THE EXPECTED STATUS BYTE
1566  4418   32   42   4B        13.0             STA     RUNSTA          ;SAVE IN MEMORY
1567                                                    ;
1568  441B   3E   0A             7.0              MVI     A,RRCMT         ;LOAD THE BASE COMMAND BYTE
1569  441D   32   43   4B        13.0             STA     RUNCMD          ;SAVE IN MEMORY
1570                                                    ;
1571  4420   3A   43   4B        13.0     STALP1: LDA     RUNCMD          ;GET THE COMMAND BYTE
1572  4423   D3   0B            10.0              OUT     RCMD            ;ISSUE THE COMMAND
1573                                                    ;
```

```
1574  4425  00          4.0          NOP                    ;WAIT
1575  4426  00          4.0          NOP                    ;WAIT
1576  4427  00          4.0          NOP                    ;WAIT
1577  4428  00          4.0          NOP                    ;WAIT
1578                                                        ;
1579  4429  DB  02      10.0          IN      RSTAT          ;GET THE STATUS BYTE
1580  442B  21  42  4B  10.0          LXI     H,RUNSTA       ;LOAD RUN TIME COMMAND ADDRESS
1581  442E  BE           7.0          CMP     M              ;EQUAL?
1582  442F  CA  3F  44  10.0          JZ      STACN1         ;YES-CONTINUE
1583  4432                           ROUT    ADATA          ;NO-SAVE ACTUAL DATA IN CAS
 (1)  4432  D3  94       10.0                  OUT     ADATA          ;WRITE AC INTO ADATA
 (1)  4434  7F           4.0                   MOV     A,A            ;RETRY LINK
1584  4435  3A  42  4B  13.0          LDA     RUNSTA         ;GET EXPECTED STATUS
1585  4438                           ROUT    EDATA          ;SAVE EXPECTED DATA IN CAS
 (1)  4438  D3  95       10.0                  OUT     EDATA          ;WRITE AC INTO EDATA
 (1)  443A  7F           4.0                   MOV     A,A            ;RETRY LINK
1586  443B  37           4.0          STC                    ;SET CARRY FOR ERROR FLAG
1587  443C  C3  54  44  10.0          JMP     RETURN         ;GO REPORT
1588
1589  443F  3A  42  4B  13.0  STACN1: LDA     RUNSTA         ;GET THE RUNTIME STATUS
1590  4442  A7           4.0          ANA     A              ;CLEAR CARRY
1591  4443  17           4.0          RAL                    ;SHIFT THE DATA LEFT
1592  4444  32  42  4B  13.0          STA     RUNSTA         ;SAVE UPDATED RUN TIME STATUS
1593  4447  3A  43  4B  13.0          LDA     RUNCMD         ;GET THE RUNTIME COMMAND
1594  444A  C6  10       7.0          ADI     @020           ;ADD INCREMENTAL VALUE
1595  444C  32  43  4B  13.0          STA     RUNCMD         ;SAVE THE NEW COMMAND
1596  444F  FE  8A       7.0          CPI     @212           ;DONE?
1597  4451  C2  20  44  10.0          JNZ     STALP1         ;NO-KEEP IT GOING
1598  4454  C9          10.0  RETURN: RET                    ;YES-END OF TEST
1599
```

```
                                .SBTTL  TEST 04 - READ PATH SELF TEST

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM - SELF TEST - TEST
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST COMMANDS THE READ PATH MICRO TO RUN ITS SELF TEST DIAGNOSTIC.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  SET NORMAL READ PATH CLOCK
                                ;*  ENABLE THE READ PATH
                                ;*  OUTPUT THE READ PATH MICRO SELF TEST COMMAND
                                ;*  DELAY 10 MICRO SECONDS
                                ;*  INPUT THE READ PATH STATUS
                                ;*  IF INPUT STATUS = "RPOK" (103)
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  IF DATA READY, STATUS VALID, PREAMBLE ERROR AND BOP-1
                                ;*    AND CLOCK STOPPED AND STATISTICS/NORMAL=0
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  CALL SUBROUTINE RCCLR
                                ;*  SET STOP CLOCK AND STATISTICS/NORMAL
                                ;*  IF CLOCK STOPPED AND STATISTICS/NORMAL-1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  ENABLE THE READ PATH CLOCK
                                ;*  OUTPUT THE READ CHANNEL SELF TEST COMMAND
                                ;*  DELAY 75 MICRO SECONDS
                                ;*  INPUT THE READ PATH STATUS
                                ;*  IF READ PATH STATUS="RCHOK" (106)
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM1 MICRO TEST 04
                                ;*RPM1 MICRO ERROR 04
                                ;*RPM1-READ PATH-SELF TEST
                                ;*M8953, M8960
```

```
                                ;*SELF DIAGNOSIS FAILURE
                                ;*
                                ;*RPM1 MICRO TEST 04
                                ;*RPM1 MICRO ERROR 05
                                ;*RPM1-READ PATH-SELF TEST
                                ;*M8953, M8960
                                ;*READ PATH IF. REGISTER 01 INCORRECT AFTER SELF TEST
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 04
                                ;*RPM1 MICRO ERROR 06
                                ;*RPM1-READ PATH-SELF TEST
                                ;*M8953, M8960
                                ;*READ PATH IF. REGISTER 01 INCORRECT AFTER CLEAR
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 04
                                ;*RPM1 MICRO ERROR 07
                                ;*RPM1-READ PATH-SELF TEST
                                ;*M8953, M8960
                                ;*READ CHANNEL INTERFACE FAILURE
4455                            S
 (1)                            * ************************************************************************
                                TEST4:  TESTX   @4              ;INITIALIZE THE TEST
4455  3E  04          7.0                       MVI     A,@4            ;DEFINE THE TEST NUMBER
4457  CD  03  28      18.0                      CALL    TSET            ;SETUP THE TEST
                                ;%RPM1-READ PATH-SELF TEST
                                ;&M8953, M8960
445A  3E  10          7.0   TST04X: MVI     A,RDCLK
445C  D3  F0         10.0           OUT     CLKCTL          ;SET NORMAL CLOCKS
445E  3E  00          7.0           MVI     A,0             ;LOAD THE READ PATH ENABLE BIT
4460  D3  09         10.0           OUT     RPCTL           ;ENABLE THE READ PATH
4462  3E  08          7.0   RMCLP1: MVI     A,RMCTST        ;LOAD THE RMC SELF TEST COMMAND
4464  D3  0B         10.0           OUT     RCMD            ;ISSUE THE COMMAND
                                                             ;
4466  3E  02          7.0           MVI     A,@2            ;WAIT 10 MICRO SECONDS
4468  3D             4.0   TLOOP0: DCR     A
4469  C2  68  44     10.0           JNZ     TLOOP0
                                                             ;
446C  DB  02         10.0           IN      RSTAT           ;GET THE SELF TEST STATUS
446E  FE  43          7.0           CPI     RPOK            ;PASS THE SELF TEST?
4470  CA  78  44     10.0           JZ      RMCCN1          ;YES-CONTINUE
```

```
1688  4473                                          ERR     RMCLP1,RMCCN1
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4473  CD  09  28      18.0                            CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0007                                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4476  07                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4477  00                                              .BYTE
 (1)  4478  CD  15  28      18.0                    RMCCN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  447B  DA  62  44      10.0                            JC      RMCLP1          ;LOOP ADDRESS IF LOOP SPECIFIED
1689                                    ;>SELF DIAGNOSIS FAILURE
1690  447E  3E  18           7.0                  MVI     A,$18           ;SAVE THE EXPECTED DATA
1691  4480                               ROUT    EDATA
 (1)  4480  D3  95          10.0                            OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4482  7F               4.0                            MOV     A,A             ;RETRY LINK
1692  4483  DB  01          10.0                  IN      RPCHI           ;GET READ PATH REGISTER
1693  4485  E6  7E           7.0                  ANI     $7E             ;SAVE WANTED BITS
1694  4487                               ROUT    ADATA
 (1)  4487  D3  94          10.0                            OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4489  7F               4.0                            MOV     A,A             ;RETRY LINK
1695  448A  FE  78           7.0                  CPI     $78             ;DATA READY, STAT VALID,PREAM ERR AND BOP=1
1696  448C  CA  94  44       10.0                 JZ      RMCCN2          ;CLOCK STOPPED AND STATISTICS/NORMAL=0
1697  448F                                       ERRB    RMCLP1,RMCCN2
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  448F  CD  12  28      18.0                            CALL    ERRB            ;PROCESS ERROR - DO 2.3
 (1)       0008                                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4492  08                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4493  00                                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4494  CD  15  28      18.0                    RMCCN2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4497  DA  62  44      10.0                            JC      RMCLP1          ;LOOP ADDRESS IF LOOP SPECIFIED
1698                                    ;>READ PATH IF. REGISTER 01 INCORRECT AFTER SELF TEST
1699  449A  CD  19  4A      18.0                  CALL    RCCLR           ;ISSUE CLEAR ALL COMMAND
1700  449D  3E  03           7.0                  MVI     A,$03           ;SET STOP CLOCK AND STATISTICS/NORMAL
1701  449F  D3  09          10.0                  OUT     RPCTL
1702  44A1  3E  06           7.0                  MVI     A,$06
1703  44A3                               ROUT    EDATA
 (1)  44A3  D3  95          10.0                            OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44A5  7F               4.0                            MOV     A,A             ;RETRY LINK
1704  44A6  DB  01          10.0                  IN      RPCHI
1705  44A8  E6  66           7.0                  ANI     $66
1706  44AA  FE  06           7.0                  CPI     $06
1707  44AC  CA  B7  44       10.0                 JZ      RMCCN3
1708  44AF                               ROUT    ADATA
 (1)  44AF  D3  94          10.0                            OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44B1  7F               4.0                            MOV     A,A             ;RETRY LINK
```

```
1710  44B2                                        ERRB    RMCLP1,RMCCN3
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44B2  CD  12  28      18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0009                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44B5  09                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44B6  00                                            .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  44B7  CD  15  28      18.0                  RMCCN3::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44BA  DA  62  44      10.0                          JC      RMCLP1          ;LOOP ADDRESS IF LOOP SPECIFIED
1711                                        ;>READ PATH IF. REGISTER 01 INCORRECT AFTER CLEAR
1712  44BD  3E  00            7.0  RLOOP1: MVI     A,0
1713  44BF  D3  09           10.0          OUT     RPCTL           ;RE-ENABLE THE READ PATH CLOCK
1714  44C1  3E  0C            7.0          MVI     A,RCHTST        ;ISSUE THE READ CHANNEL
1715  44C3  D3  0B           10.0          OUT     RCMD            ;SELF TEST COMMAND
1716                                      ;
1717                                      ;PERFORM A 75 MICROSECOND TIMEOUT
1718                                      ;
1719  44C5  3E  0F            7.0          MVI     A,15            ;DELAY 75 MICROSECONDS
1720  44C7  3D                4.0  TLOOP4: DCR     A
1721  44C8  C2  C7  44       10.0          JNZ     TLOOP4
1722  44CB  DB  02           10.0          IN      RSTAT           ;INPUT THE READ PATH STATUS
1723  44CD  FE  46            7.0          CPI     RCHOK
1724  44CF  CA  D7  44       10.0          JZ      RMCCN4
1725  44D2                                ERR     RLOOP1,RMCCN4
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44D2  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000A                                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44D5  0A                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44D6  00                                    .BYTE
 (1)  44D7  CD  15  28      18.0          RMCCN4::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44DA  DA  BD  44      10.0                  JC      RLOOP1          ;LOOP ADDRESS IF LOOP SPECIFIED
1726                                        ;>READ CHANNEL INTERFACE FAILURE
1727  44DD                                      ENDTST  TST04X
 (1)                                      ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44DD                                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44DD  CD  06  28      18.0                        CALL    REQST
 (2)  44E0  00                                          .BYTE                   ;DATA PATTERN NUMBER
 (2)  44E1  00  00                                      .WORD                   ;SYSTEM "" COUNT
 (2)  44E3  00  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  44E5  00                                          .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  44E6  07                                          .BYTE   7               ;REQUEST CODE
 (1)  44E7  3A  9A  4F      13.0                LDA     ITERA           ;GET ITERATION COUNT
 (1)  44EA  3D                4.0                DCR     A               ;DOWNCOUNT
 (1)  44EB  32  9A  4F      13.0                STA     ITERA           ;SAVE COUNT
 (1)  44EE  F2  5A  44      10.0                JP      TST04X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 05 - READ CHANNEL 0 SELF TEST
44F1                            ST
 (1)                            ; ******************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*RPM1-CHANNEL 0 SELF TEST
44F1                            SP
 (1)                            ; ******************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  LOAD THE CHANNEL MASK (001)
                                ;*  CALL SUBROUTINE RCCLR
                                ;*  SET THE EXPECTED DATA STATE (0)
                                ;*  CALL SUBROUTINE RCCK
                                ;*  CALL SUBROUTINE RCSTRT
                                ;*  SET THE EXPECTED DATA STATE (001)
                                ;*  CALL SUBROUTINE RCCK
                                ;*ENDTST
44F1                            SE
 (1)                            ; ******************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*RPM1 MICRO TEST 05
                                ;*RPM1 MICRO ERROR 24
                                ;*RPM1-CHANNEL 0 SELF TEST
                                ;*M8950
                                ;*END MARK BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 05
                                ;*RPM1 MICRO ERROR 25
                                ;*RPM1-CHANNEL 0 SELF TEST
                                ;*M8950
                                ;*DATA BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 05
                                ;*RPM1 MICRO ERROR 26
                                ;*RPM1-CHANNEL 0 SELF TEST
                                ;*M8950
                                ;*POSTAMBLE BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 05
                                ;*RPM1 MICRO ERROR 27
                                ;*RPM1-CHANNEL 0 SELF TEST
                                ;*M8950
                                ;*ILLEGAL BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 05
                                ;*RPM1 MICRO ERROR 30
                                ;*RPM1-CHANNEL 0 SELF TEST
```

```
                                        ;*M8950
                                        ;*MARK TWO BIT INCORRECT
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 05
                                        ;*RPM1 MICRO ERROR 31
                                        ;*RPM1-CHANNEL 0 SELF TEST
                                        ;*M8950
                                        ;*DONE NOT BIT INCORRECT
                                        ;*EXPECTED = NNNN
44F1                                    $
 (1)                                    ; ************************************************************************************

                                        TEST05: TESTX   @05             ;INITIALIZE THE TEST
 (1) 44F1  3E  05          7.0                       MVI     A,@05       ;DEFINE THE TEST NUMBER
 (1) 44F3  CD  03  28     18.0                       CALL    TSET        ;SETUP THE TEST
                                        ;%RPM1-CHANNEL 0 SELF TEST
                                        ;&M8950, M8953

44F6  3E  01              7.0   TST05X: MVI     A,@001      ;LOAD THE CHANNEL MASK
44F8  32  4B  4B         13.0           STA     RCMSK
44FB  CD  19  4A         18.0           CALL    RCCLR       ;CALL SUBROUTINE RCCLR
44FE  3E  00              7.0           MVI     A,0
4500  32  4A  4B         13.0           STA     RCEDAT      ;SET THE EXPECTED DATA STATE
4503  CD  4B  4A         18.0           CALL    RCCK        ;CALL SUBROUTINE RCCK
4506  DA  F6  44         10.0           JC      TST05X
4509  CD  32  4A         18.0   TST05Y: CALL    RCSTRT      ;CALL SUBROUTINE RCSTRT
450C  3E  01              7.0           MVI     A,@001
450E  32  4A  4B         13.0           STA     RCEDAT      ;SET THE EXPECTED DATA STATE
4511  CD  4B  4A         18.0           CALL    RCCK        ;CALL SUBROUTINE RCCK
4514  DA  09  45         10.0           JC      TST05Y
4517                                    ENDTST  TST05X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2) 4517                               REQ     7           ;FAKE CALL TO KEEP TEST ALIVE
 (2) 4517  CD  06  28     18.0                       CALL    REQST
 (2) 451A  00                                        .BYTE                   ;DATA PATTERN NUMBER
 (2) 451B  00  00                                    .WORD                   ;SYSTEM '"' COUNT
 (2) 451D  00  00                                    .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 451F  00                                        .BYTE                   ;DATA COMPARE FLAG IF -1
 (2) 4520  07                                        .BYTE   7               ;REQUEST CODE
 (1) 4521  3A  9A  4F     13.0           LDA     ITERA       ;GET ITERATION COUNT
 (1) 4524  3D               4.0           DCR     A           ;DOWNCOUNT
 (1) 4525  32  9A  4F     13.0           STA     ITERA       ;SAVE COUNT
 (1) 4528  F2  F6  44     10.0           JP      TST05X      ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 06 - READ CHANNEL 1 SELF TEST
452B                            ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*RPM1-CHANNEL 1 SELF TEST
452B                            SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  LOAD THE CHANNEL MASK (002)
                                ;*  CALL SUBROUTINE RCCLR
                                ;*  SET THE EXPECTED DATA STATE (0)
                                ;*  CALL SUBROUTINE RCCK
                                ;*  CALL SUBROUTINE RCSTRT
                                ;*  SET THE EXPECTED DATA STATE (002)
                                ;*  CALL SUBROUTINE RCCK
                                ;*ENDTST
452B                            SE
 (1)                            ;*********************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*RPM1 MICRO TEST 06
                                ;*RPM1 MICRO ERROR 24
                                ;*RPM1-CHANNEL 1 SELF TEST
                                ;*M8950
                                ;*END MARK BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 06
                                ;*RPM1 MICRO ERROR 25
                                ;*RPM1-CHANNEL 1 SELF TEST
                                ;*M8950
                                ;*DATA BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 06
                                ;*RPM1 MICRO ERROR 26
                                ;*RPM1-CHANNEL 1 SELF TEST
                                ;*M8950
                                ;*POSTAMBLE BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 06
                                ;*RPM1 MICRO ERROR 27
                                ;*RPM1-CHANNEL 1 SELF TEST
                                ;*M8950
                                ;*ILLEGAL BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 06
                                ;*RPM1 MICRO ERROR 30
                                ;*RPM1-CHANNEL 1 SELF TEST
```

```
                                        ;*M8950
                                        ;*MARK TWO BIT INCORRECT
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 06
                                        ;*RPM1 MICRO ERROR 31
                                        ;*RPM1-CHANNEL 1 SELF TEST
                                        ;*M8950
                                        ;*DONE NOT BIT INCORRECT
                                        ;*EXPECTED = NNNN
452B                                    $
 (1)                                    ; ***************************************************************************

                                        TEST06: TESTX   @06             ;INITIALIZE THE TEST
 (1) 452B 3E 06          7.0                            MVI     A,@06           ;DEFINE THE TEST NUMBER
 (1) 452D CD 03 28      18.0                            CALL    TSET            ;SETUP THE TEST
                                        ;%RPM1-CHANNEL 1 SELF TEST
                                        ;&M8950, M8953

4530 3E 02              7.0     TST06X: MVI     A,@002          ;LOAD THE CHANNEL MASK
4532 32 4B 4B          13.0             STA     RCMSK
4535 CD 19 4A          18.0             CALL    RCCLR           ;CALL SUBROUTINE RCCLR
4538 3E 00              7.0             MVI     A,0
453A 32 4A 4B          13.0             STA     RCEDAT          ;SET THE EXPECTED DATA STATE
453D CD 4B 4A          18.0             CALL    RCCK            ;CALL SUBROUTINE RCCK
4540 DA 30 45          10.0             JC      TST06X
4543 CD 32 4A          18.0     TST06Y: CALL    RCSTRT          ;CALL SUBROUTINE RCSTRT
4546 3E 02              7.0             MVI     A,@002
4548 32 4A 4B          13.0             STA     RCEDAT          ;SET THE EXPECTED DATA STATE
454B CD 4B 4A          18.0             CALL    RCCK            ;CALL SUBROUTINE RCCK
454E DA 43 45          10.0             JC      TST06Y
4551                                    ENDTST  TST06X
 (1)                            ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2) 4551                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2) 4551 CD 06 28      18.0                            CALL    REQST
 (2) 4554 00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2) 4555 00 00                                         .WORD                   ;SYSTEM '*' COUNT
 (2) 4557 00 00                                         .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 4559 00                                            .BYTE                   ;DATA COMPARE FLAG IF =1
 (2) 455A 07                                            .BYTE   7               ;REQUEST CODE
 (1) 455B 3A 9A 4F      13.0             LDA     ITERA           ;GET ITERATION COUNT
 (1) 455E 3D             4.0             DCR     A               ;DOWNCOUNT
 (1) 455F 32 9A 4F      13.0             STA     ITERA           ;SAVE COUNT
 (1) 4562 F2 30 45      10.0             JP      TST06X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 07 - READ CHANNEL 2 SELF TEST
4565                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*RPM1-CHANNEL 2 SELF TEST
4565                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  LOAD THE CHANNEL MASK (004)
                                ;*  CALL SUBROUTINE RCCLR
                                ;*  SET THE EXPECTED DATA STATE (0)
                                ;*  CALL SUBROUTINE RCCK
                                ;*  CALL SUBROUTINE RCSTRT
                                ;*  SET THE EXPECTED DATA STATE (004)
                                ;*  CALL SUBROUTINE RCCK
                                ;*ENDTST
4565                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*RPM1 MICRO TEST 07
                                ;*RPM1 MICRO ERROR 24
                                ;*RPM1-CHANNEL 2 SELF TEST
                                ;*M8950
                                ;*END MARK BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 07
                                ;*RPM1 MICRO ERROR 25
                                ;*RPM1-CHANNEL 2 SELF TEST
                                ;*M8950
                                ;*DATA BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 07
                                ;*RPM1 MICRO ERROR 26
                                ;*RPM1-CHANNEL 2 SELF TEST
                                ;*M8950
                                ;*POSTAMBLE BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 07
                                ;*RPM1 MICRO ERROR 27
                                ;*RPM1-CHANNEL 2 SELF TEST
                                ;*M8950
                                ;*ILLEGAL BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 07
                                ;*RPM1 MICRO ERROR 30
                                ;*RPM1-CHANNEL 2 SELF TEST
```

```
1924                                    ;*M8950
1925                                    ;*MARK TWO BIT INCORRECT
1926                                    ;*EXPECTED = NNNN
1927                                    ;*
1928                                    ;*RPM1 MICRO TEST 07
1929                                    ;*RPM1 MICRO ERROR 31
1930                                    ;*RPM1-CHANNEL 2 SELF TEST
1931                                    ;*M8950
1932                                    ;*DONE NOT BIT INCORRECT
1933                                    ;*EXPECTED = NNNN
1934  4565                              $
 (1)                                    ; ***************************************************************************
1935
1936  4565                              TEST07: TESTX   @07             ;INITIALIZE THE TEST
 (1)  4565  3E  07          7.0                        MVI     A,@07       ;DEFINE THE TEST NUMBER
 (1)  4567  CD  03  28     18.0                        CALL    TSET        ;SETUP THE TEST
1937                                    ;%RPM1-CHANNEL 2 SELF TEST
1938                                    ;&M8950, M8953
1939
1940  456A  3E  04           7.0  TST07X: MVI     A,@004          ;LOAD THE CHANNEL MASK
1941  456C  32  4B  4B      13.0          STA     RCMSK
1942  456F  CD  19  4A      18.0          CALL    RCCLR           ;CALL SUBROUTINE RCCLR
1943  4572  3E  00           7.0          MVI     A,0
1944  4574  32  4A  4B      13.0          STA     RCEDAT          ;SET THE EXPECTED DATA STATE
1945  4577  CD  4B  4A      18.0          CALL    RCCK            ;CALL SUBROUTINE RCCK
1946  457A  DA  6A  45      10.0          JC      TST07X
1947  457D  CD  32  4A      18.0  TST07Y: CALL    RCSTRT          ;CALL SUBROUTINE RCSTRT
1948  4580  3E  04           7.0          MVI     A,@004
1949  4582  32  4A  4B      13.0          STA     RCEDAT          ;SET THE EXPECTED DATA STATE
1950  4585  CD  4B  4A      18.0          CALL    RCCK            ;CALL SUBROUTINE RCCK
1951  4588  DA  7D  45      10.0          JC      TST07Y
1952  458B                              ENDTST  TST07X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  458B                          REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  458B  CD  06  28     18.0                        CALL    REQST
 (2)  458E  00                                         .BYTE                   ;DATA PATTERN NUMBER
 (2)  458F  00  00                                     .WORD                   ;SYSTEM "" COUNT
 (2)  4591  00  00                                     .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4593  00                                         .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4594  07                                         .BYTE   7               ;REQUEST CODE
 (1)  4595  3A  9A  4F      13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4598  3D               4.0          DCR     A               ;DOWNCOUNT
 (1)  4599  32  9A  4F      13.0          STA     ITERA           ;SAVE COUNT
 (1)  459C  F2  6A  45      10.0          JP      TST07X          ;DO TEST UNTIL TILL = 0
```

```
1954
1955  459F                              .SBTTL  TEST 10 - READ CHANNEL 3 SELF TEST
1955  459F                              ST
 (1)                                    ; ***************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
1956                                    ;*RPM1-CHANNEL 3 SELF TEST
1957  459F                              SP
 (1)                                    ; ***************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
1958                                    ;*BGNTST
1959                                    ;*  LOAD THE CHANNEL MASK (010)
1960                                    ;*  CALL SUBROUTINE RCCLR
1961                                    ;*  SET THE EXPECTED DATA STATE (0)
1962                                    ;*  CALL SUBROUTINE RCCK
1963                                    ;*  CALL SUBROUTINE RCSTRT
1964                                    ;*  SET THE EXPECTED DATA STATE (010)
1965                                    ;*  CALL SUBROUTINE RCCK
1966                                    ;*ENDTST
1967  459F                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1968                                    ;*RPM1 MICRO TEST 10
1969                                    ;*RPM1 MICRO ERROR 24
1970                                    ;*RPM1-CHANNEL 3 SELF TEST
1971                                    ;*M8950
1972                                    ;*END MARK BIT INCORRECT
1973                                    ;*EXPECTED = NNNN
1974                                    ;*
1975                                    ;*RPM1 MICRO TEST 10
1976                                    ;*RPM1 MICRO ERROR 25
1977                                    ;*RPM1-CHANNEL 3 SELF TEST
1978                                    ;*M8950
1979                                    ;*DATA BIT INCORRECT
1980                                    ;*EXPECTED = NNNN
1981                                    ;*
1982                                    ;*RPM1 MICRO TEST 10
1983                                    ;*RPM1 MICRO ERROR 26
1984                                    ;*RPM1-CHANNEL 3 SELF TEST
1985                                    ;*M8950
1986                                    ;*POSTAMBLE BIT INCORRECT
1987                                    ;*EXPECTED = NNNN
1988                                    ;*
1989                                    ;*RPM1 MICRO TEST 10
1990                                    ;*RPM1 MICRO ERROR 27
1991                                    ;*RPM1-CHANNEL 3 SELF TEST
1992                                    ;*M8950
1993                                    ;*ILLEGAL BIT INCORRECT
1994                                    ;*EXPECTED = NNNN
1995                                    ;*
1996                                    ;*RPM1 MICRO TEST 10
1997                                    ;*RPM1 MICRO ERROR 30
1998                                    ;*RPM1-CHANNEL 3 SELF TEST
```

```
1999                                    ;*M8950
2000                                    ;*MARK TWO BIT INCORRECT
2001                                    ;*EXPECTED = NNNN
2002                                    ;*
2003                                    ;*RPM1 MICRO TEST 10
2004                                    ;*RPM1 MICRO ERROR 31
2005                                    ;*RPM1-CHANNEL 3 SELF TEST
2006                                    ;*M8950
2007                                    ;*DONE NOT BIT INCORRECT
2008                                    ;*EXPECTED = NNNN
2009  459F                              S
 (1)                                    ; ************************************************************************************
2010
2011  459F                              TEST10: TESTX   @10             ;INITIALIZE THE TEST
 (1)  459F  3E  08          7.0                        MVI     A,@10       ;DEFINE THE TEST NUMBER
 (1)  45A1  CD  03  28     18.0                        CALL    TSET        ;SETUP THE TEST
2012                                    ;%RPM1-CHANNEL 3 SELF TEST
2013                                    ;&M8950, M8953
2014
2015  45A4  3E  08          7.0  TST10X: MVI     A,@010          ;LOAD THE CHANNEL MASK
2016  45A6  32  4B  4B       13.0         STA     RCMSK
2017  45A9  CD  19  4A       18.0         CALL    RCCLR           ;CALL SUBROUTINE RCCLR
2018  45AC  3E  00           7.0         MVI     A,0
2019  45AE  32  4A  4B       13.0         STA     RCEDAT          ;SET THE EXPECTED DATA STATE
2020  45B1  CD  4B  4A       18.0         CALL    RCCK            ;CALL SUBROUTINE RCCK
2021  45B4  DA  A4  45       10.0         JC      TST10X
2022  45B7  CD  32  4A       18.0  TST10Y: CALL    RCSTRT          ;CALL SUBROUTINE RCSTRT
2023  45BA  3E  08           7.0         MVI     A,@010
2024  45BC  32  4A  4B       13.0         STA     RCEDAT          ;SET THE EXPECTED DATA STATE
2025  45BF  CD  4B  4A       18.0         CALL    RCCK            ;CALL SUBROUTINE RCCK
2026  45C2  DA  B7  45       10.0         JC      TST10Y
2027  45C5                              ENDTST  TST10X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  45C5                          REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  45C5  CD  06  28     18.0                        CALL    REQST
 (2)  45C8  00                                        .BYTE                   ;DATA PATTERN NUMBER
 (2)  45C9  00  00                                    .WORD                   ;SYSTEM "" COUNT
 (2)  45CB  00  00                                    .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  45CD  00                                        .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  45CE  07                                        .BYTE   7               ;REQUEST CODE
 (1)  45CF  3A  9A  4F     13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1)  45D2  3D              4.0         DCR     A               ;DOWNCOUNT
 (1)  45D3  32  9A  4F     13.0         STA     ITERA           ;SAVE COUNT
 (1)  45D6  F2  A4  45     10.0         JP      TST10X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 11 - READ CHANNEL 4 SELF TEST
45D9                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*RPM1-CHANNEL 4 SELF TEST
45D9                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  LOAD THE CHANNEL MASK (020)
                                ;*  CALL SUBROUTINE RCCLR
                                ;*  SET THE EXPECTED DATA STATE (0)
                                ;*  CALL SUBROUTINE RCCK
                                ;*  CALL SUBROUTINE RCSTRT
                                ;*  SET THE EXPECTED DATA STATE (020)
                                ;*  CALL SUBROUTINE RCCK
                                ;*ENDTST
45D9                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*RPM1 MICRO TEST 11
                                ;*RPM1 MICRO ERROR 24
                                ;*RPM1-CHANNEL 4 SELF TEST
                                ;*M8950
                                ;*END MARK BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 11
                                ;*RPM1 MICRO ERROR 25
                                ;*RPM1-CHANNEL 4 SELF TEST
                                ;*M8950
                                ;*DATA BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 11
                                ;*RPM1 MICRO ERROR 26
                                ;*RPM1-CHANNEL 4 SELF TEST
                                ;*M8950
                                ;*POSTAMBLE BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 11
                                ;*RPM1 MICRO ERROR 27
                                ;*RPM1-CHANNEL 4 SELF TEST
                                ;*M8950
                                ;*ILLEGAL BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 11
                                ;*RPM1 MICRO ERROR 30
                                ;*RPM1-CHANNEL 4 SELF TEST
```

```
                                        ;*M8950
                                        ;*MARK TWO BIT INCORRECT
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 11
                                        ;*RPM1 MICRO ERROR 31
                                        ;*RPM1-CHANNEL 4 SELF TEST
                                        ;*M8950
                                        ;*DONE NOT BIT INCORRECT
                                        ;*EXPECTED = NNNN
45D9                                    $
 (1)                                    ; ************************************************************************
                                        TEST11: TESTX   @11             ;INITIALIZE THE TEST
 (1) 45D9  3E  09          7.0                       MVI     A,@11       ;DEFINE THE TEST NUMBER
 (1) 45DB  CD  03  28     18.0                       CALL    TSET        ;SETUP THE TEST
                                        ;%RPM1-CHANNEL 4 SELF TEST
                                        ;&M8950, M8953

45DE  3E  10              7.0   TST11X: MVI     A,@020      ;LOAD THE CHANNEL MASK
45E0  32  4B  4B         13.0           STA     RCMSK
45E3  CD  19  4A         18.0           CALL    RCCLR       ;CALL SUBROUTINE RCCLR
45E6  3E  00              7.0           MVI     A,0
45E8  32  4A  4B         13.0           STA     RCEDAT      ;SET THE EXPECTED DATA STATE
45EB  CD  4B  4A         18.0           CALL    RCCK        ;CALL SUBROUTINE RCCK
45EE  DA  DE  45         10.0           JC      TST11X
45F1  CD  32  4A         18.0   TST11Y: CALL    RCSTRT      ;CALL SUBROUTINE RCSTRT
45F4  3E  10              7.0           MVI     A,@020
45F6  32  4A  4B         13.0           STA     RCEDAT      ;SET THE EXPECTED DATA STATE
45F9  CD  4B  4A         18.0           CALL    RCCK        ;CALL SUBROUTINE RCCK
45FC  DA  F1  45         10.0           JC      TST11Y
45FF                                    ENDTST  TST11X
 (1)                            ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2) 45FF                               REQ     7           ;FAKE CALL TO KEEP TEST ALIVE
 (2) 45FF  CD  06  28     18.0                       CALL    REQST
 (2) 4602  00                                        .BYTE                   ;DATA PATTERN NUMBER
 (2) 4603  00  00                                    .WORD                   ;SYSTEM '*' COUNT
 (2) 4605  00  00                                    .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 4607  00                                        .BYTE                   ;DATA COMPARE FLAG IF =1
 (2) 4608  07                                        .BYTE   7               ;REQUEST CODE
 (1) 4609  3A  9A  4F     13.0           LDA     ITERA       ;GET ITERATION COUNT
 (1) 460C  3D              4.0           DCR     A           ;DOWNCOUNT
 (1) 460D  32  9A  4F     13.0           STA     ITERA       ;SAVE COUNT
 (1) 4610  F2  DE  45     10.0           JP      TST11X      ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 12 - READ CHANNEL 5 SELF TEST
4613                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM1-CHANNEL 5 SELF TEST
4613                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  LOAD THE CHANNEL MASK (040)
                                ;*  CALL SUBROUTINE RCCLR
                                ;*  SET THE EXPECTED DATA STATE (0)
                                ;*  CALL SUBROUTINE RCCK
                                ;*  CALL SUBROUTINE RCSTRT
                                ;*  SET THE EXPECTED DATA STATE (040)
                                ;*  CALL SUBROUTINE RCCK
                                ;*ENDTST
4613                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM1 MICRO TEST 12
                                ;*RPM1 MICRO ERROR 24
                                ;*RPM1-CHANNEL 5 SELF TEST
                                ;*M8950
                                ;*END MARK BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 12
                                ;*RPM1 MICRO ERROR 25
                                ;*RPM1-CHANNEL 5 SELF TEST
                                ;*M8950
                                ;*DATA BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 12
                                ;*RPM1 MICRO ERROR 26
                                ;*RPM1-CHANNEL 5 SELF TEST
                                ;*M8950
                                ;*POSTAMBLE BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 12
                                ;*RPM1 MICRO ERROR 27
                                ;*RPM1-CHANNEL 5 SELF TEST
                                ;*M8950
                                ;*ILLEGAL BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 12
                                ;*RPM1 MICRO ERROR 30
                                ;*RPM1-CHANNEL 5 SELF TEST
```

```
                                        ;*M8950
                                        ;*MARK TWO BIT INCORRECT
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 12
                                        ;*RPM1 MICRO ERROR 31
                                        ;*RPM1-CHANNEL 5 SELF TEST
                                        ;*M8950
                                        ;*DONE NOT BIT INCORRECT
                                        ;*EXPECTED = NNNN
4613                                    $
 (1)                                    ; ************************************************************************
                                        TEST12: TESTX   @12             ;INITIALIZE THE TEST
 (1) 4613  3E  0A           7.0                  MVI     A,@12           ;DEFINE THE TEST NUMBER
 (1) 4615  CD  03  28      18.0                  CALL    TSET            ;SETUP THE TEST
                                        ;%RPM1-CHANNEL 5 SELF TEST
                                        ;&M8950, M8953

4618  3E  20                7.0  TST12X: MVI     A,@040          ;LOAD THE CHANNEL MASK
461A  32  4B  4B           13.0         STA     RCMSK
461D  CD  19  4A           18.0         CALL    RCCLR           ;CALL SUBROUTINE RCCLR
4620  3E  00                 7.0         MVI     A,0
4622  32  4A  4B           13.0         STA     RCEDAT          ;SET THE EXPECTED DATA STATE
4625  CD  4B  4A           18.0         CALL    RCCK            ;CALL SUBROUTINE RCCK
4628  DA  18  46           10.0         JC      TST12X
462B  CD  32  4A           18.0  TST12Y: CALL    RCSTRT          ;CALL SUBROUTINE RCSTRT
462E  3E  20                 7.0         MVI     A,@040
4630  32  4A  4B           13.0         STA     RCEDAT          ;SET THE EXPECTED DATA STATE
4633  CD  4B  4A           18.0         CALL    RCCK            ;CALL SUBROUTINE RCCK
4636  DA  2B  46           10.0         JC      TST12Y
4639                                    ENDTST  TST12X
 (1)                            ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2) 4639                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2) 4639  CD  06  28      18.0                  CALL    REQST
 (2) 463C  00                                    .BYTE                   ;DATA PATTERN NUMBER
 (2) 463D  00  00                                .WORD                   ;SYSTEM '"' COUNT
 (2) 463F  00  00                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 4641  00                                    .BYTE                   ;DATA COMPARE FLAG IF =1
 (2) 4642  07                                    .BYTE   7               ;REQUEST CODE
 (1) 4643  3A  9A  4F      13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1) 4646  3D                4.0         DCR     A               ;DOWNCOUNT
 (1) 4647  32  9A  4F      13.0         STA     ITERA           ;SAVE COUNT
 (1) 464A  F2  18  46      10.0         JP      TST12X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 13 - READ CHANNEL 6 SELF TEST
464D                            ST
                                ; ******************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM1-CHANNEL 6 SELF TEST
464D                            SP
                                ; ******************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  LOAD THE CHANNEL MASK (100)
                                ;*  CALL SUBROUTINE RCCLR
                                ;*  SET THE EXPECTED DATA STATE (0)
                                ;*  CALL SUBROUTINE RCCK
                                ;*  CALL SUBROUTINE RCSTRT
                                ;*  SET THE EXPECTED DATA STATE (100)
                                ;*  CALL SUBROUTINE RCCK
                                ;*ENDTST
464D                            SE
                                ; ******************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM1 MICRO TEST 13
                                ;*RPM1 MICRO ERROR 24
                                ;*RPM1-CHANNEL 6 SELF TEST
                                ;*M8950
                                ;*END MARK BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 13
                                ;*RPM1 MICRO ERROR 25
                                ;*RPM1-CHANNEL 6 SELF TEST
                                ;*M8950
                                ;*DATA BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 13
                                ;*RPM1 MICRO ERROR 26
                                ;*RPM1-CHANNEL 6 SELF TEST
                                ;*M8950
                                ;*POSTAMBLE BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 13
                                ;*RPM1 MICRO ERROR 27
                                ;*RPM1-CHANNEL 6 SELF TEST
                                ;*M8950
                                ;*ILLEGAL BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 13
                                ;*RPM1 MICRO ERROR 30
                                ;*RPM1-CHANNEL 6 SELF TEST
```

```
                                        ;*M8950
                                        ;*MARK TWO BIT INCORRECT
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 13
                                        ;*RPM1 MICRO ERROR 31
                                        ;*RPM1-CHANNEL 6 SELF TEST
                                        ;*M8950
                                        ;*DONE NOT BIT INCORRECT
                                        ;*EXPECTED = NNNN
2234  464D                              S
 (1)                                    ; ************************************************************************************

2235
2236  464D                              TEST13: TESTX   @13             ;INITIALIZE THE TEST
 (1)  464D  3E  0B          7.0                        MVI     A,@13       ;DEFINE THE TEST NUMBER
 (1)  464F  CD  03  28     18.0                        CALL    TSET        ;SETUP THE TEST
2237                                    ;%RPM1-CHANNEL 6 SELF TEST
2238                                    ;&M8950, M8953
2239
2240  4652  3E  40           7.0  TST13X: MVI     A,@100          ;LOAD THE CHANNEL MASK
2241  4654  32  4B  4B       13.0         STA     RCMSK
2242  4657  CD  19  4A       18.0         CALL    RCCLR           ;CALL SUBROUTINE RCCLR
2243  465A  3E  00           7.0         MVI     A,0
2244  465C  32  4A  4B       13.0         STA     RCEDAT          ;SET THE EXPECTED DATA STATE
2245  465F  CD  4B  4A       18.0         CALL    RCCK            ;CALL SUBROUTINE RCCK
2246  4662  DA  52  46       10.0         JC      TST13X
2247  4665  CD  32  4A       18.0  TST13Y: CALL    RCSTRT          ;CALL SUBROUTINE RCSTRT
2248  4668  3E  40           7.0         MVI     A,@100
2249  466A  32  4A  4B       13.0         STA     RCEDAT          ;SET THE EXPECTED DATA STATE
2250  466D  CD  4B  4A       18.0         CALL    RCCK            ;CALL SUBROUTINE RCCK
2251  4670  DA  65  46       10.0         JC      TST13Y
2252  4673                               ENDTST  TST13X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4673                          REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4673  CD  06  28     18.0                        CALL    REQST
 (2)  4676  00                                        .BYTE           ;DATA PATTERN NUMBER
 (2)  4677  00  00                                    .WORD           ;SYSTEM '*' COUNT
 (2)  4679  00  00                                    .WORD           ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  467B  00                                        .BYTE           ;DATA COMPARE FLAG IF =1
 (2)  467C  07                                        .BYTE   7       ;REQUEST CODE
 (1)  467D  3A  9A  4F     13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1)  4680  3D              4.0         DCR     A               ;DOWNCOUNT
 (1)  4681  32  9A  4F     13.0         STA     ITERA           ;SAVE COUNT
 (1)  4684  F2  52  46     10.0         JP      TST13X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 14 - READ CHANNEL 7 SELF TEST
4687                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM1-CHANNEL 7 SELF TEST
4687                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  LOAD THE CHANNEL MASK (200)
                                ;*  CALL SUBROUTINE RCCLR
                                ;*  SET THE EXPECTED DATA STATE (0)
                                ;*  CALL SUBROUTINE RCCK
                                ;*  CALL SUBROUTINE RCSTRT
                                ;*  SET THE EXPECTED DATA STATE (200)
                                ;*  CALL SUBROUTINE RCCK
                                ;*ENDTST
4687                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM1 MICRO TEST 14
                                ;*RPM1 MICRO ERROR 24
                                ;*RPM1-CHANNEL 7 SELF TEST
                                ;*M8950
                                ;*END MARK BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 14
                                ;*RPM1 MICRO ERROR 25
                                ;*RPM1-CHANNEL 7 SELF TEST
                                ;*M8950
                                ;*DATA BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 14
                                ;*RPM1 MICRO ERROR 26
                                ;*RPM1-CHANNEL 7 SELF TEST
                                ;*M8950
                                ;*POSTAMBLE BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 14
                                ;*RPM1 MICRO ERROR 27
                                ;*RPM1-CHANNEL 7 SELF TEST
                                ;*M8950
                                ;*ILLEGAL BIT INCORRECT
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 14
                                ;*RPM1 MICRO ERROR 30
                                ;*RPM1-CHANNEL 7 SELF TEST
```

```
                                        ;*M8950
                                        ;*MARK TWO BIT INCORRECT
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 14
                                        ;*RPM1 MICRO ERROR 31
                                        ;*RPM1-CHANNEL 7 SELF TEST
                                        ;*M8950
                                        ;*DONE NOT BIT INCORRECT
                                        ;*EXPECTED = NNNN
2309  4687                              $
 (1)                                    ; ************************************************************************************

2311  4687                              TEST14: TESTX   @14              ;INITIALIZE THE TEST
 (1)  4687  3E  0C            7.0                      MVI     A,@14      ;DEFINE THE TEST NUMBER
 (1)  4689  CD  03  28       18.0                      CALL    TSET       ;SETUP THE TEST
2312                                    ;%RPM1-CHANNEL 7 SELF TEST
2313                                    ;&M8950, M8953
2314
2315  468C  3E  80               7.0   TST14X: MVI     A,@200           ;LOAD THE CHANNEL MASK
2316  468E  32  4B  4B          13.0            STA     RCMSK
2317  4691  CD  19  4A          18.0            CALL    RCCLR            ;CALL SUBROUTINE RCCLR
2318  4694  3E  00               7.0            MVI     A,0
2319  4696  32  4A  4B          13.0            STA     RCEDAT           ;SET THE EXPECTED DATA STATE
2320  4699  CD  4B  4A          18.0            CALL    RCCK             ;CALL SUBROUTINE RCCK
2321  469C  DA  8C  46          10.0            JC      TST14X
2322  469F  CD  32  4A          18.0   TST14Y: CALL    RCSTRT           ;CALL SUBROUTINE RCSTRT
2323  46A2  3E  80               7.0            MVI     A,@200
2324  46A4  32  4A  4B          13.0            STA     RCEDAT           ;SET THE EXPECTED DATA STATE
2325  46A7  CD  4B  4A          18.0            CALL    RCCK             ;CALL SUBROUTINE RCCK
2326  46AA  DA  9F  46          10.0            JC      TST14Y
2327  46AD                                    ENDTST  TST14X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  46AD                                      REQ     7                ;FAKE CALL TO KEEP TEST ALIVE
 (2)  46AD  CD  06  28       18.0                              CALL    REQST
 (2)  46B0  00                                                 .BYTE                ;DATA PATTERN NUMBER
 (2)  46B1  00  00                                             .WORD                ;SYSTEM "" COUNT
 (2)  46B3  00  00                                             .WORD                ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  46B5  00                                                 .BYTE                ;DATA COMPARE FLAG IF -1
 (2)  46B6  07                                                 .BYTE   7            ;REQUEST CODE
 (1)  46B7  3A  9A  4F      13.0            LDA     ITERA            ;GET ITERATION COUNT
 (1)  46BA  3D               4.0            DCR     A                ;DOWNCOUNT
 (1)  46BB  32  9A  4F      13.0            STA     ITERA            ;SAVE COUNT
 (1)  46BE  F2  9C  46      10.0            JP      TST14X           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 15 - READ CHANNEL P SELF TEST
2330  46C1                      ST
 (1)                            ; ******************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
2331                            ;*RPM1-CHANNEL P SELF TEST
2332  46C1                      SP
 (1)                            ; ******************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
2333                            ;*BGNTST
2334                            ;*  LOAD THE CHANNEL MASK (001)
2335                            ;*  CALL SUBROUTINE RCCLR
2336                            ;*  SET THE EXPECTED DATA STATE (0)
2337                            ;*  CALL SUBROUTINE RCCKP
2338                            ;*  CALL SUBROUTINE RCSTRT
2339                            ;*  SET THE EXPECTED DATA STATE (001)
2340                            ;*  CALL SUBROUTINE RCCKP
2341                            ;*ENDTST
2342  46C1                      SE
 (1)                            ; ******************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
2343                            ;*RPM1 MICRO TEST 15
2344                            ;*RPM1 MICRO ERROR 24
2345                            ;*RPM1-CHANNEL P SELF TEST
2346                            ;*M8950
2347                            ;*END MARK BIT INCORRECT
2348                            ;*EXPECTED = NNNN
2349                            ;*
2350                            ;*RPM1 MICRO TEST 15
2351                            ;*RPM1 MICRO ERROR 25
2352                            ;*RPM1-CHANNEL P SELF TEST
2353                            ;*M8950
2354                            ;*DATA BIT INCORRECT
2355                            ;*EXPECTED = NNNN
2356                            ;*
2357                            ;*RPM1 MICRO TEST 15
2358                            ;*RPM1 MICRO ERROR 26
2359                            ;*RPM1-CHANNEL P SELF TEST
2360                            ;*M8950
2361                            ;*POSTAMBLE BIT INCORRECT
2362                            ;*EXPECTED = NNNN
2363                            ;*
2364                            ;*RPM1 MICRO TEST 15
2365                            ;*RPM1 MICRO ERROR 27
2366                            ;*RPM1-CHANNEL P SELF TEST
2367                            ;*M8950
2368                            ;*ILLEGAL BIT INCORRECT
2369                            ;*EXPECTED = NNNN
2370                            ;*
2371                            ;*RPM1 MICRO TEST 15
2372                            ;*RPM1 MICRO ERROR 30
2373                            ;*RPM1-CHANNEL P SELF TEST
```

```
                                        ;*M8950
                                        ;*MARK TWO BIT INCORRECT
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 15
                                        ;*RPM1 MICRO ERROR 31
                                        ;*RPM1-CHANNEL P SELF TEST
                                        ;*M8950
                                        ;*DONE NOT BIT INCORRECT
                                        ;*EXPECTED = NNNN
46C1                                    S
 (1)                                    ; ************************************************************************************

46C1                                    TEST15: TESTX   @15             ;INITIALIZE THE TEST
 (1) 46C1  3E  0D          7.0                           MVI     A,@15           ;DEFINE THE TEST NUMBER
 (1) 46C3  CD  03  28     18.0                           CALL    TSET            ;SETUP THE TEST
                                        ;%RPM1-CHANNEL P SELF TEST
                                        ;&M8950, M8953

46C6  3E  01              7.0   TST15X: MVI     A,@001          ;LOAD THE CHANNEL MASK
46C8  32  4B  4B         13.0           STA     RCMSK
46CB  CD  19  4A         18.0           CALL    RCCLR           ;CALL SUBROUTINE RCCLR
46CE  3E  00              7.0           MVI     A,0
46D0  32  4A  4B         13.0           STA     RCEDAT          ;SET THE EXPECTED DATA STATE
46D3  CD  6C  4A         18.0           CALL    RCCKP           ;CALL SUBROUTINE RCCK
46D6  DA  C6  46         10.0           JC      TST15X
46D9  CD  32  4A         18.0   TST15Y: CALL    RCSTRT          ;CALL SUBROUTINE RCSTRT
46DC  3E  01              7.0           MVI     A,@001
46DE  32  4A  4B         13.0           STA     RCEDAT          ;SET THE EXPECTED DATA STATE
46E1  CD  6C  4A         18.0           CALL    RCCKP           ;CALL SUBROUTINE RCCK
46E4  DA  D9  46         10.0           JC      TST15Y
46E7                                    ENDTST  TST15X
 (1)                            ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2) 46E7                               REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2) 46E7  CD  06  28     18.0                           CALL    REQST
 (2) 46EA  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2) 46EB  00  00                                        .WORD                   ;SYSTEM '"' COUNT
 (2) 46ED  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2) 46EF  00                                            .BYTE                   ;DATA COMPARE FLAG IF =1
 (2) 46F0  07                                            .BYTE   7               ;REQUEST CODE
 (1) 46F1  3A  9A  4F     13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1) 46F4  3D               4.0           DCR     A               ;DOWNCOUNT
 (1) 46F5  32  9A  4F     13.0           STA     ITERA           ;SAVE COUNT
 (1) 46F8  F2  C6  46     10.0           JP      TST15X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 16 - READ CHANNEL AMTIE TEST - FROM TU PORT #0
46FB                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
46FB                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE AMTIE LINES FROM TAPE UNIT PORT 0
                                ;*THROUGH THE READ CHANNEL BOARDS AND TO THE READ PATH AMTIE REGISTER.
                                ;*256 AMTIE PATTERNS FROM 000-377(8) ARE USED.
46FB                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      IF TU PORT 0 NOT SELECTED BY THE USER
                                ;*          THEN-NEXT TEST
                                ;*          ELSE-CONTINUE
                                ;*      ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD TU PORT #0 SELECT CODE
                                ;*    CALL SUBROUTINE RDCTL
                                ;*ENDTST
46FB                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM1 MICRO TEST 16
                                ;*RPM1 MICRO ERROR 10
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
                                ;*M8950, M8953, M8955'S
                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*RPM1 MICRO TEST 16
                                ;*RPM1 MICRO ERROR 13
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 16
                                ;*RPM1 MICRO ERROR 14
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 1 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 16
```

```
2446                    ;*RPM1 MICRO ERROR 15
2447                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
2448                    ;*M8950, M8953. M8955'S
2449                    ;*RPM1-CHANNEL 2 FAILED
2450                    ;*ACTUAL = NNNN
2451                    ;*EXPECTED = NNNN
2452                    ;*
2453                    ;*RPM1 MICRO TEST 16
2454                    ;*RPM1 MICRO ERROR 1o
2455                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
2456                    ;*M8950, M8953, M8955'S
2457                    ;*RPM1-CHANNEL 3 FAILED
2458                    ;*ACTUAL = NNNN
2459                    ;*EXPECTED = NNNN
2460                    ;*
2461                    ;*RPM1 MICRO TEST 16
2462                    ;*RPM1 MICRO ERROR 17
2463                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
2464                    ;*M8950, M8953, M8955'S
2465                    ;*RPM1-CHANNEL 4 FAILED
2466                    ;*ACTUAL = NNNN
2467                    ;*EXPECTED = NNNN
2468                    ;*
2469                    ;*RPM1 MICRO TEST 16
2470                    ;*RPM1 MICRO ERROR 20
2471                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
2472                    ;*M8950, M8953, M8955'S
2473                    ;*RPM1-CHANNEL 5 FAILED
2474                    ;*ACTUAL = NNNN
2475                    ;*EXPECTED = NNNN
2476                    ;*
2477                    ;*RPM1 MICRO TEST 16
2478                    ;*RPM1 MICRO ERROR 21
2479                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
2480                    ;*M8950, M8953, M8955'S
2481                    ;*RPM1-CHANNEL 6 FAILED
2482                    ;*ACTUAL = NNNN
2483                    ;*EXPECTED = NNNN
2484                    ;*
2485                    ;*RPM1 MICRO TEST 16
2486                    ;*RPM1 MICRO ERROR 22
2487                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
2488                    ;*M8950, M8953, M8955'S
2489                    ;*RPM1-CHANNEL 7 FAILED
2490                    ;*ACTUAL = NNNN
2491                    ;*EXPECTED = NNNN
2492                    ;*
2493                    ;*RPM1 MICRO TEST 16
2494                    ;*RPM1 MICRO ERROR 23
2495                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
2496                    ;*M8950, M8953, M8955'S
2497                    ;*RPM1-CHANNEL P FAILED
2498                    ;*PARITY BIT WRONG ON ACTUAL DATA BYTE
2499                    ;*ACTUAL = NNNN
```

```
2500  46FB                                  S
 (1)                                        ; ************************************************************************
2501
2502  46FB                                  TEST16: TESTX   @16              ;INITIALIZE THE TEST
 (1)  46FB  3E  0E            7.0                      MVI     A,@16          ;DEFINE THE TEST NUMBER
 (1)  46FD  CD  03  28       18.0                      CALL    TSET           ;SETUP THE TEST
2503                                        ;%RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #0
2504                                        ;&M8950, M8953, M8955'S
2505
2506  4700                                          REQ     @7,0,0,0,0
 (1)  4700  CD  06  28       18.0                      CALL    REQST
 (1)  4703  00                                         .BYTE   0              ;DATA PATTERN NUMBER
 (1)  4704  00  00                                     .WORD   0              ;SYSTEM "0" COUNT
 (1)  4706  00  00                                     .WORD   0              ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  4708  00                                         .BYTE   0              ;DATA COMPARE FLAG IF =1
 (1)  4709  07                                         .BYTE   @7             ;REQUEST CODE
2507  470A                                          RIN     R12L             ;GET THE UNITS DESIRED
 (1)  470A  DB  94           10.0                      IN      R12L                   ;READ R12L INTO AC
 (1)  470C  7F                4.0                      MOV     A,A                    ;RETRY LINK
2508  470D  32  41  4B       13.0          STA     UNITMP           ;STORE IN MEMORY
2509  4710  A7                4.0          ANA     A                ;SET THE CONDITION CODE
2510  4711  C2  22  47       10.0          JNZ     TST16X           ;GO TEST THE SPECIFIED UNITS
2511
2512  4714                                          ERR     EXIT,DUMMY
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4714  CD  09  28       18.0                      CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)        000B                                    MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4717  0B                                      .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  4718  00                                      .BYTE
 (1)  4719  CD  15  28       18.0                DUMMY:: CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  471C  DA  B8  48       10.0                        JC      EXIT             ;LOOP ADDRESS IF LOOP SPECIFIED
2513                                        ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
2514                                        ;>FATAL ERROR - TEST ABORTED
2515  471F  C3  B8  48       10.0          JMP     EXIT
2516  4722  3A  41  4B       13.0    TST16X: LDA     UNITMP           ;GET THE UNIT MAP
2517  4725  E6  01             7.0          ANI     @001             ;TEST UNIT 0?
2518  4727  CA  46  47       10.0          JZ      TEST17           ;NO-CHECK FOR TEST 17
2519  472A  CD  A0  49       18.0          CALL    CLEAR            ;CLEAR ALL THE TU PORTS
2520  472D  3E  00             7.0          MVI     A,0
2521  472F  CD  CB  49       18.0          CALL    RDCTL
2522
2523  4732                                          ENDTST  TST16X
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4732                                          REQ     7                ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4732  CD  06  28       18.0                      CALL    REQST
 (2)  4735  00                                         .BYTE                  ;DATA PATTERN NUMBER
 (2)  4736  00  00                                     .WORD                  ;SYSTEM "" COUNT
 (2)  4738  00  00                                     .WORD                  ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  473A  00                                         .BYTE                  ;DATA COMPARE FLAG IF =1
 (2)  473B  07                                         .BYTE   7              ;REQUEST CODE
 (1)  473C  3A  9A  4F       13.0          LDA     ITERA            ;GET ITERATION COUNT
 (1)  473F  3D                4.0          DCR     A                ;DOWNCOUNT
 (1)  4740  32  9A  4F       13.0          STA     ITERA            ;SAVE COUNT
 (1)  4743  F2  22  47       10.0          JP      TST16X           ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 17 - READ CHANNEL AMTIE TEST - FROM TU PORT #1
 4746                           ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*----- -----
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
 4746                           SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE AMTIE LINES FROM TAPE UNIT PORT 1
                                ;*THROUGH THE READ CHANNEL BOARDS AND TO THE READ PATH AMTIE REGISTER.
                                ;*256 AMTIE PATTERNS FROM 000-377(8) ARE USED.
 4746                           SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      IF TU PORT 1 NOT SELECTED BY THE USER
                                ;*          THEN-NEXT TEST
                                ;*          ELSE-CONTINUE
                                ;*      ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD TU PORT #1 SELECT CODE
                                ;*    CALL SUBROUTINE RDCTL
                                ;*ENDTST
 4746                           SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM1 MICRO TEST 17
                                ;*RPM1 MICRO ERROR 13
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 17
                                ;*RPM1 MICRO ERROR 14
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 1 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 17
                                ;*RPM1 MICRO ERROR 15
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 2 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
```

```
                                        ;*RPM1 MICRO TEST 17
                                        ;*RPM1 MICRO ERROR 16
                                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
                                        ;*M8950, M8953, M8955'S
                                        ;*RPM1-CHANNEL 3 FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 17
                                        ;*RPM1 MICRO ERROR 17
                                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
                                        ;*M8950, M8953, M8955'S
                                        ;*RPM1-CHANNEL 4 FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 17
                                        ;*RPM1 MICRO ERROR 20
                                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
                                        ;*M8950, M8953, M8955'S
                                        ;*RPM1-CHANNEL 5 FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 17
                                        ;*RPM1 MICRO ERROR 21
                                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
                                        ;*M8950, M8953, M8955'S
                                        ;*RPM1-CHANNEL 6 FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 17
                                        ;*RPM1 MICRO ERROR 22
                                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
                                        ;*M8950, M8953, M8955'S
                                        ;*RPM1-CHANNEL 7 FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM1 MICRO TEST 17
                                        ;*RPM1 MICRO ERROR 23
                                        ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
                                        ;*M8950, M8953, M8955'S
                                        ;*RPM1-CHANNEL P FAILED
                                        ;*PARITY BIT WRONG ON ACTUAL DATA BYTE
                                        ;*ACTUAL = NNNN
4746                                    $
 (1,                                    ; ***************************************************************************
4746                                    TEST17: TESTX   @17             ;INITIALIZE THE TEST
 (1)  4746  3E   0F          7.0                 MVI     A,@17           ;DEFINE THE TEST NUMBER
 (1)  4748  CD   03   28    18.0                 CALL    TSET            ;SETUP THE TEST
                                        ;%RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #1
                                        ;&M8950, M8953, M8955'S
```

```
 2618  474B  3A  41  4B   13.0  TST17X: LDA     UNITMP          ;GET THE UNIT MAP
 2619  474E  E6  02        7.0          ANI     @002            ;TEST UNIT 1?
 2620  4750  CA  6F  47   10.0          JZ      TEST20          ;NO-CHECK FOR TEST 20
 2621  4753  CD  A0  49   18.0          CALL    CLEAR           ;CLEAR ALL THE TU PORTS
 2622  4756  3E  01        7.0          MVI     A,@1
 2623  4758  CD  CB  49   18.0          CALL    RDCTL
 2624  475B                             ENDTST  TST17X
  (1)                          ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  475B                             REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  475B  CD  06  28   18.0                             CALL    REQST
  (2)  475E  00                                            .BYTE                   ;DATA PATTERN NUMBER
  (2)  475F  00  00                                        .WORD                   ;SYSTEM '"' COUNT
  (2)  4761  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  4763  00                                            .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  4764  07                                            .BYTE   7               ;REQUEST CODE
  (1)  4765  3A  9A  4F   13.0          LDA     ITERA           ;GET ITERATION COUNT
  (1)  4768  3D        4.0          DCR     A               ;DOWNCOUNT
  (1)  4769  32  9A  4F   13.0          STA     ITERA           ;SAVE COUNT
  (1)  476C  F2  4B  47   10.0          JP      TST17X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 20 - READ CHANNEL AMTIE TEST - FROM TU PORT #2
476F                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
476F                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE AMTIE LINES FROM TAPE UNIT PORT 2
                                ;*THROUGH THE READ CHANNEL BOARDS AND TO THE READ PATH AMTIE REGISTER.
                                ;*256 AMTIE PATTERNS FROM 000-377(8) ARE USED.
476F                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*      IF TU PORT 2 NOT SELECTED BY THE USER
                                ;*          THEN-NEXT TEST
                                ;*          ELSE-CONTINUE
                                ;*      ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD TU PORT #2 SELECT CODE
                                ;*    CALL SUBROUTINE RDCTL
                                ;*ENDTST
476F                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*RPM1 MICRO TEST 20
                                ;*RPM1 MICRO ERROR 13
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 20
                                ;*RPM1 MICRO ERROR 14
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 1 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 20
                                ;*RPM1 MICRO ERROR 15
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 2 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
```

```
2668                                    ;*RPM1 MICRO TEST 20
2669                                    ;*RPM1 MICRO ERROR 16
2670                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
2671                                    ;*M8950, M8953, M8955'S
2672                                    ;*RPM1-CHANNEL 3 FAILED
2673                                    ;*ACTUAL = NNNN
2674                                    ;*EXPECTED = NNNN
2675                                    ;*
2676                                    ;*RPM1 MICRO TEST 20
2677                                    ;*RPM1 MICRO ERROR 17
2678                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
2679                                    ;*M8950, M8953, M8955'S
2680                                    ;*RPM1-CHANNEL 4 FAILED
2681                                    ;*ACTUAL = NNNN
2682                                    ;*EXPECTED = NNNN
2683                                    ;*
2684                                    ;*RPM1 MICRO TEST 20
2685                                    ;*RPM1 MICRO ERROR 20
2686                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
2687                                    ;*M8950, M8953, M8955'S
2688                                    ;*RPM1-CHANNEL 5 FAILED
2689                                    ;*ACTUAL = NNNN
2690                                    ;*EXPECTED = NNNN
2691                                    ;*
2692                                    ;*RPM1 MICRO TEST 20
2693                                    ;*RPM1 MICRO ERROR 21
2694                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
2695                                    ;*M8950, M8953, M8955'S
2696                                    ;*RPM1-CHANNEL 6 FAILED
2697                                    ;*ACTUAL = NNNN
2698                                    ;*EXPECTED = NNNN
2699                                    ;*
2700                                    ;*RPM1 MICRO TEST 20
2701                                    ;*RPM1 MICRO ERROR 22
2702                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
2703                                    ;*M8950, M8953, M8955'S
2704                                    ;*RPM1-CHANNEL 7 FAILED
2705                                    ;*ACTUAL = NNNN
2706                                    ;*EXPECTED = NNNN
2707                                    ;*
2708                                    ;*RPM1 MICRO TEST 20
2709                                    ;*RPM1 MICRO ERROR 23
2710                                    ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
2711                                    ;*M8950, M8953, M8955'S
2712                                    ;*RPM1-CHANNEL P FAILED
2713                                    ;*PARITY BIT WRONG ON ACTUAL DATA BYTE
2714                                    ;*ACTUAL = NNNN
2715  476F                              $
 (1)                                    ; ***************************************************************************
2716  476F                              TEST20: TESTX   @020            ;INITIALIZE THE TEST
 (1)  476F  3E  10          7.0                     MVI     A,@020          ;DEFINE THE TEST NUMBER
 (1)  4771  CD  03  28     18.0                     CALL    TSET            ;SETUP THE TEST
2717                                    ;%RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #2
2718                                    ;&M8950, M8953, M8955'S
```

```
2719  4774  3A  41  4B   13.0  TST020X: LDA   UNITMP          ;GET THE UNIT MAP
2720  4777  E6  04        7.0           ANI   @004            ;TEST UNIT 2?
2721  4779  CA  98  47   10.0           JZ    TEST21          ;NO-CHECK FOR TEST 21
2722  477C  CD  A0  49   18.0           CALL  CLEAR           ;CLEAR ALL THE TU PORTS
2723  477F  3E  02        7.0           MVI   A,@2
2724  4781  CD  CB  49   18.0           CALL  RDCTL
2725  4784                               ENDTST TST020X
 (1)                           ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4784                               REQ   7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4784  CD  06  28   18.0                           CALL    REQST
 (2)  4787  00                                         .BYTE                 ;DATA PATTERN NUMBER
 (2)  4788  00  00                                     .WORD                 ;SYSTEM '"' COUNT
 (2)  478A  00  00                                     .WORD                 ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  478C  00                                         .BYTE                 ;DATA COMPARE FLAG IF =1
 (2)  478D  07                                         .BYTE   7             ;REQUEST CODE
 (1)  478E  3A  9A  4F   13.0           LDA   ITERA           ;GET ITERATION COUNT
 (1)  4791  3D        4.0           DCR   A               ;DOWNCOUNT
 (1)  4792  32  9A  4F   13.0           STA   ITERA           ;SAVE COUNT
 (1)  4795  F2  74  47   10.0           JP    TST020X         ;DO TEST UNTIL TILL = 0
2726
```

```
                                .SBTTL  TEST 21 - READ CHANNEL AMTIE TEST - FROM TU PORT #3
4798                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
4798                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE INTEGRITY OF THE AMTIE LINES FROM TAPE UNIT PORT 3
                                ;*THROUGH THE READ CHANNEL BOARDS AND TO THE READ PATH AMTIE REGISTER.
                                ;*256 AMTIE PATTERNS FROM 000-377(8) ARE USED.
4798                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*      IF TU PORT 3 NOT SELECTED BY THE USER
                                ;*          THEN-NEXT TEST
                                ;*          ELSE-CONTINUE
                                ;*      ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD TU PORT #3 SELECT CODE
                                ;*    CALL SUBROUTINE RDCTL
                                ;*ENDTST
4798                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM1 MICRO TEST 21
                                ;*RPM1 MICRO ERROR 13
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 21
                                ;*RPM1 MICRO ERROR 14
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 1 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 21
                                ;*RPM1 MICRO ERROR 15
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 2 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
```

```
                                ;*RPM1 MICRO TEST 21
                                ;*RPM1 MICRO ERROR 16
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 3 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 21
                                ;*RPM1 MICRO ERROR 17
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 4 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 21
                                ;*RPM1 MICRO ERROR 20
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 5 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 21
                                ;*RPM1 MICRO ERROR 21
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 6 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 21
                                ;*RPM1 MICRO ERROR 22
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL 7 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM1 MICRO TEST 21
                                ;*RPM1 MICRO ERROR 23
                                ;*RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*RPM1-CHANNEL P FAILED
                                ;*PARITY BIT WRONG ON ACTUAL DATA BYTE
                                ;*ACTUAL = NNNN
                                $
 (1)                            ; ***************************************************************************
4798                            TEST21: TESTX   @021            ;INITIALIZE THE TEST
 (1)  4798  3E  11        7.0                   MVI     A,@021          ;DEFINE THE TEST NUMBER
 (1)  479A  CD  03  28   18.0                   CALL    TSET            ;SETUP THE TEST
                                ;%RPM1-READ CHANNEL AMTIE TEST-FROM TU PORT #3
                                ;&M8950, M8953, M8955'S
```

```
 2821  479D  3A  41  4B   13.0  TST21X: LDA     UNITMP          ;GET THE UNIT MAP
 2822  47A0  E6  08        7.0          ANI     @010            ;TEST UNIT 3?
 2823  47A2  CA  C1  47   10.0          JZ      TEST22          ;NO-GO RUN TEST 22
 2824  47A5  CD  A0  49   18.0          CALL    CLEAR           ;CLEAR ALL THE TU PORTS
 2825  47A8  3E  03        7.0          MVI     A,@3
 2826  47AA  CD  CB  49   18.0          CALL    RDCTL
 2827  47AD                             ENDTST  TST21X
  (1)                           ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  47AD                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  47AD  CD  06  28   18.0                      CALL    REQST
  (2)  47B0  00                                     .BYTE                   ;DATA PATTERN NUMBER
  (2)  47B1  00  00                                 .WORD                   ;SYSTEM '"' COUNT
  (2)  47B3  00  00                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  47B5  00                                     .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  47B6  07                                     .BYTE   7               ;REQUEST CODE
  (1)  47B7  3A  9A  4F   13.0          LDA     ITERA           ;GET ITERATION COUNT
  (1)  47BA  3D        4.0          DCR     A               ;DOWNCOUNT
  (1)  47BB  32  9A  4F   13.0          STA     ITERA           ;SAVE COUNT
  (1)  47BE  F2  9D  47   10.0          JP      TST21X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 22 - ECC SELF TEST

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*ECC - SELF TEST
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST COMMANDS THE ERROR CORRECTION CODE MICRO CONTROLLER TO RUN
                                ;*ITS SELF TEST DIAGNOSTICS IN BOTH THE FORWARD AND REVERSE MODES.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    PUT THE TM78 SUBSYSTEM IN THE READ FORWARD GCR MODE
                                ;*    SET NORMAL WRITE AND READ CLOCKS
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    CLEAR THE AMTIE LINES FROM THE USER SPECIFIED PORT
                                ;*    CLEAR THE DATA LINES FRON THE USER SPECIFIED PORT
                                ;*    DELAY 50 MICROSECONDS
                                ;*    ENABLE THE READ PATH
                                ;*    OUTPUT THE ECC SELF TEST COMMAND
                                ;*    DELAY 75 MICRO SECONDS
                                ;*    INPUT THE READ PATH STATUS
                                ;*      IF THE READ PATH STATUS = "ECCOK" (ECC PASS)
                                ;*          THEN-CONTINUE
                                ;*          ELSE-ERROR
                                ;*      ENDIF
                                ;*      IF ECC STATUS=NO ROM PE,NO AMTIE,PRRT ERR,UNC,TTEC,STEC
                                ;*          THEN-CONTINUE
                                ;*          ELSE-ERROR
                                ;*      ENDIF
                                ;*    PUT THE TM78 SUBSYSTEM IN THE READ REVERSE GCR MODE
                                ;*    CLEAR THE AMTIE LINES FROM THE USER SPECIFIED PORT
                                ;*    CLEAR THE DATA LINES FRON THE USER SPECIFIED PORT
                                ;*    DELAY 50 MICROSECONDS
                                ;*    ENABLE THE READ PATH
                                ;*    OUTPUT THE READ CHANNEL SELF TEST COMMAND
                                ;*    DELAY 75 MICRO SECONDS
                                ;*    OUTPUT THE ECC SELF TEST COMMAND
                                ;*    DELAY 75 MICRO SECONDS
                                ;*    INPUT THE READ PATH STATUS
                                ;*      IF THE READ PATH STATUS = "ECCOK" (ECC PASS)
                                ;*          THEN-CONTINUE
                                ;*          ELSE-ERROR
                                ;*      ENDIF
                                ;*      IF ECC STATUS=NO ROM PE, NO AMTIE,PTR ERR, UNC, TTEC, STEC
                                ;*          THEN-CONTINUE
                                ;*          ELSE-ERROR
                                ;*      ENDIF
```

```
2874                                    ;*ENDTST
2875  47C1                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
2876                                    ;*RPM1 MICRO TEST 22
2877                                    ;*RPM1 MICRO ERROR 11
2878                                    ;*RPM1-ERROR CORRECTION-SELF TEST
2879                                    ;*M8951, M8953, M8950
2880                                    ;*ECC MICRO FAILED FORWARD
2881                                    ;*ACTUAL = NNNN
2882                                    ;*EXPECTED = NNNN
2883                                    ;*ACTUAL = ECCSTA EXPECTED = RSTAT
2884                                    ;*
2885                                    ;*RPM1 MICRO TEST 22
2886                                    ;*RPM1 MICRO ERROR 12
2887                                    ;*RPM1-ERROR CORRECTION-SELF TEST
2888                                    ;*M8951, M8953, M8950
2889                                    ;*ECC MICRO FAILED REVERSE
2890                                    ;*ACTUAL = NNNN
2891                                    ;*EXPECTED = NNNN
2892                                    ;*ACTUAL = ECCSTA EXPECTED = RSTAT
2893  47C1                              S
 (1)                                    ; ***************************************************************************
```

```
2895  47C1                          TEST22: TESTX   @22              ;INITIALIZE TEST 22
 (1)  47C1  3E  12          7.0                     MVI     A,@22          ;DEFINE THE TEST NUMBER
 (1)  47C3  CD  03  28     18.0                     CALL    TSET           ;SETUP THE TEST
2896                                ;%RPM1-ERROR CORRECTION-SELF TEST
2897                                ;&M8951, M8953, M8950
2898  47C6  CD  A0  49     18.0  T22ST:  CALL    CLEAR            ;CLEAR THE TU PORTS
2899  47C9  06  00           7.0          MVI     B,@0             ;LOAD THE ADDRESS OF THE BASE PORT
2900  47CB  3A  41  4B      13.0          LDA     UNITMP           ;GET THE USER SPECIFIED PORTS
2901  47CE  E6  01           7.0          ANI     $01              ;UNIT 0?
2902  47D0  C2  E6  47      10.0          JNZ     T22OUT           ;YES - EXIT UNIT SEARCH
2903  47D3  04               4.0          INR     B                ;UPDATE THE UNIT NUMBER
2904  47D4  3A  41  4B      13.0          LDA     UNITMP           ;GET THE USER SPECIFIED PORTS
2905  47D7  E6  02           7.0          ANI     $02              ;UNIT 1?
2906  47D9  C2  E6  47      10.0          JNZ     T22OUT           ;YES - EXIT UNIT SEARCH
2907  47DC  04               4.0          INR     B                ;UPDATE THE UNIT NUMBER
2908  47DD  3A  41  4B      13.0          LDA     UNITMP           ;GET THE USER SPECIFIED PORTS
2909  47E0  E6  04           7.0          ANI     $04              ;UNIT 2?
2910  47E2  C2  E6  47      10.0          JNZ     T22OUT           ;YES - EXIT UNIT SEARCH
2911  47E5  04               4.0          INR     B                ;UPDATE THE UNIT NUMBER
2912  47E6  78               4.0  T22OUT: MOV     A,B              ;SAVE THE SELECTED PORT NUMBER
2913  47E7  32  48  4B      13.0          STA     SUNIT
2914  47EA  DB  E0          10.0          IN      INTSTA           ;GET THE MASS BUS SELECT BIT
2915  47EC  E6  80           7.0          ANI     BIT7             ;SAVE THE BIT
2916  47EE  B0               4.0          ORA     B                ;SET THE PORT BITS
2917  47EF  D3  E0          10.0          OUT     MBSEL            ;SELECT THE DESIRED TU PORT
2918  47F1  3E  10           7.0          MVI     A,W.GCR          ;PUT ENTIRE TM78 IN
2919  47F3  D3  D3          10.0          OUT     WMCCTL           ;THE WRITE GCR FORWARD MODE
2920                                                               ;
2921  47F5  3E  08           7.0  ECCLO1: MVI     A,@10            ;LOAD +10% READ CLOCK
2922  47F7  D3  F0          10.0          OUT     CLKCTL           ;SET NORMAL CLOCKS
2923  47F9  AF               4.0          XRA     A                ;
2924  47FA  D3  48          10.0          OUT     PDIAG            ;CLEAR THE DATA LINES
2925  47FC  D3  44          10.0          OUT     TAMT             ;CLEAR THE AMTIE LINES
2926  47FE  3E  60           7.0          MVI     A,P.LCS!P.LWR
2927  4800  D3  48          10.0          OUT     PDIAG            ;JOIN THE TU PORT TO THE READ PATH
2928  4802  3E  14           7.0          MVI     A,P.WPEN!P.RPEN  ;SET THE DATA PATH ENABLE BITS
2929  4804  D3  4C          10.0          OUT     PENAB
2930  4806  3E  0A           7.0          MVI     A,10             ;DELAY
2931  4808  3D               4.0  1$:     DCR     A                ;50
2932  4809  C2  08  48      10.0          JNZ     1$               ;MICROSECONDS
2933
2934  480C  3E  0A           7.0          MVI     A,$A             ;LOAD READ PATH ENABLE BIT
2935                                                               ;PLO DISABLE AND STATISTICS
2936  480E  D3  09          10.0          OUT     RPCTL            ;ENABLE THE READ PATH
2937                                                               ;
2938  4810  3E  0E           7.0          MVI     A,ECCTST         ;LOAD THE ECC SELF TEST COMMAND
2939  4812  D3  0B          10.0          OUT     RCMD             ;START THE TEST
2940                                ;
2941                                ;PERFORM A 75 MICROSECOND TIMEOUT
2942                                ;
2943  4814  3E  0F           7.0          MVI     A,15             ;SET TIME LOOP COUNTER
2944  4816  3D               4.0  TLOOP3: DCR     A                ;DECREMENT THE COUNTER
2945  4817  C2  16  48      10.0          JNZ     TLOOP3           ;CONTINUE-UNTIL DONE
2946                                                               ;
```

```
2947  481A  DB  02        10.0          IN      RSTAT           ;GET THE TEST RESULTS
2948  481C  FE  41         7.0          CPI     ECCOK           ;ECC OK?
2949  481E  C2  2A  48     10.0          JNZ     1$              ;NO-ERROR
2950  4821  DB  1A        10.0          IN      ECCSTA          ;YES-CHECK THE ECC STATUS
2951  4823  E6  6F         7.0          ANI     @157            ;JUST LOOK AT ROM PE,AMTIE,
2952                                                            ;POINTER MISMATCH,SINGLE TRACK ERROR,
2953                                                            ;AND DOUBLE TRACK ERROR.
2954  4825  FE  0F         7.0          CPI     @017            ;ANY ERRORS ?
2955  4827  CA  39  48     10.0          JZ      ECCCO1          ;NO-CONTINUE
2956  482A  DB  1A        10.0  1$:     IN      ECCSTA          ;GET THE ECC STATUS
2957  482C                              ROUT    ADATA           ;PUT IT IN ADATA FOR ERROR REPORT
 (1)  482C  D3  94        10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  482E  7F             4.0                  MOV     A,A             ;RETRY LINK
2958  482F  DB  02        10.0          IN      RSTAT           ;GET THE READ STATUS
2959  4831                              ROUT    EDATA           ;PUT IT IN EDATA FOR ERROR REPORT
 (1)  4831  D3  95        10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4833  7F             4.0                  MOV     A,A             ;RETRY LINK
2960  4834                              ERRB    T22ST,ECCCO1
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4834  CD  12  28     18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000C                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4837  0C                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4838  00                                 .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4839  CD  15  28     18.0          ECCCO1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  483C  DA  C6  47     10.0                  JC      T22ST           ;LOOP ADDRESS IF LOOP SPECIFIED
2961                                    ;>ECC MICRO FAILED FORWARD
2962                                    ;<ACTUAL DATA=ECCSTA, EXPECTED DATA=RSTAT
2963  483F  3E  14         7.0  T22TX: MVI     A,W.GCR!W.REV
2964  4841  D3  D3        10.0          OUT     WMCCTL
2965                                                            ;
2966                                                            ;
2967  4843  3A  48  43     13.0  ECCLOP: LDA     SUNIT           ;GET THE UNIT NUMBER
2968  4846  47              4.0          MOV     B,A
2969  4847  DB  E0        10.0          IN      INTSTA          ;GET THE MASS BUS SELECT BIT
2970  4849  E6  80         7.0          ANI     BIT7            ;SAVE THE BIT
2971  484B  B0              4.0          ORA     B               ;SET THE PORT BITS
2972  484C  D3  E0        10.0          OUT     MBSEL           ;SELECT THE DESIRED TU PORT
2973  484E  3E  08         7.0          MVI     A,@10           ;LOAD +10% READ CLOCK
2974  4850  D3  F0        10.0          OUT     CLKCTL          ;SET NORMAL CLOCKS
2975  4852  AF              4.0          XRA     A               ;
2976  4853  D3  40        10.0          OUT     TCMD            ;CLEAR THE DATA LINES
2977  4855  D3  44        10.0          OUT     TAMT            ;CLEAR THE AMTIE LINES
2978  4857  3E  60         7.0          MVI     A,P.LCS!P.LWR
2979  4859  D3  48        10.0          OUT     PDIAG           ;JOIN THE TU PORT TO THE READ PATH
2980  485B  3E  14         7.0          MVI     A,P.RPEN!P.WPEN
2981  485D  D3  4C        10.0          OUT     PENAB
2982  485F  3E  0A         7.0          MVI     A,10            ;DELAY
2983  4861  3D              4.0  1$:     DCR     A               ;50
2984  4862  C2  61  48     10.0          JNZ     1$              ;MICROSECONDS
2985  4865  3E  0A         7.0          MVI     A,$A            ;LOAD READ PATH ENABLE BIT
2986                                                            ;PLO DISABLE AND STATISTICS
2987  4867  D3  09        10.0          OUT     RPCTL           ;ENABLE THE READ PATH
2988                                                            ;
2989  4869  D3  08        10.0          OUT     RFIFOL          ;CLOCK THE FIFO
```

```
2990  486B  3E  0D          7.0          MVI   A,RCLRT          ;LOAD THE READ PATH CLEAR COMMAND
2991  486D  D3  0B         10.0          OUT   RCMD             ;ISSUE THE COMMAND
2992                                     ;
2993  486F  3E  0F          7.0          MVI   A,15             ;WAIT FOR APPROX 75 MICROSEC
2994  4871  3D              4.0  T22WT:  DCR   A
2995  4872  C2  71  48      10.0          JNZ   T22WT            ;STAY HERE TILL READY
2996                                     ;
2997  4875  3E  0E          7.0          MVI   A,ECCTST         ;LOAD THE ECC SELF TEST COMMAND
2998  4877  D3  0B         10.0          OUT   RCMD             ;START THE ECC SELF TEST
2999                                     ;
3000                                ;
3001                                ; PERFORM A 75 MICRO SECOND TIMEOUT
3002                                ;
3003  4879  3E  0F          7.0          MVI   A,15             ;SET TIME LOOP COUNTER
3004  487B  3D              4.0  TLOOP2: DCR   A                ;DECREMENT COUNTER
3005  487C  C2  7B  48      10.0          JNZ   TLOOP2           ;CONTINUE
3006                                     ;
3007                                     ;
3008  487F  DB  02         10.0          IN    RSTAT            ;GET THE TEST RESULTS
3009  4881  FE  41          7.0          CPI   ECCOK            ;ECC MICRO OK?
3010  4883  C2  8F  48      10.0          JNZ   1$               ;NO - ERROR
3011  4886  DB  1A         10.0          IN    ECCSTA           ;YES - CHECK ECC STATUS
3012  4888  E6  6F          7.0          ANI   @157             ;JUST LOOK AT ROM PE, AMTIE,
3013                                     ;POINTER MISMATCH,SINGLE TRACK ERROR,
3014                                     ;AND DOUBLE TRACK ERROR.
3015  488A  FE  0F          7.0          CPI   @017             ;ANY ERRORS ?
3016  488C  CA  8E  48      10.0          JZ    ECCCON           ;NO-CONTINUE
3017  488F  DB  1A         10.0  1$:     IN    ECCSTA           ;GET THE ECC STATUS
3018  4891                             ROUT    ADATA            ;PUT IT IN ADATA FOR ERROR REPORT
 (1)  4891  D3  94         10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4893  7F              4.0                 MOV     A,A             ;RETRY LINK
3019  4894  DB  02         10.0          IN      RSTAT            ;GET THE READ STATUS
3020  4896                             ROUT    EDATA            ;PUT IT IN EDATA FOR ERROR REPORT
 (1)  4896  D3  95         10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4898  7F              4.0                 MOV     A,A             ;RETRY LINK
3021  4899                             ERRB    ECCLOP,ECCCON
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4899  CD  12  28      18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      000D                                 MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  489C  0D                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  489D  00                                 .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  489E  CD  15  28      18.0         ECCCON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48A1  DA  43  48      10.0                 JC      ECCLOP          ;LOOP ADDRESS IF LOOP SPECIFIED
3022                                ;>ECC MICRO FAILED REVERSE
3023                                ;<ACTUAL DATA=ECCSTA, EXPECTED DATA=RSTAT
3024  48A4                              ENDTST  T22ST
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  48A4                          REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  48A4  CD  06  28      18.0                 CALL    REQST
 (2)  48A7  00                                 .BYTE                   ;DATA PATTERN NUMBER
 (2)  48A8  00  00                             .WORD                   ;SYSTEM "" COUNT
 (2)  48AA  00  00                             .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  48AC  00                                 .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  48AD  07                                 .BYTE   7               ;REQUEST CODE
```

```
   (1)  48AE   3A   9A   4F      13.0          LDA     ITERA           ;GET ITERATION COUNT
   (1)  48B1   3D                 4.0          DCR     A               ;DOWNCOUNT
   (1)  48B2   32   9A   4F      13.0          STA     ITERA           ;SAVE COUNT
   (1)  48B5   F2   C6   47      10.0          JP      T22ST           ;DO TEST UNTIL TILL = 0
3025    48B8   CD   A0   49      18.0  EXIT:   CALL    CLEAR           ;CLEAR THE PROTS BEFORE LEAVING
3026    48BB   C3   18   28      10.0          JMP     TSTEND          ;END OF TEST
```

```
                                .SBTTL  SUBROUTINE CHECK AMTIE
48BE                            SSUB
                                ;*************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*CHECK AMTIE SUBROUTINE
48BE                            SD
                                ;*************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO ISOLATE A BIT FAILURE IN THE READ PATH DATA
                                ;*OR AMTIE BYTES TO A SPECIFIC READ CHANNEL BY COMPARING THE ACTUAL AND
                                ;*EXPECTED INPUTS TO THE SUBROUTINE ON A BIT BY BIT BASIS.
48BE                            SP
                                ;*************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*    IF CHANNEL 0 ACTUAL INFORMATION NOT = CHANNEL 0 EXPECTED
                                ;*      THEN-ERROR CHANNEL 0
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 1 ACTUAL INFORMATION NOT = CHANNEL 1 EXPECTED
                                ;*      THEN-ERROR CHANNEL 1
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 2 ACTUAL INFORMATION NOT = CHANNEL 2 EXPECTED
                                ;*      THEN-ERROR CHANNEL 2
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 3 ACTUAL INFORMATION NOT = CHANNEL 3 EXPECTED
                                ;*      THEN-ERROR CHANNEL 3
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 4 ACTUAL INFORMATION NOT - CHANNEL 4 EXPECTED
                                ;*      THEN-ERROR CHANNEL 4
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 5 ACTUAL INFORMATION NOT = CHANNEL 5 EXPECTED
                                ;*      THEN-ERROR CHANNEL 5
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 6 ACTUAL INFORMATION NOT = CHANNEL 6 EXPECTED
                                ;*      THEN-ERROR CHANNEL 6
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 7 ACTUAL INFORMATION NOT = CHANNEL 7 EXPECTED
                                ;*      THEN-ERROR CHANNEL 7
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL P ACTUAL INFORMATION NOT - CHANNEL P EXPECTED
                                ;*      THEN-ERROR CHANNEL P
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
```

```
3073                                        ;*ENDSUB
3074  48BE                                  $
 (1)                                        ; ************************************************************************
3075
3076  48BE   21   46   4B        10.0  CKAMT:  LXI     H,DATAE          ;LOAD EXPECTED DATA ADDRESS
3077  48C1   3A   44   4B        13.0          LDA     DATAA            ;GET THE ACTUAL DATA
3078  48C4                                    ROUT    ADATA            ;WRITE THE ACTUAL DATA TO THE CAS
 (1)  48C4   D3   94              10.0                  OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  48C6   7F                    4.0                  MOV     A,A              ;RETRY LINK
3079  48C7   E6   01               7.0          ANI     $01              ;REMOVE UNWANTED BITS
3080  48C9   47                    4.0          MOV     B,A              ;SAVE IN B
3081  48CA   7E                    7.0          MOV     A,M              ;GET EXPECTED DATA
3082  48CB                                      ROUT    EDATA            ;WRITE THE EXPECTED DATA TO THE CAS
 (1)  48CB   D3   95              10.0                  OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  48CD   7F                    4.0                  MOV     A,A              ;RETRY LINK
3083  48CE   E6   01               7.0          ANI     $01              ;REMOVE UNWANTED BITS
3084  48D0   B8                    4.0          CMP     B                ;COMPARE
3085  48D1   CA   D9   48         10.0          JZ      CKAM0            ;CONTINUE IF EQUAL
3086  48D4                                      ERRB    OUT,CKAM0
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  48D4   CD   12   28         18.0                  CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)       000E                                       MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48D7   0E                                       .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  48D8   00                                       .BYTE                    ;PRINT ROUTINE NUMBER
 (1)  48D9   CD   15   28         18.0          CKAM0:: CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48DC   DA   9F   49         10.0                  JC      OUT              ;LOOP ADDRESS IF LOOP SPECIFIED
3087                                      ;>CHANNEL 0 FAILED
3088  48DF   3A   44   4B        13.0          LDA     DATAA            ;GET THE ACTUAL DATA
3089  48E2   E6   02               7.0          ANI     $02              ;REMOVE UNWANTED BITS
3090  48E4   47                    4.0          MOV     B,A              ;SAVE IN B
3091  48E5   7E                    7.0          MOV     A,M              ;GET EXPECTED DATA
3092  48E6   E6   02               7.0          ANI     $02              ;REMOVE UNWANTED BITS
3093  48E8   B8                    4.0          CMP     B                ;COMPARE
3094  48E9   CA   F1   48         10.0          JZ      CKAM1            ;CONTINUE IF EQUAL
3095  48EC                                      ERRB    OUT,CKAM1
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  48EC   CD   12   28         18.0                  CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)       000F                                       MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48EF   0F                                       .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  48F0   00                                       .BYTE                    ;PRINT ROUTINE NUMBER
 (1)  48F1   CD   15   28         18.0          CKAM1:: CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48F4   DA   9F   49         10.0                  JC      OUT              ;LOOP ADDRESS IF LOOP SPECIFIED
3096                                      ;>CHANNEL 1 FAILED
3097  48F7   3A   44   4B        13.0          LDA     DATAA            ;GET THE ACTUAL DATA
3098  48FA   E6   04               7.0          ANI     $04              ;REMOVE UNWANTED BITS
3099  48FC   47                    4.0          MOV     B,A              ;SAVE IN B
3100  48FD   7E                    7.0          MOV     A,M              ;GET EXPECTED DATA
3101  48FE   E6   04               7.0          ANI     $04              ;REMOVE UNWANTED BITS
3102  4900   B8                    4.0          CMP     B                ;COMPARE
3103  4901   CA   09   49         10.0          JZ      CKAM2            ;CONTINUE IF EQUAL
```

```
3105  4904                                                ERRB    OUT,CKAM2
 (1)                                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4904   CD   12   28        18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0010                                                 MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4907   10                                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4908   00                                                 .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4909   CD   15   28        18.0                    CKAM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  490C   DA   9F   49        10.0                            JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
3106                                            ;>CHANNEL 2 FAILED
3107  490F   3A   44   4B        13.0                    LDA     DATAA           ;GET THE ACTUAL DATA
3108  4912   E6   08              7.0                    ANI     $08             ;REMOVE UNWANTED BITS
3109  4914   47                    4.0                    MOV     B,A             ;SAVE IN B
3110  4915   7E                    7.0                    MOV     A,M             ;GET EXPECTED DATA
3111  4916   E6   08              7.0                    ANI     $08             ;REMOVE UNWANTED BITS
3112  4918   B8                    4.0                    CMP     B               ;COMPARE
3113  4919   CA   21   49        10.0                    JZ      CKAM3           ;CONTINUE IF EQUAL
3114  491C                                                ERRB    OUT,CKAM3
 (1)                                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  491C   CD   12   28        18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0011                                                 MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  491F   11                                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4920   00                                                 .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4921   CD   15   28        18.0                    CKAM3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4924   DA   9F   49        10.0                            JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
3115                                            ·>CHANNEL 3 FAILED
3116  4927   3A   44   4B        13.0                    LDA     DATAA           ;GET THE ACTUAL DATA
3117  492A   E6   10              7.0                    ANI     $10             ;REMOVE UNWANTED BITS
3118  492C   47                    4.0                    MOV     B,A             ;SAVE IN B
3119  492D   7E                    7.0                    MOV     A,M             ;GET EXPECTED DATA
3120  492E   E6   10              7.0                    ANI     $10             ;REMOVE UNWANTED BITS
3121  4930   B8                    4.0                    CMP     B               ;COMPARE
3122  4931   CA   39   49        10.0                    JZ      CKAM4           ;CONTINUE IF EQUAL
3123  4934                                                ERRB    OUT,CKAM4
 (1)                                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4934   CD   12   28        18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0012                                                 MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4937   12                                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4938   00                                                 .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4939   CD   15   28        18.0                    CKAM4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  493C   DA   9F   49        10.0                            JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
3124                                            ;>CHANNEL 4 FAILED
```

```
3126  493F  3A  44  4B  13.0          LDA     DATAA           ;GET THE ACTUAL DATA
3127  4942  E6  20      7.0          ANI     $20             ;REMOVE UNWANTED BITS
3128  4944  47          4.0          MOV     B,A             ;SAVE IN B
3129  4945  7E          7.0          MOV     A,M             ;GET EXPECTED DATA
3130  4946  E6  20      7.0          ANI     $20             ;REMOVE UNWANTED BITS
3131  4948  B8          4.0          CMP     B               ;COMPARE
3132  4949  CA  51  49  10.0         JZ      CKAM5           ;CONTINUE IF EQUAL
3133  494C                           ERRB    OUT,CKAM5
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  494C  CD  12  28  18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0013                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  494F  13                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4950  00                                  .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4951  CD  15  28  18.0           CKAM5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4954  DA  9F  49  10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
3134                           ;>CHANNEL 5 FAILED
3135  4957  3A  44  4B  13.0          LDA     DATAA           ;GET THE ACTUAL DATA
3136  495A  E6  40      7.0          ANI     $40             ;REMOVE UNWANTED BITS
3137  495C  47          4.0          MOV     B,A             ;SAVE IN B
3138  495D  7E          7.0          MOV     A,M             ;GET EXPECTED DATA
3139  495E  E6  40      7.0          ANI     $40             ;REMOVE UNWANTED BITS
3140  4960  B8          4.0          CMP     B               ;COMPARE
3141  4961  CA  69  49  10.0         JZ      CKAM6           ;CONTINUE IF EQUAL
3142  4964                           ERRB    OUT,CKAM6
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4964  CD  12  28  18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0014                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4967  14                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4968  00                                  .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4969  CD  15  28  18.0           CKAM6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  496C  DA  9F  49  10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
3143                           ;>CHANNEL 6 FAILED
3144  496F  3A  44  4B  13.0          LDA     DATAA           ;GET THE ACTUAL DATA
3145  4972  E6  80      7.0          ANI     $80             ;REMOVE UNWANTED BITS
3146  4974  47          4.0          MOV     B,A             ;SAVE IN B
3147  4975  7E          7.0          MOV     A,M             ;GET EXPECTED DATA
3148  4976  E6  80      7.0          ANI     $80             ,REMOVE UNWANTED BITS
3149  4978  B8          4.0          CMP     B               ;COMPARE
3150  4979  CA  81  49  10.0         JZ      CKAM7           ;CONTINUE IF EQUAL
3151  497C                           ERRB    OUT,CKAM7
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  497C  CD  12  28  18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0015                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  497F  15                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4980  00                                  .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4981  CD  15  28  18.0           CKAM7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4984  DA  9F  49  10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
3152                           ;>CHANNEL 7 FAILED
```

```
3154  4987  3A  45  4B      13.0          LDA     DATAAP          ;GET THE ACTUAL DATA
3155  498A  E6  01            7.0          ANI     $01             ;REMOVE UNWANTED BITS
3156  498C  47                4.0          MOV     B,A             ;SAVE IN B
3157  498D  7E                7.0          MOV     A,M             ;GET EXPECTED DATA
3158  498E  E6  01            7.0          ANI     $01             ;REMOVE UNWANTED BITS
3159  4990  B8                4.0          CMP     B               ;COMPARE
3160  4991  CA  99  49       10.0          JZ      CKAMP           ;CONTINUE IF EQUAL
3161  4994                                 ERRA    OUT,CKAMP
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (     4994  CD  0F  28       18.0                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1,         0016                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)   4997  16                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)   4998  00                           .BYTE                   ;PRINT ROUTINE NUMBER
 (1)   4999  CD  15  28       18.0  CKAMP:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)   499C  DA  9F  49       10.0          JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
3162                                 ;>CHANNEL P FAILED
3163                                 ;>PARITY BIT WRONG ON ACTUAL DATA BYTE
3164
3165  499F  C9                10.0  OUT:    RET
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ;*********************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ;*********************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ;*********************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*    CLEAR PORT SELECT FOR TRANSPORT
                                        ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*    CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ;*********************************************************************
49A0  F5              12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
49A1  C5              12.0          PUSH    B
49A2  06  00           7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
49A4  DB  E0          10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
49A6  E6  80           7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
49A8  B0               4.0          ORA     B               ;ADD IN THE SELECTED PORT #
49A9  D3  E0          10.0          OUT     MBSEL           ;RESET TO THIS PORT
49AB  3E  80           7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
49AD  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
49AF  AF               4.0          XRA     A               ;CLEAR TU COMMAND A
49B0  D3  40          10.0          OUT     TCMD            ;
49B2  3E  81           7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
49B4  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
49B6  3E  00           7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
49B8  D3  40          10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
49BA  AF               4.0          XRA     A
49BB  D3  44          10.0          OUT     TAMT            ;CLEAR AMTIE WORD
49BD  D3  48          10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
49BF  D3  4C          10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
49C1  04               4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
49C2  78               4.0          MOV     A,B
49C3  FE  04           7.0          CPI     4               ;DONE?
49C5  C2  A4  49      10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
49C8  C1              10.0          POP     B               ;RESET B & C
49C9  F1              10.0          POP     PSW             ;ALL DONE
49CA  C9              10.0          RET                     .EXIT
```

```
                                .SBTTL  SUBROUTINE RDCTL

                                SSUB
                                ; ***************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*READ PATH CONTROL SUBROUTINE
                                SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO CONTROL THE SEQUENCING AND LOAD OF
                                ;*AMTIE LINES TO THE SELECTED TU PORT BOARD.  DATA PATTERS ARE INCREMENTAL
                                ;*FROM 000-377.
                                SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*     SET "DPEN" (DATA PATH ENABLE) AND "LWR" (LOOP W/R) ON THE TU PORT
                                ;*     SET NORMAL READ PATH CLOCK
                                ;*     CLEAR READ PATH CONTROLLER STOP BIT
                                ;*     CLEAR THE AMTIE DATA BYTE
                                ;*     BGNDO
                                ;*        :    GET THE AMTIE DATA BYTE
                                ;*        :    OUTPUT THE AMTIE DATA BYTE TO THE TU PORT AMTIE BYTE
                                ;*        :    GET THE LEAST SIGNIFICANT BIT OF THE AMTIE DATA BYTE
                                ;*        :    OUTPUT THE LEAST SIGNIFICANT BIT AS AMTIE PARITY
                                ;*        :    WAIT 50 MICRO SECONDS
                                ;*        :    INPUT THE AMTIE DATA FROM THE READ CHANNEL
                                ;*        :    INPUT THE AMTIE PARITY FROM THE READ CHANNEL
                                ;*        :    CALL "CKAMT" SUBROUTINE
                                ;*        :    INCREMENT THE AMTIE DATA BYTE
                                ;*        :    DO UNTIL THE AMTIE DATA BYTE = ZERO
                                ;*     ENDDO
                                ;*ENDSUB
                                S
                                ; ***************************************************************************

 49CB  47              4.0  RDCTL:  MOV     B,A              ;SAVE THE UNIT NUMBER
 49CC  DB  E0         10.0         IN      INTSTA           ;GET THE MASS BUS PORT
 49CE  E6  80          7.0         ANI     BIT7
 49D0  B0              4.0         ORA     B
 49D1  D3  E0         10.0         OUT     MBSEL
 49D3  3E  20          7.0         MVI     A,P.LWR          ;CLEAR PORT ZERO CONTROL BYTE
 49D5  D3  48         10.0         OUT     PDIAG            ;SET LOOP MODE ON THE PORT
 49D7  3E  14          7.0         MVI     A,P.WPEN!P.RPEN  ;SET THE DATA PATH ENABLE BITS
 49D9  D3  4C         10.0         OUT     PENAB
                                                           ;
 49DB  3E  10          7.0         MVI     A,RDCLK          ;SET NORMAL READ PATH CLOCK
 49DD  D3  F0         10.0         OUT     CLKCTL           ;
 49DF  3E  00          7.0         MVI     A,0      ;
 49E1  D3  09         10.0         OUT     RPCTL
                                                           ;
```

```
3255  49E3  AF          4.0          XRA   A          ;CLEAR THE ACCUMULATOR
3256  49E4  32 46 4B    13.0         STA   DATAE      ;STORE AS AMTIE DATA
3257                                                  ;
3258  49E7  3A 46 4B    13.0  AMTLP0: LDA   DATAE      ;GET THE AMTIE DATA
3259  49EA  47           4.0          MOV   B,A        ;SAVE IN B
3260  49EB  D3 44        10.0         OUT   TAMT       ;WRITE TO THE AMTIE LINES
3261  49ED  E6 01        7.0          ANI   $01        ;GET THE LSB
3262  49EF  32 47 4B     13.0         STA   DATAEP     ;SAVE EXPECTED DATA PARITY
3263  49F2  F6 20        7.0          ORI   P.LWR      ;
3264  49F4  D3 48        10.0         OUT   PDIAG      ;OUTPUT AS PARITY BIT
3265
3266  49F6  3E 0A        7.0          MVI   A,10       ;WAIT
3267  49F8  3D           4.0   1$:    DCR   A          ;50
3268  49F9  C2 F8 49     10.0         JNZ   1$         ;MICROSECONDS
3269                                                  ;
3270  49FC  DB 10        10.0         IN    RAMT       ;GET THE AMTIE DATA
3271  49FE  32 44 4B     13.0         STA   DATAA      ;SAVE IT
3272  4A01  DB 15        10.0         IN    RPSTA      ;GET THE AMTIE P BIT
3273  4A03  E6 01        7.0          ANI   $01        ;REMOVE EXTRA BITS
3274  4A05  32 45 4B     13.0         STA   DATAAP     ;SAVE IT
3275  4A08  CD BE 48     18.0         CALL  CKAMT      ;GO SORT OUT ERRORS
3276  4A0B  DA E7 49     10.0         JC    AMTLP0     ;ERROR RETURN
3277  4A0E  3A 46 4B     13.0         LDA   DATAE      ;NORMAL RETURN
3278  4A11  3C           4.0          INR   A          ;UPDATE PATTERN
3279  4A12  32 46 4B     13.0         STA   DATAE      ;SAVE UPDATED PATTERN
3280  4A15  C2 E7 49     10.0         JNZ   AMTLP0     ;CONTINUE UNTIL DONE
3281  4A18  C9           10.0         RET              ;RETURN TO USER
```

```
                                .SBTTL  SUBROUTINE RCCLR
4A19                            SSUB
 (1)                            ; ***************************************************************************
 (1)                            ;*SUBROUTINE TITLE
 (1)                            ;*---------- -----
                                ;*RPM1-READ CHANNEL CLEAR SUBROUTINE
4A19                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNSUB
                                ;*  SET NORMAL READ PATH CLOCK
                                ;*  SET PLO DISABLE
                                ;*  CLOCK THE FIFO'S
                                ;*  ISSUE THE CLEAR ALL TEST COMMAND
                                ;*  WAIT 40 MICROSECONDS
                                ;*  STOP THE READ PATH AND SET PLO DISABLE
                                ;*ENDSUB
4A19                            S
 (1)                            ; ***************************************************************************

4A19  3E  10          7.0  RCCLR:  MVI   A,RDCLK          ;SET NORMAL READ PATH CLOCK
4A1B  D3  F0         10.0         OUT   CLKCTL
4A1D  3E  08          7.0         MVI   A,R.PLOD         ;SET PLO DISABLE
4A1F  D3  09         10.0         OUT   RPCTL
4A21  D3  08         10.0         OUT   RFIFOL           ;CLOCK THE FIFI'S
4A23  3E  0D          7.0         MVI   A,RCLRT          ;ISSUE CLEAR ALL TEST COMMAND
4A25  D3  0B         10.0         OUT   RCMD
4A27  3E  08          7.0         MVI   A,8              ;WAIT 40 MICRO SECONDS
4A29  3D              4.0  RCCLR1: DCR   A
4A2A  C2  29  4A     10.0         JNZ   RCCLR1
4A2D  3F  09          7.0         MVI   A,R.STPC.R.PLOD ;STOP THE READ PATH
4A2F  D3  09         10.0         OUT   RPCTL
4A31  C9             10.0         RET
```

```
                                        .SBTTL  SUBROUTINE RCSTRT
4A32                                    SSUB
 (1)                                    ; ************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
                                        ;*RPM1-READ CHANNEL START SUBROUTINE
4A32                                    SP
 (1)                                    ; ************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNSUB
                                        ;*  SET NORMAL READ PATH CLOCK
                                        ;*  SET PLO DISABLE
                                        ;*  CLOCK THE FIFO'S
                                        ;*  ISSUE THE READ CHANNEL SELF TEST COMMAND
                                        ;*  WAIT 40 MICROSECONDS
                                        ;*  STOP THE READ PATH AND SET PLO DISABLE
                                        ;*ENDSUB
4A32                                    S
 (1)                                    ; ************************************************************************

4A32  3E  10          7.0  RCSTRT: MVI     A,RDCLK          ;SET NORMAL READ PATH CLOCK
4A34  D3  F0         10.0         OUT     CLKCTL
4A36  3E  08          7.0         MVI     A,R.PLOD         ;SET PLO DISABLE
4A38  D3  09         10.0         OUT     RPCTL
4A3A  D3  08         10.0         OUT     RFIFOL           ;CLOCK THE FIFI'S
4A3C  3E  0C          7.0         MVI     A,RCHTST         ;ISSUE READ CHANNEL SELF TEST COMMAND
4A3E  D3  0B         10.0         OUT     RCMD
4A40  3E  08          7.0         MVI     A,8              ;WAIT 40 MICRO SECONDS
4A42  3D              4.0  RCSTR1: DCR     A
4A43  C2  42  4A     10.0         JNZ     RCSTR1
4A46  3E  09          7.0         MVI     A,R.STPC!R.PLOD ;STOP THE READ PATH
4A48  D3  09         10.0         OUT     RPCTL
4A4A  C9             10.0         RET
```

```
                                .SBTTL  SUBROUTINE RCCK
4A4B                            SSUB
 (1)                            ; ************************************************************************
 (1)                            ;*SUBROUTINE TITLE
 (1)                            ;*----------
                                ;*RPM1-READ CHANNEL SELF TEST CHECK
4A4B                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNSUB
                                ;*  GET THE READ PATH END MARK REGISTER
                                ;*  IF END MARK REGISTER ANDED WITH CHANNEL MASK=EXPECTED DATA
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  GET THE READ PATH DATA REGISTER
                                ;*  IF DATA REGISTER ANDED WITH CHANNEL MASK=EXPECTED DATA
                                ;*  :  THEN
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  GET THE READ PATH POSTAMBLE REGISTER
                                ;*  IF POSTAMBLE REGISTER ANDED WITH CHANNEL MASK=EXPECTED DATA
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  GET THE READ PATH ILLEGAL REGISTER
                                ;*  IF ILLEGAL REGISTER ANDED WITH CHANNEL MASK=EXPECTED DATA
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  GET THE READ PATH MARK TWO REGISTER
                                ;*  IF MARK TWO REGISTER ANDED WITH CHANNEL MASK=EXPECTED DATA
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  GET THE READ PATH NOT DONE REGISTER
                                ;*  IF NOT DONE REGISTER ANDED WITH CHANNEL MASK=EXPECTED DATA
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*ENDSUB
4A4B                            S
 (1)                            ; ************************************************************************

4A4B  DB  14        10.0  RCCK:   IN    REND        ;GET THE END BITS
4A4D  32  4C  4B    13.0          STA   XEND        ;SAVE
4A50  DB  17        10.0          IN    RDATA       ;GET THE DATA BITS
4A52  32  4D  4B    13.0          STA   XDATA       ;SAVE
4A55  DB  16        10.0          IN    RPOSTN      ;GET THE POSTAMBLE BITS
4A57  32  4E  4B    13.0          STA   XPOSTN      ;SAVE
4A5A  DB  12        10.0          IN    RILL        ;GET THE ILLEGAL BITS
4A5C  32  4F  4B    13.0          STA   XILL        ;SAVE
4A5F  DB  13        10.0          IN    RMK2        ;GET THE MARK 2 BITS
```

```
3388  4A61  32  50  4B   13.0          STA   XMK2          ;SAVE
3389  4A64  DB  11      10.0          IN    RDON          ;GET THE DONE BITS
3390  4A66  32  51  4B   13.0          STA   XDON          ;SAVE
3391  4A69  C3  B7  4A   10.0          JMP   RCCKX         ;CONTINUE
3392
3393  4A6C  DB  15       10.0  RCCKP:  IN    RPSTA         ;GET THE END PARITY BIT
3394  4A6E  E6  10        7.0          ANI   R.END         ;
3395  4A70  CA  75  4A   10.0          JZ    RCCKP1        ;
3396  4A73  3E  01        7.0          MVI   A,1           ;
3397  4A75  32  4C  4B   13.0  RCCKP1: STA   XEND          ;STORE THE END PARITY BIT RIGHT JUSTIFIED
3398  4A78  DB  15       10.0          IN    RPSTA         ;GET THE DATA PARITY BIT
3399  4A7A  E6  40        7.0          ANI   R.DATA        ;
3400  4A7C  CA  81  4A   10.0          JZ    RCCKP2        ;
3401  4A7F  3E  01        7.0          MVI   A,1           ;
3402  4A81  32  4D  4B   13.0  RCCKP2: STA   XDATA         ;STORE THE DATA PARITY BIT RIGHT JUSTIFIED
3403  4A84  DB  15       10.0          IN    RPSTA         ;GET THE POSTAMBLE PARITY BIT
3404  4A86  E6  20        7.0          ANI   R.POST        ;
3405  4A88  CA  8D  4A   10.0          JZ    RCCKP3        ;
3406  4A8B  3E  01        7.0          MVI   A,1           ;
3407  4A8D  32  4E  4B   13.0  RCCKP3: STA   XPOSTN        ;STORE THE POSTAMBLE PARITY BIT RIGHT JUSTIFIED
3408  4A90  DB  15       10.0          IN    RPSTA         ;GET THE ILLEGAL PARITY BIT
3409  4A92  E6  04        7.0          ANI   R.ILL         ;
3410  4A94  CA  99  4A   10.0          JZ    RCCKP4        ;
3411  4A97  3E  01        7.0          MVI   A,1           ;
3412  4A99  32  4F  4B   13.0  RCCKP4: STA   XILL          ;STORE THE ILLEGAL PARITY BIT RIGHT JUSTIFIED
3413  4A9C  DB  15       10.0          IN    RPSTA         ;GET THE MARK 2 PARITY BIT
3414  4A9E  E6  08        7.0          ANI   R.MK2         ;
3415  4AA0  CA  A5  4A   10.0          JZ    RCCKP5        ;
3416  4AA3  3E  01        7.0          MVI   A,1           ;
3417  4AA5  32  50  4B   13.0  RCCKP5: STA   XMK2          ;STORE THE MARK2 PARITY BIT RIGHT JUSTIFIED
3418  4AA8  DB  15       10.0          IN    RPSTA         ;GET THE DONE PARITY BIT
3419  4AAA  E6  02        7.0          ANI   R.DON         ;
3420  4AAC  CA  B1  4A   10.0          JZ    RCCKP6        ;
3421  4AAF  3E  01        7.0          MVI   A,1           ;
3422  4AB1  32  51  4B   13.0  RCCKP6: STA   XDON          ;STORE THE DONE PARITY BIT RIGHT JUSTIFIED
3423  4AB4  C3  B7  4A   10.0          JMP   RCCKX
3424
3425  4AB7  3A  4B  4B   13.0  RCCKX:  LDA   RCMSK         ;GET THE CHANNEL MASK
3426  4ABA  4F           4.0          MOV   C,A           ;SAVE IN C
3427  4ABB  3A  4A  4B   13.0          LDA   RCEDAT        ;GET THE EXPECTED DATA
3428  4ABE  47           4.0          MOV   B,A           ;SAVE IN B
3429  4ABF  A7           4.0          ANA   A             ;SET CONDITION BITS
3430  4AC0  C2  C9  4A   10.0          JNZ   RCCK1         ;GO PROCESS NOT 0
3431  4AC3                             ROUT  EDATA         ;ZERO
 (1)  4AC3  D3  95       10.0                OUT   EDATA           ;WRITE AC INTO EDATA
 (1)  4AC5  7F           4.0                 MOV   A,A             ;RETRY LINK
3432  4AC6  C3  CE  4A   10.0          JMP   RCCK2
3433  4AC9  3E  01        7.0  RCCK1:  MVI   A,1
3434  4ACB                             ROUT  EDATA
 (1)  4ACB  D3  95       10.0                OUT   EDATA           ;WRITE AC INTO EDATA
 (1)  4ACD  7F           4.0                 MOV   A,A             ;RETRY LINK
3435  4ACE  3A  4C  4B   13.0  RCCK2:  LDA   XEND          ;GET THE END MARK REGISTER
3436  4AD1  A1           4.0          ANA   C             ;AND WITH CHANNEL MASK
3437  4AD2  B8           4.0          CMP   B             ;IF RESULT=EXPECTED DATA
```

```
3438  4AD3   CA   DB   4A      10.0          JZ      RCCN1           ;THEN-CONTINUE
3439  4AD6                                  ERRE    RCOUT,RCCN1
 (1)                                        ;FLAG ERROR - WITH EXPECTED DATA "EDATA" VALID
 (1)  4AD6   CD   0C   28      18.0                  CALL    ERLPE
 (1)       0017                                     MSGN    =       MSGN+1          ;PROCESS ERROR - DO 2.3
 (1)  4AD9   17                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4ADA   00                                     .BYTE
 (1)  4ADB   CD   15   28      18.0          RCCN1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4ADE   DA   40   4B      10.0                  JC      RCOUT           ;LOOP ADDRESS IF LOOP SPECIFIED
3440                              ;>END MARK BIT INCORRECT
3441  4AE1   3A   4D   4B      13.0          LDA     XDATA           ;GET THE DATA REGISTER
3442  4AE4   A1                   4.0          ANA     C               ;AND WITH CHANNEL MASK
3443  4AE5   B8                   4.0          CMP     B               ;IF RESULT=EXPECTED DATA
3444  4AE6   CA   EE   4A      10.0          JZ      RCCN2           ;THEN-CONTINUE
3445  4AE9                                  ERRE    RCOUT,RCCN2
 (1)                                        ;FLAG ERROR - WITH EXPECTED DATA "EDATA" VALID
 (1)  4AE9   CD   0C   28      18.0                  CALL    ERLPE
 (1)       0018                                     MSGN    =       MSGN+1          ;PROCESS ERROR - DO 2.3
 (1)  4AEC   18                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4AED   00                                     .BYTE
 (1)  4AEE   CD   15   28      18.0          RCCN2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4AF1   DA   40   4B      10.0                  JC      RCOUT           ;LOOP ADDRESS IF LOOP SPECIFIED
3446                              ;>DATA BIT INCORRECT
3447  4AF4   3A   4E   4B      13.0          LDA     XPOSTN          ;GET THE POSTAMBLE REGISTER
3448  4AF7   A1                   4.0          ANA     C               ;AND WIITH CHANNEL MASK
3449  4AF8   B8                   4.0          CMP     B               ;IF RESULT=EXPECTED DATA
3450  4AF9   CA   01   4B      10.0          JZ      RCCN3           ;THEN-CONTINUE
3451  4AFC                                  ERRE    RCOUT,RCCN3
 (1)                                        ;FLAG ERROR - WITH EXPECTED DATA "EDATA" VALID
 (1)  4AFC   CD   0C   28      18.0                  CALL    ERLPE
 (1)       0019                                     MSGN    =       MSGN+1          ;PROCESS ERROR - DO 2.3
 (1)  4AFF   19                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4B00   00                                     .BYTE
 (1)  4B01   CD   15   28      18.0          RCCN3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4B04   DA   40   4B      10.0                  JC      RCOUT           ;LOOP ADDRESS IF LOOP SPECIFIED
3452                              ;>POSTAMBLE BIT INCORRECT
3453  4B07   3A   4F   4B      13.0          LDA     XILL            ;GET THE ILLEGAL REGISTER
3454  4B0A   A1                   4.0          ANA     C               ;AND WITH CHANNEL MASK
3455  4B0B   B8                   4.0          CMP     B               ;IF RESULT=EXPECTED DATA
3456  4B0C   CA   14   4B      10.0          JZ      RCCN4           ;THEN-CONTINUE
3457  4B0F                                  ERRE    RCOUT,RCCN4
 (1)                                        ;FLAG ERROR - WITH EXPECTED DATA "EDATA" VALID
 (1)  4B0F   CD   0C   28      18.0                  CALL    ERLPE
 (1)       001A                                     MSGN    =       MSGN+1          ;PROCESS ERROR - DO 2.3
 (1)  4B12   1A                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4B13   00                                     .BYTE
 (1)  4B14   CD   15   28      18.0          RCCN4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4B17   DA   40   4B      10.0                  JC      RCOUT           ;LOOP ADDRESS IF LOOP SPECIFIED
3458                              ;>ILLEGAL BIT INCORRECT
3459  4B1A   3A   50   4B      13.0          LDA     XMK2            ;GET THE MARK TWO REGISTER
3460  4B1D   A1                   4.0          ANA     C               ;AND WITH CHANNEL MASK
3461  4B1E   B8                   4.0          CMP     B               ;IF RESULT=EXPECTED DATA
3462  4B1F   CA   27   4B      10.0          JZ      RCCN5           ;THEN-CONTINUE
```

```
3464  4B22                                        ERRE    RCOUT,RCCN5
 (1)                                              ;FLAG ERROR - WITH EXPECTED DATA "EDATA" VALID
 (1)  4B22  CD  0C  28        18.0                        CALL    ERLPE
 (1)        001B                                      MSGN    =       MSGN+1          ;PROCESS ERROR - DO 2.3
 (1)  4B25  1B                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4B26  00                                        .BYTE
 (1)  4B27  CD  15  28        18.0                RCCN5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4B2A  DA  40  4B        10.0                        JC      RCOUT           ;LOOP ADDRESS IF LOOP SPECIFIED
3465                                        ;>MARK TWO BIT INCORRECT
3466  4B2D  3A  51  4B        13.0                LDA     XDON            ;GET THE NOT DONE REGISTER
3467  4B30  A1                  4.0                ANA     C               ;AND WITH CHANNEL MASK
3468  4B31  B8                  4.0                CMP     B               ;IF RESULT=EXPECTED DATA
3469  4B32  CA  3A  4B          10.0                JZ      RCCN6           ;THEN-CONTINUE
3470  4B35                                        ERRE    RCOUT,RCCN6
 (1)                                              ;FLAG ERROR - WITH EXPECTED DATA "EDATA" VALID
 (1)  4B35  CD  0C  28        18.0                        CALL    ERLPE
 (1)        001C                                      MSGN    =       MSGN+1          ;PROCESS ERROR - DO 2.3
 (1)  4B38  1C                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4B39  00                                        .BYTE
 (1)  4B3A  CD  15  28        18.0                RCCN6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4B3D  DA  40  4B        10.0                        JC      RCOUT           ;LOOP ADDRESS IF LOOP SPECIFIED
3471                                        ;>DONE NOT BIT INCORRECT
3472  4B40  C9                  10.0          RCOUT:  RET
```

```
3474                                    .SBTTL  PROGRAM VARIABLES
3475  4B41   00                 UNITMP: .BYTE   0               ;UNIT MAP BYTE
3476  4B42   00                 RUNSTA: .BYTE   0               ;EXPECTED RUN TIME STATUS BYTE
3477  4B43   00                 RUNCMD: .BYTE   0               ;RUN TIME COMMAND BYTE
3478  4B44   00                 DATAA:  .BYTE   0               ;DATA BYTE ACTUAL
3479  4B45   00                 DATAAP: .BYTE   0               ;DATA BYTE ACTUAL PARITY
3480  4B46   00                 DATAE:  .BYTE   0               ;DATA BYTE EXPECTED
3481  4B47   00                 DATAEP: .BYTE   0               ;DATA BYTE EXPECTED PARITY
3482  4B48   00                 SUNIT:  .BYTE   0               ;TAPE UNIT NUMBER
3483  4B49   00                 JAMDAT: .BYTE   0               ;TIE BUS JAM VALUE
3484  4B4A   00                 RCEDAT: .BYTE   0               ;EXPECTED READ CHANNEL DATA
3485  4B4B   00                 RCMSK:  .BYTE   0               ;READ CHANNEL MASK
3486  4B4C   00                 XEND:   .BYTE   0               ;COPY OF REND REGISTER
3487  4B4D   00                 XDATA:  .BYTE   0               ;COPY OF RDATA REGISTER
3488  4B4E   00                 XPOSTN: .BYTE   0               ;COPY OF RPOSTN REGISTER
3489  4B4F   00                 XILL:   .BYTE   0               ;COPY OF RILL REGISTER
3490  4B50   00                 XMK2:   .BYTE   0               ;COPY OF RMK2 REGISTER
3491  4B51   00                 XDON:   .BYTE   0               ;COPY OF RDON REGISTER
3492        0000                         .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
AMTLPO  49E7           ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000         BADST  = 0090          BIT0   = 0001
BIT1   = 0002          BIT15  = 8000         BIT2   = 0004          BIT3   = 0008
BIT4   = 0010          BIT5   = 0020         BIT6   = 0040          BIT7   = 0080
BIT8   = 0100          BIT9   = 0200         BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60           BRKXCT= 4F00          BSAVE   4F9C          BYTCNT= 00D4
BYTEH   4F24           BYTEL   4F23          C      =%0001         CASCT   4F21
CASCTL= 00A0           CASSTA= 00A0          CATTH  = 0089          CATTL  = 0088
CBUSST= 00A1           CBYTH  = 008B         CBYTL  = 008A          CDG1H  = 0087
CDG1L  = 0086          CDG2H  = 0093         CDG2L  = 0092          CDG3H  = 0095
CDG3L  = 0094          CDVTH  = 008D         CDVTL  = 008C          CHPTIE= 0028
CHOTIE= 0020           CH1TIE= C021          CH2TIE= 0022          CH3TIE= 0023
CH4TIE= 0024           CH5TIE= 0025          CH6TIE= 0026          CH7TIE= 0027
CKAMP   4999  G        CKAMT   48BE          CKAM0   48D9  G       CKAM1   48F1  G
CKAM2   4909  G        CKAM3   4921  G       CKAM4   4939  G       CKAM5   4951  G
CKAM6   4969  G        CKAM7   4981  G       CKLOP  = 2815          CLEAR   49A0
CLKCTL= 00F0           CLOCK   4F26          CLRLP   49A4          CMC0H  = 0099
CMC0L  = 0098          CMC1H  = 009B         CMC1L  = 009A          CMC2H  = 009D
CMC2L  = 009C          CMC3H  = 009F         CMC3L  = 009E          CMINH  = 0097
CMINL  = 0096          CNTCTL= 00D7          CRCWRD= 0018          CSAVE   4F9D
CSRLH  = 0091          CSRLL  = 0090         CTCH   = 0085          CTCL   = 0084
CTSTH  = 008F          CTSTL  = 008E         CXCTH  = 0081          CXCTL  = 0080
CXINH  = 0083          CXINL  = 0082         C.     = 0001          C.AVAI= 0080
C.DP   = 0008          C.DSE  = 0010         C.DTU  = 0003          C.DVA  = 0008
C.FAIL= 00FC           C.FMT  = 0070         C.FNCT= 003E          C.GO   = 0001
C.INTC= 00FE           C.MAIN= 0020          C.NSA  = 0080          C.RCT  = 00FC
C.SER  = 0080          C.SHR  = 0040         C.SKPC= 000F          C.TAPE= 0040
C.WCS  = 0002          D      =%0002         DATAA   4B44          DATAAP  4B45
DATACT= 00D0           DATAE   4B46          DATAEP  4B47          DBUS    4F28
DBUSCT= 00C0           DBUSST= 00C0          DDRA   = 00D8          DDRAIN= 0010
DDRB   = 00D9          DDRBIN= 0002          DDRC   = 00DA          DDRCIN= 0001
DDRCU  = 0088          DDRCTL= 00DB          DIAFLG  4F22          DIAGPG= 4300
DIAGRM= 4F90           DIARD  = 000B         DONE1  = 0045          DONINT= 0010
DSAVE   4F9E           DSE    = 0006         DUMMY   4719  G       D.ATH0= 0001
D.ATH1= 0002           D.EOTD= 0010          D.LAGC= 0020          D.NOTW= 0040
D.NTHR= 0004           D.TACH= 0008          D.WR4  = 0080          E      =%0003
ECCBAD= 0042           ECCCON  489E  G       ECCCOR= 0019          ECCC01  4839  G
ECCLOP  4843           ECCL01  47F5          ECCOK  = 0041          ECCSTA= 001A
ECCTST= 000E           EDATA  = 0095         EOTCLR= 0003          ERFLG   4F93
ERLP   = 2809          ERLPA  = 280F         ERLPB  = 2812          ERLPE  = 280C
ERNUM   4F90           ERRCNT= 00D6          ESAVE   4F9F          EXIT    48B8
E.ACRC= 0010           E.AMT  = 0020         E.CDP  = 0080          E.CRC  = 0080
E.PNTR= 0008           E.RPE  = 0040         E.STEC= 0001          E.TTEC= 0002
E.UNC  = 0004          FIFORD= 006A          FORMAT  4F25          FWDTST= 0061
GCRID  = 0089          GCRSET= 0002          GOODTM= 0092          H      =%0004
HLSAVE  4FA0           IE     = 0008         INTSTA= 00E0          ITERA   4F9A
I.PWR  = 0020          I.RMPE= 0040          I5.5   = 0010          I6.5   = 0020
I7.5   = 0040          JAMDAT  4B49          KCALL  = 005F          KCLR   = 007B
KDEP   = 003F          KENAB  = 0078         KEXAM  = 003E          KEYBRD= 00C8
KEY1   = 0078          KEY10  = 006D         KEY11  = 006E          KEY12  = 006F
KEY13  = 005C          KEY14  = 005D         KEY15  = 005E          KEY16  = 005F
KEY17  = 003C          KEY18  = 003D         KEY19  = 003E          KEY2   = 0079
KEY20  = 003F          KEY3   = 007A         KEY4   = 007B          KEY5   = 0074
```

```
KEY6   = 0075    KEY7   = 0076    KEY8   = 0077    KEY9   = 006C
KINTA  = 006F    KLDAD  = 003D    KN0    = 003C    KN1    = 005C
KN2    = 005D    KN3    = 005E    KN4    = 006C    KN5    = 006D
KN6    = 006E    KN7    = 0074    KN8    = 0075    KN9    = 0076
KU2    = 0079    KU3    = 007A    KU8    = 0077    L      =%0005
LBLANK= 000F     LCE    = 000B    LCH    = 000C    LCL    = 000D
LCP    = 000E    LC0    = 0000    LC1    = 0001    LC2    = 0002
LC3    = 0003    LC4    = 0004    LC5    = 0005    LC6    = 0006
LC7    = 0007    LC8    = 0008    LC9    = 0009    LDLEDA= 00CA
LDLEDB= 00CB     LDLEDC= 00CC     LDLEDD= 00CD     LDLEDE= 00CE
LDLEDF= 00CF     LKDIAG= 2800     LKKBD  = 004C    LKKEY  = 0049
LKLWMG= 0058     LKLWMP= 0055     LKLWPG= 0052     LKLWPP= 004F
LKMOD7= 0046     LKOPR  = 0046    LPFLG   4F94     LPNUM   4F92
M      =%0006    MBSEL  = 00E0    MB.A   = 0008    MB.B   = 0004
MEMTOP= 4FFF     MINUS  = 000A    MM     = 8000    MSE    = 0008
MSGN   = 001C    MTACLR= 0000     MT.ARA= 0020     MT.CPE= 0080
MT.DSE= 0001     MT.FWD= 0040     MT.INH= 0008     MT.LWR= 0004
MT.MOT= 0002     MT.NWT= 0080     MT.PEC= 0040     MT.PSB= 0004
MT.PS0= 0001     MT.PS1= 0002     MT.REV= 0020     MT.WRT= 0010
MT.Z   = 0008    M.ATA  = 0080    M.CAPE= 0020     M.CONT= 0080
M.DEM  = 0020    M.EBL  = 0010    M.EXC  = 0008    M.FAIL= 0008
M.ILR  = 0010    M.INIT= 0010     M.OCC  = 0020    M.ONLI= 0001
M.PE   = 0040    M.PORT= 0080     M.RDEN= 0002     M.RDPE= 0008
M.RUN  = 0004    M.SCLK= 0001     M.TRA  = 0040    M.UNIT= 0007
M.WCLK= 0040     M.WCLN= 0080     M.WREN= 0080     M5.5   = 0001
M6.5   = 0002    M7.5   = 0004    NOTCAP= 0088     OKAY   = 00FF
OPRRAM= 4300     OPSTRT= 0058     OPVER  = 0040    OUT     499F
PADCNT= 00D5     PADCRC= 0080     PDIAG  = 0048    PEID   = 008A
PENAB  = 004C    PESET  = 0001    PL     = 00B1    PRDD   = 004C
PRENF  = 009C    PS     = 00B2    PSTAT  = 0048    PSW    =%0009
P.AMTP= 0001     P.BCTC= 0040     P.CMDP= 0020     P.INTE= 0080
P.LCS  = 0040    P.LWR  = 0020    P.RDP  = 0002    P.RPEN= 0004
P.RPST= 0002     P.RP0E= 0020     P.RP1E= 0010     P.RP2E= 0020
P.RP3E= 0010     P.SING= 0080     P.STAT= 0002     P.STPE= 0080
P.TACH= 0008     P.TUPR= 0010     P.WCSP= 0004     P.WDS  = 0040
P.WFLP= 0001     P.WPEN= 0010     P.WP0E= 0008     P.WP1E= 0004
P.WP2E= 0008     P.WP3E= 0004     P.5VOK= 0002     QUE    = 281B
QUEM   = 281E    RAMT   = 0010    RARA   = 0006    RARAI  = 0004
RCCK    4A4B     RCCKP   4A6C     RCCKP1  4A75     RCCKP2  4A81
RCCKP3  4A8D     RCCKP4  4A99     RCCKP5  4AA5     RCCKP6  4AB1
RCCKX   4AB7     RCCK1   4AC9     RCCK2   4ACE     RCCLR   4A19
RCCLR1  4A29     RCCN1   4ADB  G  RCCN2   4AEE  G  RCCN3   4B01  G
RCCN4   4B14  G  RCCN5   4B27  G  RCCN6   4B3A  G  RCEDAT  4B4?
RCHBD0= 0048     RCHBD1= 0047     RCHOK  = 0046    RCHTST= 000C
RCLRT  = 000D    RCMD   = 000B    RCMLP  = 0003    RCMSK   4B4B
RCONT  = 0080    RCOUT   4B40     RCSTRT  4A32     RCSTR1  4A42
RDATA  = 0017    RDCLK  = 0010    RDCTL   49CB     RDON   = 0011
READG  = 0007    REND   = 0014    REQST  = 2806    RESCHR= 00D1
RETURN  4454     REVTST= 0064     REWIND= 0004     RFIFOL= 0008
RGCLK  = 0002    RGCRI  = 0003    RIBG   = 0001    RILL   = 0012
RINST  = 000C    RLOOP1  44BD     RMCCN0  4378  G  RMCCN1  4478  G
RMCCN2  4494  G  RMCCN3  44B7  G  RMCCN4  44D7  G  RMCLP0  4360
RMCLP1  4462     RMCTST= 0008     RMK2   = 0013    RNOP   = 0000
RPATH  = 0001    RPBAD  = 0044    RPCHI  = 0001    RPCLK  = 0003
```

```
RPCTL = 0009            RPEI  = 0002            RPFAIL= 0000            RPF1  = 009D
RPF2  = 0C9E            RPOK  = 0043            RPOSTN= 0016            RPSTA = 0015
RRCMT = 000A            RSTAT = 0002            RTIEB = 000A            RTIER = 0030
RTM   = 0005            RUNCMD  4B43            RUNKI = 0009            RUNSTA  4B42
RUPTST= 005E            RWDUNL= 0005            R.AMT = 0001            R.BOP = 0008
R.DATA= 0040            R.DON = 0002            R.DRDY= 0010            R.END = 0010
R.ILL = 0004            R.JVOK= 0004            R.MK2 = 0008            R.PLOD= 0008
R.PLO0= 0010            R.PLO1= 0020            R.POST= 0020            R.STNM= 0002
R.STOP= 0004            R.STPC= 0001            R.TBJN= 0080            R.TSTD= 0040
R.VOK = 0080            R00H  = 0081            R00L  = 0080            R01H  = 0083
R01L  = 0082            R02H  = 0085            R02L  = 0084            R03H  = 0087
R03L  = 0086            R04H  = 0089            R04L  = 0088            R05H  = 008B
R05L  = 008A            R06H  = 008D            R06L  = 008C            R07H  = 008F
R07L  = 008E            R10H  = 0091            R10L  = 0090            R11H  = 0093
R11L  = 0092            R12H  = 0095            R12L  = 0094            R13H  = 0097
R13L  = 0096            R14H  = 0099            R14L  = 0098            R15H  = 009B
R15L  = 009A            R16H  = 009D            R16L  = 009C            R17H  = 009F
R17L  = 009E            R7.5  = 0010            SELCLR= 0000            SETATA= 00A1
SID   = 0080            SOD   = 0080            SOE   = 0040            SP    =%0008
SSCLK = 0040            SSTEP = 0005            STACK = 4FFF            STACN1  443F
STALP1  4420            STATCK  4412            STATRM= 4F20            STPCT   4F20
STRSP = 5000            SUNIT   4B48            TADR00= 0080            TADR01= 0081
TADR02= 0082            TADR03= 0083            TADR04= 0084            TADR05= 0085
TADR06= 0086            TADR07= 0087            TADR10= 0088            TADR11= 0089
TADR12= 008A            TADR13= 008B            TAMT  = 0044            TASEL = 0080
TCMD  = 0040            TC.FWD= 0040            TC.INH= 0008            TC.LWR= 0004
TC.REV= 0020            TC.WRT= 0010            TEMP    4F99            TEST05  44F1
TEST06  452B            TEST07  4565            TEST1   4300            TEST10  459F
TEST11  45D9            TEST12  4613            TEST13  464D            TEST14  4687
TEST15  46C1            TEST16  46FB            TEST17  4746            TEST2   434F
TEST20  476F            TEST21  4798            TEST22  47C1            TEST3   439D
TEST4   4455            TIECN0  4329  G         TIEL0   4311            TLOOP0  4468
TLOOP2  487B            TLOOP3  4816            TLOOP4  44C7            TMF   = 0099
TMRDY = 0040            TRKENA= 00D2            TSET  = 2803            TSTEND= 2818
TSTS  = 0040            TST01X  4305            TST02X  4354            TST020  4774
TST03A  43A3            TST03B  43B2  C         TST03C  43C8  G         TST03D  43DE  G
TST03E  43F4  G         TST03X  43A2            TST04X  445A            TST05X  44F6
TST05Y  4509            TST06X  4530            TST06Y  4543            TST07X  456A
TST07Y  457D            TST10X  45A4            TST10Y  45B7            TST11X  45DE
TST11Y  45F1            TST12X  4618            TST12Y  462B            TST13X  4652
TST13Y  4665            TST14X  468C            TST14Y  469F            TST15X  46C6
TST15Y  46D9            TST16X  4722            TST17X  474B            TST21X  479D
TST22X  483F            TUSEL0= 00D1            TUSEL1= 00D2            TU78  = 0010
T.ATH0= 0001            T.ATH1= 0002            T.BOT = 0004            T.EOT = 0002
T.FPT = 0001            T.NTHR= 0004            T.ONL = 0020            T.PES = 0008
T.PSBJ= 0020            T.PSOJ= 0008            T.PS1J= 0010            T.RDY = 0080
T.RDYO= 0040            T.RWD = 0010            T.SCLK= 0002            T220UT  47E6
T22ST   47C6            T22WT   4871            UIBG  = 00A1            UNITMP  4B41
VALFC   4F98            VALTB   4F95            VELTST= 0C5B            WDR.P = 0010
WMCCTL= 00D3            WMCERR= 00DA            WMCSTA= 00D0            WRTCLK= 0000
WRTDAT= 00D3            W.ACRC= 0004            W.CRC = 0008            W.DIAG= 0002
W.DONN= 0040            W.ECC = 0010            W.ENAB= 0080            W.ERR = 0020
W.FMT = 0070            W.GCR = 0010            W.LEFT= 0004            W.ONES= 0020
W.RESI= 0002            W.REV = 0004            W.ROME= 0010            W.RST = 0001
```

```
W.SKIP= 000F            W.WRIT= 0008            W.XFER= 0020            X     =%000A
XDATA   4B4D            XDON    4B51            XEND    4B4C            XILL    4B4F
XMK2    4B50            XPOSTN  4B4E            X.DONN= 0080            X.ENAB= 0040
X.PEPE= 0002            X.ROME= 0001            X.WCLK= 0001            Y     =%000B
.     = 4B52
```

```
ERRORS DETECTED:  0

*RPM1.A78/P1P,RPM1=NLIST,PARAM,MACRO,LIST,RPM1
RUN-TIME: 5 9 0 SECONDS
CORE USED:  10K
```

3 "MEMORY MAP" EXTENDED ADDRESS BIT
19 READ MICRO CONTROLLER CONTROL & STATUS WORDS
139 TRANSPORT/PORT COMMAND + STATUS WORDS
380 COMMON ADDRESS SPACE REGISTERS
444 MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485 MASSBUS CONTROL WORD DEFINITIONS
557 WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709 KEYBOARD/DISPLAY CONTROL + STATUS WORD
797 MISCELLANEOUS VARIABLE DEFINITIONS
801 RAM ADDRESS DEFINITIONS
819 LINKAGE TABLE ADDRESSES
842 BIT DEFINITIONS
856 RIM AND SIM INSTRUCTION BIT DEFINITIONS
877 INTERRUPTABLE INSTRUCTION MACROS
924 DIAGNOSTIC VARIABLES
945 COMMON VARIABLE STORAGE
959 MACRO CALL DEFINITIONS
1047 TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101 8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1333 TEST 01 - READ CHANNEL FIFO STATIC TEST/ALL ONES
1486 TEST 02 - READ CHANNEL FIFO STATIC TEST/ALL ZEROS
1640 TEST 03 - READ CHANNEL FIFO STATIC TEST/ALTERNATING ZERO/ONES
1841 TEST 04 - READ DATA PATH TEST - FROM TU PORT #0
1962 TEST 05 - READ DATA PATH TEST - FROM TU PORT #1
2066 TEST 06 - READ DATA PATH TEST - FROM TU PORT #2
2171 TEST 07 - READ DATA PATH TEST - FROM TU PORT #3
2276 TEST 10 - VELOCITY BIT JAM TEST
2342 TEST 11 - VELOCITY TEST - FROM TU PORT #0
2390 TEST 12 - VELOCITY TEST - FROM TU PORT #1
2438 TEST 13 - VELOCITY TEST - FROM TU PORT #2
2487 TEST 14 - VELOCITY TEST - FROM TU PORT #3
2538 SUBROUTINE TACH PULSE GENERATOR
2580 SUBROUTINE VELCTL
2643 SUBROUTINE SET UP THE READ PATH FOR FIFO TEST
2733 SUBROUTINE READ THE FIFO'S
2777 SUBROUTINE CLEAR ALL TU PORTS
2822 SUBROUTINE CHECK AMTIE
2972 TABLE EXPECTED FIFO DATA
2996 PROGRAM VARIABLES

```
961                     ;*************************************************************************
962                     ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
963                     ;ERROR MACRO CALLS
964                     ;
965                     ;       1  -    REQUEST HOST CPU TO PRINT:
966                     ;               "BYTE/SCLK COUNT NUMBER = LLL"
967                     ;               WHERE:
968                     ;               LLL = THE VALUE STORED IN CAS REGISTER 5
969                     ;                     (THE BYTE COUNT REGISTER 16 BITS).
970                     ;
971                     ;       2  -    REQUEST HOST CPU TO PRINT:
972                     ;               DATA FORMAT = MM
973                     ;               SKIP COUNT  = NN
974                     ;               WHERE:
975                     ;               MM = DATA FORMAT FROM CAS REGISTER 2
976                     ;               NN = SKIP COUNT FROM CAS REGISTER 2
977                     ;
978                     ;       3  -    REQUEST HOST CPU TO PRINT:
979                     ;               BYTE-SCLK COUNT = LLL
980                     ;               DATA FORMAT = MM
981                     ;               SKIP COUNT  = NN
982                     ;               WHERE:
983                     ;               LLL = AS ABOVE
984                     ;               MM  - AS ABOVE
985                     ;               NN  = AS ABOVE
986                     ;
987                     ;       4  -    REQUEST HOST TO PRINT:
988                     ;               TRANSITION COUNT = LLL
989                     ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
990                     ;
991                     ;       5  -    REQUEST HOST CPU TO PRINT:
992                     ;               EXPECTED 18 BITS =      E EEEEEE
993                     ;               ACTUAL 18 BITS =        A AAAAAA
994                     ;
995                     ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
996                     ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
997                     ;               OF CAS REG 15 LOW BYTE.
998                     ;
999                     ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
1000                    ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
1001                    ;               17 LOW BYTE SIGN BIT.
1002                    ;
1003                    ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
1004                    ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
1005                    ;               AND/OR ACTUAL DATA.
1006                    ;
1007                    ;       7  -    REQUEST HOST CPU TO PRINT:
1008                    ;               "SUBGROUP NUMBER = LLL"
1009                    ;               WHERE:
1010                    ;               LLL = THE VALUE STORED IN CAS REGISTER 5
1011                    ;                     (THE BYTE COUNT REGISTER 16 BITS).
1012                    ;
1013                    ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
1014                    ;*************************************************************************
```

```
1016                    ; ************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ; ************************************************************************
1035                    ; ************************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ; ************************************************************************
1329    4300                    .-      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  RPM2 - READ PATH MICROCONTROLLER PART #2
                                .SBTTL  TEST 01 - READ CHANNEL FIFO STATIC TEST/ALL ONES
                                ;ID RPM2-READ PATH MICRO CONTROLLER PART #2

4300                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*READ CHANNEL - FIFO STATIC TEST - ALL ONES
4300                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE FIFO'S IN PLO BYPASS MODE 3 WHERE ALL 9 FIFO'S ARE
                                ;*PARALLEL LOADED WITH ONES FROM THE READ CHANNEL TEST DATA BIT.  ONCE
                                ;*LOADED THE READ PATH RUNS A SPECIAL TEST ROUTINE TO READ ONE BIT FROM
                                ;*EACH FIFO AND LOAD IT IN THE DATA REGISTER.  THEN THIS REGISTER IS
                                ;*COMPARED WITH THE VALUE WRITTEN AND ANY ERRORS ARE REPORTED.  THE FAILING
                                ;*M8950 MODULE CAN BE LOCATED BY THE BIT POSITION OF THE FAILING DATA BIT(S).
4300                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    SET PLO BYPASS MODE 3
                                ;*    CLOCK THE FIFO'S
                                ;*    ISSUE THE CLEAR ALL FIFO'S COMMAND
                                ;*    DELAY 5 MICRO SECONDS
                                ;*    STOP THE READ PATH AND READ CHANNELS CLOCK
                                ;*    SET THE FIFO TEST DATA BIT
                                ;*    CLOCK THE TEST DATA INTO THE FIFO'S
                                ;*    SET THE SUBSYSTEM IN THE PE READ MODE
                                ;*    ISSUE THE FIFO READ COMMAND
                                ;*    ENABLE THE READ PATH AND READ CHANNELS CLOCK
                                ;*    DELAY 5 MICRO SECONDS
                                ;*    STOP THE READ PATH AND READ CHANNELS CLOCK
                                ;*    READ THE FIFO DATA (9 BITS)
                                ;*    SET UP THE EXPECTED DATA OF ALL ONES
                                ;*    CALL SUBROUTINE CKAMT
                                ;*ENDTST
4300                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*RPM2 MICRO TEST 01
                                ;*RPM2 MICRO ERROR 07
                                ;*RPM2-FIFO STATUS TEST-ALL ONES
                                ;*M8950, M8953
                                ;*CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 01
```

```
1374                            ;*RPM2 MICRO ERROR 10
1375                            ;*RPM2-FIFO STATUS TEST-ALL ONES
1376                            ;*M8950, M8953
1377                            ;*CHANNEL 1 FAILED
1378                            ;*ACTUAL = NNNN
1379                            ;*EXPECTED = NNNN
1380                            ;*
1381                            ;*RPM2 MICRO TEST 01
1382                            ;*RPM2 MICRO ERROR 11
1383                            ;*RPM2-FIFO STATUS TEST-ALL ONES
1384                            ;*M8950, M8953
1385                            ;*CHANNEL 2 FAILED
1386                            ;*ACTUAL = NNNN
1387                            ;*EXPECTED = NNNN
1388                            ;*
1389                            ;*RPM2 MICRO TEST 01
1390                            ;*RPM2 MICRO ERROR 12
1391                            ;*RPM2-FIFO STATUS TEST-ALL ONES
1392                            ;*M8950, M8953
1393                            ;*CHANNEL 3 FAILED
1394                            ;*ACTUAL = NNNN
1395                            ;*EXPECTED = NNNN
1396                            ;*
1397                            ;*RPM2 MICRO TEST 01
1398                            ;*RPM2 MICRO ERROR 13
1399                            ;*RPM2-FIFO STATUS TEST-ALL ONES
1400                            ;*M8950, M8953
1401                            ;*CHANNEL 4 FAILED
1402                            ;*ACTUAL = NNNN
1403                            ;*EXPECTED = NNNN
1404                            ;*
1405                            ;*RPM2 MICRO TEST 01
1406                            ;*RPM2 MICRO ERROR 14
1407                            ;*RPM2-FIFO STATUS TEST-ALL ONES
1408                            ;*M8950, M8953
1409                            ;*CHANNEL 5 FAILED
1410                            ;*ACTUAL = NNNN
1411                            ;*EXPECTED = NNNN
1412                            ;*
1413                            ;*RPM2 MICRO TEST 01
1414                            ;*RPM2 MICRO ERROR 15
1415                            ;*RPM2-FIFO STATUS TEST-ALL ONES
1416                            ;*M8950, M8953
1417                            ;*CHANNEL 6 FAILED
1418                            ;*ACTUAL = NNNN
1419                            ;*EXPECTED = NNNN
1420                            ;*
1421                            ;*RPM2 MICRO TEST 01
1422                            ;*RPM2 MICRO ERROR 16
1423                            ;*RPM2-FIFO STATUS TEST-ALL ONES
1424                            ;*M8950, M8953
1425                            ;*CHANNEL 7 FAILED
1426                            ;*ACTUAL = NNNN
1427                            ;*EXPECTED = NNNN
```

```
                                        ;*
                                        ;*RPM2 MICRO TEST 01
                                        ;*RPM2 MICRO ERROR 17
                                        ;*RPM2-FIFO STATUS TEST-ALL ONES
                                        ;*M8950, M8953
                                        ;*CHANNEL P FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
4300                                    S
 (1)                                    ; ************************************************************************
                                        TEST1:  TESTX   @1              ;INITIALIZE THE TEST
4300  3E  01              7.0                       MVI     A,@1              ;DEFINE THE TEST NUMBER
4302  CD  03  28         18.0                       CALL    TSET              ;SETUP THE TEST
                                        ;%RPM2-FIFO STATIC TEST-ALL ONES
                                        ;&M8950, M8953
4305  3E  10              7.0       TST01X: MVI     A,RDCLK           ;SET NORMAL READ PATH CLOCK
4307  D3  F0             10.0               OUT     CLKCTL            ;
4309  3E  30              7.0       FIFOL1: MVI     A,R.PLO1!R.PLO0   ;SET PLO BYPASS MODE 3
430B  D3  09             10.0               OUT     RPCTL             ;
430D  D3  08             10.0               OUT     RFIFOL            ;CLOCK THE FIFO
430F  3E  0D              7.0               MVI     A,RCLRT           ;LOAD ALL FIFO'S CLEAR COMMAND
4311  D3  0B             10.0               OUT     RCMD              ;ISSUE FIFO CLEAR
4313  00                  4.0               NOP                       ;WAIT
4314  00                  4.0               NOP                       ;WAIT
4315  00                  4.0               NOP                       ;WAIT
4316  00                  4.0               NOP                       ;WAIT
4317  3E  71              7.0               MVI     A,R.PLO1!R.PLO0.R.STPC!@100
4319  D3  09             10.0               OUT     RPCTL             ;STOP THE RMC WITH FIFO DATA
                                                                       ;OF ONES
431B  D3  08             10.0               OUT     RFIFOL            ;CLOCK ''1'S'' INTO THE FIFO
431D  AF                 4.0               XRA     A                 ;
431E  D3  D3             10.0               OUT     WMCCTL            ;PUT MACHINE IN PE MODE
4320  3E  6A              7.0               MVI     A,FIFORD          ;LOAD THE FIFO READ COMMAND
4322  D3  0B             10.0               OUT     RCMD              ;
4324  3E  30              7.0               MVI     A,R.PLO0!R.PLO1   ;RUN THE RMC
4326  D3  09             10.0               OUT     RPCTL             ;
4328  00                  4.0               NOP                       ;WAIT
4329  00                  4.0               NOP                       ;WAIT
432A  00                  4.0               NOP                       ;WAIT
432B  00                  4.0               NOP                       ;WAIT
432C  3E  31              7.0               MVI     A,R.PLO0!R.PLO1!R.STPC  ;STOP THE RMC
432E  D3  09             10.0               OUT     RPCTL
4330  DB  17             10.0               IN      RDATA             ;READ THE FIFO DATA
4332  32  45  48         13.0               STA     DATAA             ;STORE ACTUAL DATA
4335  3E  FF              7.0               MVI     A,$FF             ;LOAD EXPECTED DATA
4337  32  43  48         13.0               STA     DATAE             ;STORE EXPECTED DATA
433A  DB  15             10.0               IN      RPSTA             ;GET PARITY BITS
433C  E6  40              7.0               ANI     R.DATA            ;GET DATA PARITY
433E  07                  4.0               RLC                       ;JUSTIFY THE
433F  07                  4.0               RLC                       ;ACTUAL PARITY DATA
4340  32  46  48         13.0               STA     DATAAP            ;SAVE ACTUAL DATA PARITY
4343  3E  01              7.0               MVI     A,$01             ;LOAD EXPECTED DATA
```

```
1479  4345  32  44  48  13.0          STA     DATAEP          ;STORE EXPECTED DATA
1480  4348  CD  49  47  18.0          CALL    CKAMT           ;
1481  434B  DA  09  43  10.0          JC      FIFOL1          ;LOOP ON ERROR
1482  434E                           ENDTST  TST01X
 (1)                         ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  434E                           REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  434E  CD  06  28  18.0                         CALL    REQST
 (2)  4351  00                                       .BYTE                   ;DATA PATTERN NUMBER
 (2)  4352  00  00                                   .WORD                   ;SYSTEM "" COUNT
 (2)  4354  00  00                                   .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4356  00                                       .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4357  07                                       .BYTE   7               ;REQUEST CODE
 (1)  4358  3A  9A  4F  13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  435B  3D                 4.0    DCR     A               ;DOWNCOUNT
 (1)  435C  32  9A  4F  13.0          STA     ITERA           ;SAVE COUNT
 (1)  435F  F2  05  43  10.0          JP      TST01X          ;DO TEST UNTIL TILL = 0
1483
1484
```

```
                                .SBTTL  TEST 02 - READ CHANNEL FIFO STATIC TEST/ALL ZEROS

                                ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*READ CHANNEL - FIFO STATIC TEST - ALL ZEROS
                                SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE FIFO'S IN PLO BYPASS MODE 3 WHERE ALL 9 FIFO'S ARE
                                ;*PARALLEL LOADED WITH ZEROS FROM THE READ CHANNEL TEST DATA BIT.  ONCE
                                ;*LOADED THE READ PATH RUNS A SPECIAL TEST ROUTINE TO READ ONE BIT FROM
                                ;*EACH FIFO AND LOAD IT IN THE DATA REGISTER.  THEN THIS REGISTER IS
                                ;*COMPARED WITH THE VALUE WRITTEN AND ANY ERRORS ARE REPORTED.  THE FAILING
                                ;*M8950 MODULE CAN BE LOCATED BY THE BIT POSITION OF THE FAILING DATA BIT(S).
                                SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    SET PLO BYPASS MODE 3
                                ;*    CLOCK THE FIFO'S
                                ;*    ISSUE THE CLEAR ALL FIFO'S COMMAND
                                ;*    DELAY 5 MICRO SECONDS
                                ;*    STOP THE READ PATH AND READ CHANNELS CLOCK
                                ;*    CLEAR THE FIFO TEST DATA BIT
                                ;*    CLOCK THE TEST DATA INTO THE FIFO'S
                                ;*    SET THE SUBSYSTEM IN THE PE READ MODE
                                ;*    ISSUE THE FIFO READ COMMAND
                                ;*    ENABLE THE READ PATH AND READ CHANNELS CLOCK
                                ;*    DELAY 5 MICRO SECONDS
                                ;*    STOP THE READ PATH AND READ CHANNELS CLOCK
                                ;*    READ THE FIFO DATA (9 BITS)
                                ;*    SET UP THE EXPECTED DATA OF ALL ZEROS
                                ;*    CALL SUBROUTINE CKAMT
                                ;*ENDTST
                                SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM2 MICRO TEST 02
                                ;*RPM2 MICRO ERROR 07
                                ;*RPM2-FIFO STATUS TEST-ALL ZEROS
                                ;*M8950, M8953
                                ;*CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 02
                                ;*RPM2 MICRO ERROR 10
                                ;*RPM2-FIFO STATUS TEST-ALL ZEROS
```

```
1528                    ;*M8950, M8953
1529                    ;*CHANNEL 1 FAILED
1530                    ;*ACTUAL = NNNN
1531                    ;*EXPECTED = NNNN
1532                    ;*
1533                    ;*RPM2 MICRO TEST 02
1534                    ;*RPM2 MICRO ERROR 11
1535                    ;*RPM2-FIFO STATUS TEST-ALL ZEROS
1536                    ;*M8950, M8953
1537                    ;*CHANNEL 2 FAILED
1538                    ;*ACTUAL = NNNN
1539                    ;*EXPECTED = NNNN
1540                    ;*
1541                    ;*RPM2 MICRO TEST 02
1542                    ;*RPM2 MICRO ERROR 12
1543                    ;*RPM2-FIFO STATUS TEST-ALL ZEROS
1544                    ;*M8950, M8953
1545                    ;*CHANNEL 3 FAILED
1546                    ;*ACTUAL = NNNN
1547                    ;*EXPECTED = NNNN
1548                    ;*
1549                    ;*RPM2 MICRO TEST 02
1550                    ;*RPM2 MICRO ERROR 13
1551                    ;*RPM2-FIFO STATUS TEST-ALL ZEROS
1552                    ;*M8950, M8953
1553                    ;*CHANNEL 4 FAILED
1554                    ;*ACTUAL = NNNN
1555                    ;*EXPECTED = NNNN
1556                    ;*
1557                    ;*RPM2 MICRO TEST 02
1558                    ;*RPM2 MICRO ERROR 14
1559                    ;*RPM2-FIFO STATUS TEST-ALL ZEROS
1560                    ;*M8950, M8953
1561                    ;*CHANNEL 5 FAILED
1562                    ;*ACTUAL = NNNN
1563                    ;*EXPECTED = NNNN
1564                    ;*
1565                    ;*RPM2 MICRO TEST 02
1566                    ;*RPM2 MICRO ERROR 15
1567                    ;*RPM2-FIFO STATUS TEST-ALL ZEROS
1568                    ;*M8950, M8953
1569                    ;*CHANNEL 6 FAILED
1570                    ;*ACTUAL = NNNN
1571                    ;*EXPECTED = NNNN
1572                    ;*
1573                    ;*RPM2 MICRO TEST 02
1574                    ;*RPM2 MICRO ERROR 16
1575                    ;*RPM2-FIFO STATUS TEST-ALL ZEROS
1576                    ;*M8950, M8953
1577                    ;*CHANNEL 7 FAILED
1578                    ;*ACTUAL = NNNN
1579                    ;*EXPECTED = NNNN
1580                    ;*
1581                    ;*RPM2 MICRO TEST 02
```

```
                                        ;*RPM2 MICRO ERROR 17
                                        ;*RPM2-FIFO STATUS TEST-ALL ZEROS
                                        ;*M8950, M8953
                                        ;*CHANNEL P FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
1589  4362                              $
 (1)                                    ; ************************************************************************************

                                        TEST2:  TESTX   @2              ;INITIALIZE THE TEST
 (1)  4362  3E  02            7.0                    MVI     A,@2        ;DEFINE THE TEST NUMBER
 (1)  4364  CD  03  28       18.0                    CALL    TSET        ;SETUP THE TEST
                                        ;%RPM2-FIFO STATIC TEST-ALL ZEROS
                                        ;&M8950, M8953
1594  4367  3E  10            7.0  TST02X: MVI     A,RDCLK             ;SET NORMAL READ PATH CLOCK
1595  4369  D3  F0           10.0          OUT     CLKCTL              ;
                                                                       ;
1597  436B  3E  30            7.0  FIFOL2: MVI     A,R.PLO1.R.PLO0     ;SET PLO BYPASS MODE 3
1598  436D  D3  09           10.0          OUT     RPCTL               ;
1599  436F  D3  08           10.0          OUT     RFIFOL              ;CLOCK THE FIFO
1600  4371  3E  0D            7.0          MVI     A,RCLRT             ;LOAD ALL FIFO'S CLEAR COMMAND
1601  4373  D3  0B           10.0          OUT     RCMD                ;ISSUE FIFO CLEAR
1602  4375  00                4.0          NOP                         ;WAIT
1603  4376  00                4.0          NOP                         ;WAIT
1604  4377  00                4.0          NOP                         ;WAIT
1605  4378  00                4.0          NOP                         ;WAIT
1606  4379  3E  31            7.0          MVI     A,R.PLO1!R.PLO0!R.STPC
1607  437B  D3  09           10.0          OUT     RPCTL               ;STOP THE RMC WITH FIFO DATA
1608                                                                   ;OF ONES
1609  437D  D3  08           10.0          OUT     RFIFOL              ;CLOCK ''0'S'' INTO THE FIFO
1610  437F  AF                4.0          XRA     A                   ;
1611  4380  D3  D3           10.0          OUT     WMCCTL              ;PUT MACHINE IN PE MODE
1612  4382  3E  6A            7.0          MVI     A,FIFORD            ;LOAD THE FIFO READ COMMAND
1613  4384  D3  0B           10.0          OUT     RCMD                ;
1614                                                                   ;
1615  4386  3E  30            7.0          MVI     A,R.PLO0!R.PLO1     ;RUN THE RMC
1616  4388  D3  09           10.0          OUT     RPCTL               ;
1617  438A  00                4.0          NOP                         ;WAIT
1618  438B  00                4.0          NOP                         ;WAIT
1619  438C  00                4.0          NOP                         ;WAIT
1620  438D  00                4.0          NOP                         ;WAIT
1621  438E  3E  31            7.0          MVI     A,R.PLO0!R.PLO1!R.STPC  ;STOP THE RMC
1622  4390  D3  09           10.0          OUT     RPCTL
1623  4392  DB  17           10.0          IN      RDATA               ;READ THE FIFO DATA
1624  4394  32  45  48       13.0          STA     DATAA               ;STORE ACTUAL DATA
1625  4397  AF                4.0          XRA     A                   ;LOAD EXPECTED DATA
1626  4398  32  43  48       13.0          STA     DATAE               ;STORE EXPECTED DATA
1627  439B  DB  15           10.0          IN      RPSTA               ;GET PARITY BITS
1628  439D  E6  40            7.0          ANI     R.DATA              ;GET DATA PARITY
1629  439F  07                4.0          RLC                         ;JUSTIFY THE
1630  43A0  07                4.0          RLC                         ;ACTUAL PARITY DATA
1631  43A1  32  46  48       13.0          STA     DATAAP              ;SAVE ACTUAL DATA PARITY
1632  43A4  AF                4.0          XRA     A                   ;LOAD EXPECTED DATA PARITY
```

```
1633  43A5  32  44  48  13.0          STA     DATAEP          ;STORE EXPECTED DATA PARITY
1634  43A8  CD  49  47  18.0          CALL    CKAMT           ;CHECK FOR BAD READ CHANNELS
1635  43AB  DA  6B  43  10.0          JC      FIFOL2          ;LOOP ON ERROR
1636  43AE                           ENDTST  TST02X
 (1)                         ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43AE                           REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43AE  CD  06  28  18.0                         CALL    REQST
 (2)  43B1  00                                       .BYTE                   ;DATA PATTERN NUMBER
 (2)  43B2  00  00                                   .WORD                   ;SYSTEM '"' COUNT
 (2)  43B4  00  00                                   .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43B6  00                                       .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  43B7  07                                       .BYTE   7               ;REQUEST CODE
 (1)  43B8  3A  9A  4F  13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  43BB  3D           4.0          DCR     A               ;DOWNCOUNT
 (1)  43BC  32  9A  4F  13.0          STA     ITERA           ;SAVE COUNT
 (1)  43BF  F2  67  43  10.0          JP      TST02X          ;DO TEST UNTIL TILL = 0
1637
1638
```

```
                                .SBTTL  TEST 03 - READ CHANNEL FIFO STATIC TEST/ALTERNATING ZERO/ONES
1641  43C2                      ST
 (1)                            ;*****************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1642                            ;*READ CHANNEL - FIFO STATIC TEST - ALTERNATING ZERO/ONES
1643  43C2                      SD
 (1)                            ;*****************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1644                            ;*THIS TEST CHECKS THE FIFO'S IN PLO BYPASS MODE 3 WHERE ALL 9 FIFO'S ARE
1645                            ;*PARALLEL LOADED WITH ALTERNATE ZEROS/ONES FROM THE READ CHANNEL TEST
1646                            ;*DATA BIT.  ONCE LOADED THE READ PATH RUNS A SPECIAL TEST ROUTINE TO
1647                            ;*READ ONE BIT FORM EACH FIFO AND LOAD IT IN THE DATA REGISTER.  THEN
1648                            ;*THIS REGISTER IS COMPARED WITH THE VALUE WRITTEN AND ANY ERRORS ARE
1649                            ;*REPORTED.  THE FAILING M8950 MODULE CAN BE LOCATED BY THE BIT POSITION
1650                            ;*OF THE FAILING DATA BIT(S).
1651  43C2                      SP
 (1)                            ;*****************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1652                            ;*BGNTST
1653                            ;*     SET NORMAL READ PATH CLOCK
1654                            ;*     SET THE READ PASS COUNT TO 1
1655                            ;*     SET PLO BYPASS MODE 3
1656                            ;*     CLOCK THE FIFO'S
1657                            ;*     ISSUE CLEAR ALL FIFO'S COMMAND
1658                            ;*     DELAY 5 MICRO SECONDS
1659                            ;*     SET THE LOOP COUNTER TO 5
1660                            ;*     BGNDO
1661                            ;*        :     LOAD FIFO DATA OF ZERO
1662                            ;*        :     CLOCK THE DATA INTO THE FIFO
1663                            ;*        :     LOAD FIFO DATA OF ONE
1664                            ;*        :     CLOCK THE DATA INTO THE FIFO
1665                            ;*        :     DECREMENT THE LOOP COUNTER
1666                            ;*        :     DO UNTIL LOOP COUNTER=ZERO
1667                            ;*     ENDDO
1668                            ;*     BGNDO
1669                            ;*        :     PUT THE SUBSYSTEM IN READ PE MODE
1670                            ;*        :     ISSUE THE FIFO READ COMMAND
1671                            ;*        :     ENABLE THE READ PATH AND READ CHANNEL CLOCKS
1672                            ;*        :     DELAY 5 MICRO SECONDS
1673                            ;*        :     DISABLE THE READ PATH AND READ CHANNEL CLOCKS
1674                            ;*        :     READ THE FIFO DATA (9 BITS)
1675                            ;*        :     GET THE EXPECTED FIFO DATA FROM THE EXPECTED DATA TABLE
1676                            ;*        :     CALL SUBROUTINE "CKAMT"
1677                            ;*        :     INCREMENT THE READ PASS COUNT
1678                            ;*        :     DO UNTIL THE READ PASS COUNT = 13(8)
1679                            ;*     ENDDO
1680                            ;*ENDTST
1681  43C2                      SE
 (1)                            ;*****************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
```

```
1682                    ;*RPM2 MICRO TEST 03
1683                    ;*RPM2 MICRO ERROR 07
1684                    ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
1685                    ;*M8950, M8953
1686                    ;*CHANNEL 0 FAILED
1687                    ;*ACTUAL = NNNN
1688                    ;*EXPECTED = NNNN
1689                    ;*
1690                    ;*RPM2 MICRO TEST 03
1691                    ;*RPM2 MICRO ERROR 10
1692                    ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
1693                    ;*M8950, M8953
1694                    ;*CHANNEL 1 FAILED
1695                    ;*ACTUAL = NNNN
1696                    ;*EXPECTED = NNNN
1697                    ;*
1698                    ;*RPM2 MICRO TEST 03
1699                    ;*RPM2 MICRO ERROR 11
1700                    ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
1701                    ;*M8950, M8953
1702                    ;*CHANNEL 2 FAILED
1703                    ;*ACTUAL = NNNN
1704                    ;*EXPECTED = NNNN
1705                    ;*
1706                    ;*RPM2 MICRO TEST 03
1707                    ;*RPM2 MICRO ERROR 12
1708                    ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
1709                    ;*M8950, M8953
1710                    ;*CHANNEL 3 FAILED
1711                    ;*ACTUAL = NNNN
1712                    ;*EXPECTED = NNNN
1713                    ;*
1714                    ;*RPM2 MICRO TEST 03
1715                    ;*RPM2 MICRO ERROR 13
1716                    ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
1717                    ;*M8950, M8953
1718                    ;*CHANNEL 4 FAILED
1719                    ;*ACTUAL = NNNN
1720                    ;*EXPECTED = NNNN
1721                    ;*
1722                    ;*RPM2 MICRO TEST 03
1723                    ;*RPM2 MICRO ERROR 14
1724                    ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
1725                    ;*M8950, M8953
1726                    ;*CHANNEL 5 FAILED
1727                    ;*ACTUAL = NNNN
1728                    ;*EXPECTED = NNNN
1729                    ;*
1730                    ;*RPM2 MICRO TEST 03
1731                    ;*RPM2 MICRO ERROR 15
1732                    ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
1733                    ;*M8950, M8953
1734                    ;*CHANNEL 6 FAILED
1735                    ;*ACTUAL = NNNN
```

```
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM2 MICRO TEST 03
                                        ;*RPM2 MICRO ERROR 16
                                        ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
                                        ;*M8950, M8953
                                        ;*CHANNEL 7 FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM2 MICRO TEST 03
                                        ;*RPM2 MICRO ERROR 17
                                        ;*RPM2-FIFO STATUS TEST-ALTERNATE ONES/ZEROS
                                        ;*M8950, M8953
                                        ;*CHANNEL P FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
43C2                                    $
 (1)                                    ; ************************************************************************************

                                        TEST3:  TESTX   @3              ;INITIALIZE THE TEST
 (1)  43C2  3E  03          7.0                  MVI     A,@3            ;DEFINE THE TEST NUMBER
 (1)  43C4  CD  03  28     18.0                  CALL    TSET            ;SETUP THE TEST
                                        ;%RPM2-FIFO STATIC TEST-ALTERNATE ONES/ZEROS
                                        ;&M8950, M8953

43C7  3E  10               7.0  TST03X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCK
43C9  D3  F0              10.0         OUT     CLKCTL          ;
43CB  AF                   4.0         XRA     A               ;CLEAR THE FAILING PASS NUMBER
43CC  32  49  48          13.0         STA     PASFL
43CF  3E  01               7.0  1$:     MVI     A,@1            ;LOAD THE BEGINNING PASS COUNTER
43D1  32  48  48          13.0         STA     PASCT           ;SAVE PASS COUNTER
43D4  3E  30               7.0         MVI     A,R.PLO1!R.PLO0 ;SET PLO BYPASS MODE 3
43D6  D3  09              10.0         OUT     RPCTL           ;
43D8  D3  08              10.0         OUT     RFIFOL          ;CLOCK THE FIFO
43DA  3E  0D               7.0         MVI     A,RCLRT         ;LOAD ALL FIFO'S CLEAR COMMAND
43DC  D3  0B              10.0         OUT     RCMD            ;ISSUE FIFO CLEAR
43DE  00                   4.0         NOP                     ;WAIT
43DF  00                   4.0         NOP                     ;WAIT
43E0  00                   4.0         NOP                     ;WAIT
43E1  00                   4.0         NOP                     ;WAIT
43E2  06  05               7.0         MVI     B,@5            ;LOAD CONTROL COUNT
43E4  3E  31               7.0  2$:     MVI     A,R.PLO0!R.PLO1!R.STPC  ;LOAD FIFO DATA OF ZERO
43E6  D3  09              10.0         OUT     RPCTL
43E8  D3  08              10.0         OUT     RFIFOL          ;CLOCK-ZEROS INTO THE FIFOS
43EA  3E  71               7.0         MVI     A,R.PLO0!R.PLO1!R.STPC!@100  ;LOAD FIFO DATA OF ONE
43EC  D3  09              10.0         OUT     RPCTL
43EE  D3  08              10.0         OUT     RFIFOL          ;CLOCK-ONES INTO THE FIFOS
43F0  05                   4.0         DCR     B
43F1  C2  E4  43          10.0         JNZ     2$

43F4  21  2E  48          10.0         LXI     H,FIFOEX        ;LOAD START OF EXPECTED DATA TBL.
43F7  3A  48  48          13.0  3$:     LDA     PASCT           ;WRITE THE PASS COUNT TO CAS REGISTER 5
```

```
1787  43FA                                  ROUT    ROSL
 (1)  43FA  D3  8A          10.0                    OUT     ROSL            ;WRITE AC INTO ROSL
 (1)  43FC  7F               4.0                    MOV     A,A             ;RETRY LINK
1788  43FD  AF               4.0            XRA     A               ;CLEAR CAS REGISTER 5 HIGH
1789  43FE                                  ROUT    ROSH
 (1)  43FE  D3  8B          10.0                    OUT     ROSH            ;WRITE AC INTO ROSH
 (1)  4400  7F               4.0                    MOV     A,A             ;RETRY LINK
1790  4401  AF               4.0            XRA     A               ;
1791  4402  D3  D3          10.0            OUT     WMCCTL          ;PUT MACHINE IN PE MODE
1792  4404  3E  6A           7.0            MVI     A,FIFORD        ;LOAD THE FIFO READ COMMAND
1793  4406  D3  0B          10.0            OUT     RCMD            ;
1794  4408  3E  30           7.0            MVI     A,R.PL00!R.PL01 ;RUN THE RMC
1795  440A  D3  09          10.0            OUT     RPCTL           ;
1796  440C  00               4.0            NOP                     ;WAIT
1797  440D  00               4.0            NOP                     ;WAIT
1798  440E  00               4.0            NOP                     ;WAIT
1799  440F  00               4.0            NOP                     ;WAIT
1800  4410  3E  31           7.0            MVI     A,R.PL00.R.PL01!R.STPC  ;STOP THE RMC
1801  4412  D3  09          10.0            OUT     RPCTL
1802  4414  DB  17          10.0            IN      RDATA           ;READ THE FIFO DATA
1803  4416  32  45  48      13.0            STA     DATAA           ;STORE ACTUAL DATA
1804  4419  7E               7.0            MOV     A,M             ;GET EXPECTED DATA
1805  441A  32  43  48      13.0            STA     DATAE           ;STORE EXPECTED DATA
1806  441D  23               6.0            !NX     H               ;UPDATE TABLE POINTER
1807  441E  DB  15          10.0            IN      RPSTA           ;GET PARITY BITS
1808  4420  E6  40           7.0            ANI     R.DATA          ;GET DATA PARITY
1809  4422  07               4.0            RLC                     ;JUSTIFY PARITY BIT
1810  4423  07               4.0            RLC                     ;
1811  4424  32  46  48      13.0            STA     DATAAP          ;SAVE ACTUAL DATA PARITY
1812  4427  7E               7.0            MOV     A,M             ;GET EXPECTED DATA PARITY
1813  4428  32  44  48      13.0            STA     DATAEP          ;SAVE EXPECTED DATA PARITY
1814  442B  E5              12.0            PUSH    H               ;SAVE H
1815  442C  CD  49  47      18.0            CALL    CKAMT           ;CHECK FOR BAD
1816  442F  E1              10.0            POP     H               ;RESTORE H
1817  4430  D2  4E  44      10.0            JNC     4$              ;NO ERROR CONTINUE
1818  4433  3A  49  48      13.0            LDA     PASFL           ;GET THE FAILING PASS NUMBER
1819  4436  A7               4.0            ANA     A               ;SET THE CONDITION BITS
1820  4437  CA  45  44      10.0            JZ      5$              ;FIRST ERROR GO INITIALIZE THE LOOP
1821  443A  47               4.0            MOV     B,A             ;SAVE THE FAILING PASS NUMBER
1822  443B  3A  48  48      13.0            LDA     PASCT           ;GET THE NUMBER OF THIS PASS
1823  443E  B8               4.0            CMP     B               ;EQUAL?
1824  443F  C2  4E  44      10.0            JNZ     4$              ;NO - CONTINUE WITH THE TEST
1825  4442  C3  CF  43      10.0            JMP     1$              ;YES - RESTART THE TEST
1826  4445  3A  48  48      13.0  5$:       LDA     PASCT           ;GET THE PASS COUNT
1827  4448  32  49  48      13.0            STA     PASFL           ;STORE AS THE FAILING PASS NUMBER
1828  444B  C3  CF  43      10.0            JMP     1$              ;RESTART THE TEST
1829  444E  23               6.0  4$:       INX     H               ;UPDATE POINTER
1830  444F  3A  48  48      13.0            LDA     PASCT           ;GET THE PASS COUNT
1831  4452  3C               4.0            INR     A               ;UPDATE THE COUNTER
1832  4453  32  48  48      13.0            STA     PASCT           ;RESTORE TO MEMORY
1833  4456  01  42  48      10.0            LXI     B,FIFOFN        ;GET END OF TABLE
1834  4459  79               4.0            MOV     A,C
1835  445A  BD               4.0            CMP     L               ;DONE?
1836  445B  C2  F7  43      10.0            JNZ     3$              ;NO
```

```
1837  445E                                        ENDTST  TST03X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  445E                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  445E  CD  06  28        18.0                          CALL    REQST
 (2)  4461  00                                              .BYTE                   ;DATA PATTERN NUMBER
 (2)  4462  00  00                                          .WORD                   ;SYSTEM '"' COUNT
 (2)  4464  00  00                                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4466  00                                              .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4467  07                                              .BYTE   7               ;REQUEST CODE
 (1)  4468  3A  9A  4F        13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  446B  3D                 4.0          DCR     A               ;DOWNCOUNT
 (1)  446C  32  9A  4F        13.0          STA     ITERA           ;SAVE COUNT
 (1)  446F  F2  C7  43        10.0          JP      TST03X          ;DO TEST UNTIL TILL = 0
1838
1839
```

```
                                .SBTTL  TEST 04 - READ DATA PATH TEST - FROM TU PORT #0
1842  4472                      ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1843                            ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1844  4472                      SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1845                            ;*THIS TEST CHECKS THE DATA PATH FROM TAPE UNIT PORT 0 TO THE READ
1846                            ;*CHANNELS.
1847  4472                      SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1848                            ;*BGNTST
1849                            ;*    IF TU PORT #0 SELECTED BY USER
1850                            ;*        THEN-CONTINUE
1851                            ;*        ELSE-NEXT TEST
1852                            ;*    ENDIF
1853                            ;*    CALL SUBROUTINE CLEAR
1854                            ;*    LOAD TU PORT #0 SELECT CODE
1855                            ;*    CALL SUBROJTINE RDSET
1856                            ;*ENDTST
1857  4472                      SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1858                            ;*RPM2 MICRO TEST 04
1859                            ;*RPM2 MICRO ERROR 01
1860                            ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1861                            ;*M8950, M8953, M8955'S
1862                            ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1863                            ;*FATAL ERROR - TEST ABORTED!
1864                            ;*
1865                            ;*RPM2 MICRO TEST 04
1866                            ;*RPM2 MICRO ERROR 07
1867                            ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1868                            ;*M8950, M8953, M8955'S
1869                            ;*CHANNEL 0 FAILED
1870                            ;*ACTUAL = NNNN
1871                            ;*EXPECTED = NNNN
1872                            ;*
1873                            ;*RPM2 MICRO TEST 04
1874                            ;*RPM2 MICRO ERROR 10
1875                            ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1876                            ;*M8950, M8953, M8955'S
1877                            ;*CHANNEL 1 FAILED
1878                            ;*ACTUAL = NNNN
1879                            ;*EXPECTED = NNNN
1880                            ;*
1881                            ;*RPM2 MICRO TEST 04
1882                            ;*RPM2 MICRO ERROR 11
```

```
1883                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1884                    ;*M8950, M8953, M8955'S
1885                    ;*CHANNEL 2 FAILED
1886                    ;*ACTUAL = NNNN
1887                    ;*EXPECTED = NNNN
1888                    ;*
1889                    ;*RPM2 MICRO TEST 04
1890                    ;*RPM2 MICRO ERROR 12
1891                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1892                    ;*M8950, M8953, M8955'S
1893                    ;*CHANNEL 3 FAILED
1894                    ;*ACTUAL = NNNN
1895                    ;*EXPECTED = NNNN
1896                    ;*
1897                    ;*RPM2 MICRO TEST 04
1898                    ;*RPM2 MICRO ERROR 13
1899                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1900                    ;*M8950, M8953, M8955'S
1901                    ;*CHANNEL 4 FAILED
1902                    ;*ACTUAL = NNNN
1903                    ;*EXPECTED = NNNN
1904                    ;*
1905                    ;*RPM2 MICRO TEST 04
1906                    ;*RPM2 MICRO ERROR 14
1907                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1908                    ;*M8950, M8953, M8955'S
1909                    ;*CHANNEL 5 FAILED
1910                    ;*ACTUAL = NNNN
1911                    ;*EXPECTED = NNNN
1912                    ;*
1913                    ;*RPM2 MICRO TEST 04
1914                    ;*RPM2 MICRO ERROR 15
1915                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1916                    ;*M8950, M8953, M8955'S
1917                    ;*CHANNEL 6 FAILED
1918                    ;*ACTUAL = NNNN
1919                    ;*EXPECTED = NNNN
1920                    ;*
1921                    ;*RPM2 MICRO TEST 04
1922                    ;*RPM2 MICRO ERROR 16
1923                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1924                    ;*M8950, M8953, M8955'S
1925                    ;*CHANNEL 7 FAILED
1926                    ;*ACTUAL = NNNN
1927                    ;*EXPECTED = NNNN
1928                    ;*
1929                    ;*RPM2 MICRO TEST 04
1930                    ;*RPM2 MICRO ERROR 17
1931                    ;*RPM2-READ DATA PATH TEST-FROM TU PORT #0
1932                    ;*M8950, M8953, M8955'S
1933                    ;*CHANNEL P FAILED
1934                    ;*ACTUAL = NNNN
1935                    ;*EXPECTED = NNNN
1936                    ;*BYTE/SCLK COUNT NUMBER = LLL
```

```
1937  4472                                  S
 (1)                                        ; ***************************************************************************
1938
1939  4472                                  TEST4:  TESTX   @004            ;INITIALIZE THE TEST
 (1)  4472  3E  04              7.0                     MVI     A,@004          ;DEFINE THE TEST NUMBER
 (1)  4474  CD  03  28         18.0                     CALL    TSET            ;SETUP THE TEST
1940                                    ;%RPM2-READ DATA PATH TEST-FROM TU PORT #0
1941                                    ;&M8950, M8953, M8955'S
1942
1943  4477                                      REQ     @7,0,0,0,0
 (1)  4477  CD  06  28         '8.0                     CALL    REQST
 (1)  447A  00                                          .BYTE   0               ;DATA PATTERN NUMBER
 (1)  447B  00  00                                      .WORD   0               ;SYSTEM "0" COUNT
 (1)  447D  00  00                                      .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  447F  00                                          .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1)  4480  07                                          .BYTE   @7              ;REQUEST CODE
1944
1945  4481                                      RIN     R12L            ;GET THE UNITS DESIRED
 (1)  4481  DB  94              10.0                    IN      R12L            ;READ R12L INTO AC
 (1)  4483  7F                   4.0                    MOV     A,A             ;RETRY LINK
1946  4484  32  47  48          13.0              STA     UNITMP          ;STORE IN MEMORY
1947  4487  A7                    4.0              ANA     A               ;SET THE CONDITION CODE
1948  4488  C2  99  44          10.0              JNZ     TST04X          ;GO TEST SPECIFIED UNITS
1949  448B                                      ERR     EXIT,DUMMY
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  448B  CD  09  28         18.0                     CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0001                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  448E  01                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  448F  00                                          .BYTE
 (1)  4490  CD  15  28         18.0              DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4493  DA  2C  46         10.0                     JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1950                                    ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1951                                    ;>FATAL ERROR - TEST ABORTED!
1952  4496  C3  2C  46         10.0              JMP     EXIT
1953  4499  3A  47  48          13.0      TST04X: LDA     UNITMP          ;GET THE UNIT MAP BYTE
1954  449C  E6  01                7.0              ANI     $01             ;TEST FOR TU PORT #0
1955  449E  CA  BD  44          10.0              JZ      TEST5           ;NO-GO TO NEXT TEST
1956  44A1  CD  1E  47          18.0              CALL    CLEAR           ;
1957  44A4  3E  00                7.0              MVI     A,0             ;
1958
1959  44A6  CD  9B  46          18.0              CALL    RDSET           ;GO SET UP THE READ CHANNELS
1960  44A9                                      ENDTST  TST04X
 (1)                                    ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44A9                                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44A9  CD  06  28         18.0                     CALL    REQST
 (2)  44AC  00                                          .BYTE                   ;DATA PATTERN NUMBER
 (2)  44AD  00  00                                      .WORD                   ;SYSTEM '"' COUNT
 (2)  44AF  00  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  44B1  00                                          .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  44B2  07                                          .BYTE   7               ;REQUEST CODE
 (1)  44B3  3A  9A  4F          13.0              LDA     ITERA           ;GET ITERATION COUNT
 (1)  44B6  3D                   4.0              DCR     A               ;DOWNCOUNT
 (1)  44B7  32  9A  4F          13.0              STA     ITERA           ;SAVE COUNT
 (1)  44BA  F2  99  44          10.0              JP      TST04X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 05 - READ DATA PATH TEST - FROM TU PORT #1
44BD                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
44BD                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE DATA PATH FROM TAPE UNIT PORT 1 TO THE READ
                                ;*CHANNELS.
44BD                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #1 SELECTED BY USER
                                ;*        THEN-CONTINUE
                                ;*        ELSE-NEXT TEST
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD TU PORT #1 SELECT CODE
                                ;*    CALL SUBROUTINE RDSET
                                ;*ENDTST
44BD                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*RPM2 MICRO TEST 05
                                ;*RPM2 MICRO ERROR 07
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 05
                                ;*RPM2 MICRO ERROR 10
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 1 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 05
                                ;*RPM2 MICRO ERROR 11
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 2 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 05
```

```
2004                                ;*RPM2 MICRO ERROR 12
2005                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
2006                                ;*M8950, M8953, M8955'S
2007                                ;*CHANNEL 3 FAILED
2008                                ;*ACTUAL = NNNN
2009                                ;*EXPECTED = NNNN
2010                                ;*
2011                                ;*RPM2 MICRO TEST 05
2012                                ;*RPM2 MICRO ERROR 13
2013                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
2014                                ;*M8950, M8953, M8955'S
2015                                ;*CHANNEL 4 FAILED
2016                                ;*ACTUAL = NNNN
2017                                ;*EXPECTED = NNNN
2018                                ;*
2019                                ;*RPM2 MICRO TEST 05
2020                                ;*RPM2 MICRO ERROR 14
2021                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
2022                                ;*M8950, M8953, M8955'S
2023                                ;*CHANNEL 5 FAILED
2024                                ;*ACTUAL = NNNN
2025                                ;*EXPECTED = NNNN
2026                                ;*
2027                                ;*RPM2 MICRO TEST 05
2028                                ;*RPM2 MICRO ERROR 15
2029                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
2030                                ;*M8950, M8953, M8955'S
2031                                ;*CHANNEL 6 FAILED
2032                                ;*ACTUAL = NNNN
2033                                ;*EXPECTED = NNNN
2034                                ;*
2035                                ;*RPM2 MICRO TEST 05
2036                                ;*RPM2 MICRO ERROR 16
2037                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
2038                                ;*M8950, M8953, M8955'S
2039                                ;*CHANNEL 7 FAILED
2040                                ;*ACTUAL = NNNN
2041                                ;*EXPECTED = NNNN
2042                                ;*
2043                                ;*RPM2 MICRO TEST 05
2044                                ;*RPM2 MICRO ERROR 17
2045                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #1
2046                                ;*M8950, M8953, M8955'S
2047                                ;*CHANNEL P FAILED
2048                                ;*ACTUAL = NNNN
2049                                ;*EXPECTED = NNNN
2050                                ;*BYTE/SCLK COUNT NUMBER = LLL
2051  44BD                          S
 (1)                                ; ***************************************************************************
```

```
2053  44BD                                TEST5:  TESTX   @005            ;INITIALIZE THE TEST
 (1)  44BD  3E  05              7.0                      MVI     A,@005          ;DEFINE THE TEST NUMBER
 (1)  44BF  CD  03  28         18.0                      CALL    TSET            ;SETUP THE TEST
2054                                ;%RPM2-READ DATA PATH TEST-FROM TU PORT #1
2055                                ;&M8950, M8953, M8955'S
2056  44C2  3A  47  48         13.0  TST05X: LDA     UNITMP          ;GET THE UNIT MAP BYTE
2057  44C5  E6  02                7.0          ANI     $02             ;TEST FOR TU PORT #1
2058  44C7  CA  E6  44          10.0          JZ      TEST6           ;NO-GO TO NEXT TEST
2059  44CA  CD  1E  47          18.0          CALL    CLEAR           ;
2060  44CD  3E  01                7.0          MVI     A,@1
2061  44CF  CD  9B  46          18.0          CALL    RDSET           ;GO SET UP THE READ CHANNELS
2062  44D2                                    ENDTST  TST05X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44D2                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44D2  CD  06  28         18.0                      CALL    REQST
 (2)  44D5  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  44D6  00  00                                       .WORD                   ;SYSTEM "" COUNT
 (2)  44D8  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  44DA  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  44DB  07                                           .BYTE   7               ;REQUEST CODE
 (1)  44DC  3A  9A  4F         13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  44DF  3D                 4.0          DCR     A               ;DOWNCOUNT
 (1)  44E0  32  9A  4F         13.0          STA     ITERA           ;SAVE COUNT
 (1)  44E3  F2  C2  44         10.0          JP      TST05X          ;DO TEST UNTIL TILL = 0
2063
2064
```

```
                                .SBTTL  TEST 06 - READ DATA PATH TEST - FROM TU PORT #2
44E6                            ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
44E6                            SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE DATA PATH FROM TAPE UNIT PORT 2 TO THE READ
                                ;*CHANNELS.
44E6                            SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #2 SELECTED BY USER
                                ;*        THEN-CONTINUE
                                ;*        ELSE-NEXT TEST
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD TU PORT #2 SELECT CODE
                                ;*    CALL SUBROUTINE RDSET
                                ;*ENDTST
44E6                            SE
 (1)                            ;*********************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*RPM2 MICRO TEST 06
                                ;*RPM2 MICRO ERROR 07
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 06
                                ;*RPM2 MICRO ERROR 10
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 1 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 06
                                ;*RPM2 MICRO ERROR 11
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 2 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 06
```

```
2108                            ;*RPM2 MICRO ERROR 12
2109                            ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
2110                            ;*M8950, M8953, M8955'S
2111                            ;*CHANNEL 3 FAILED
2112                            ;*ACTUAL = NNNN
2113                            ;*EXPECTED = NNNN
2114                            ;*
2115                            ;*RPM2 MICRO TEST 06
2116                            ;*RPM2 MICRO ERROR 13
2117                            ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
2118                            ;*M8950, M8953, M8955'S
2119                            ;*CHANNEL 4 FAILED
2120                            ;*ACTUAL = NNNN
2121                            ;*EXPECTED = NNNN
2122                            ;*
2123                            ;*RPM2 MICRO TEST 06
2124                            ;*RPM2 MICRO ERROR 14
2125                            ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
2126                            ;*M8950, M8953, M8955'S
2127                            ;*CHANNEL 5 FAILED
2128                            ;*ACTUAL = NNNN
2129                            ;*EXPECTED = NNNN
2130                            ;*
2131                            ;*RPM2 MICRO TEST 06
2132                            ;*RPM2 MICRO ERROR 15
2133                            ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
2134                            ;*M8950, M8953, M8955'S
2135                            ;*CHANNEL 6 FAILED
2136                            ;*ACTUAL = NNNN
2137                            ;*EXPECTED = NNNN
2138                            ;*
2139                            ;*RPM2 MICRO TEST 06
2140                            ;*RPM2 MICRO ERROR 16
2141                            ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
2142                            ;*M8950, M8953, M8955'S
2143                            ;*CHANNEL 7 FAILED
2144                            ;*ACTUAL = NNNN
2145                            ;*EXPECTED = NNNN
2146                            ;*
2147                            ;*RPM2 MICRO TEST 06
2148                            ;*RPM2 MICRO ERROR 17
2149                            ;*RPM2-READ DATA PATH TEST-FROM TU PORT #2
2150                            ;*M8950, M8953, M8955'S
2151                            ;*CHANNEL P FAILED
2152                            ;*ACTUAL = NNNN
2153                            ;*EXPECTED = NNNN
2154                            ;*BYTE/SCLK COUNT NUMBER = LLL
2155  44E6                      S
 (1)                            ; ***************************************************************************************
```

```
2157  44E6                                TEST6:  TESTX   @006            ;INITIALIZE THE TEST
 (1)  44E6  3E  06              7.0                      MVI     A,@006          ;DEFINE THE TEST NUMBER
 (1)  44E8  CD  03  28         18.0                      CALL    TSET            ;SETUP THE TEST
2158                                ;%RPM2-READ DATA PATH TEST-FROM TU PORT #2
2159                                ;&M8950, M8953, M8955'S
2160
2161  44EB  3A  47  48         13.0  TST06X: LDA     UNITMP          ;GET THE UNIT MAP BYTE
2162  44EE  E6  04                7.0          ANI     $04             ;TEST FOR TU PORT #2
2163  44F0  CA  0F  45          10.0          JZ      TEST7           ;NO-GO TO NEXT TEST
2164  44F3  CD  1E  47          18.0          CALL    CLEAR
2165  44F6  3E  02                7.0          MVI     A,@2
2166  44F8  CD  9B  46          18.0          CALL    RDSET           ;GO SET UP THE READ CHANNELS
2167  44FB                                    ENDTST  TST06X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44FB                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44FB  CD  06  28         18.0                      CALL    REQST
 (2)  44FE  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  44FF  00  00                                       .WORD                   ;SYSTEM '"' COUNT
 (2)  4501  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4503  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4504  07                                           .BYTE   7               ;REQUEST CODE
 (1)  4505  3A  9A  4F         13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4508  3D                 4.0          DCR     A               ;DOWNCOUNT
 (1)  4509  32  9A  4F         13.0          STA     ITERA           ;SAVE COUNT
 (1)  450C  F2  EB  44         10.0          JP      TST06X          ;DO TEST UNTIL TILL = 0
2168
2169
```

```
                                .SBTTL  TEST 07 - READ DATA PATH TEST - FROM TU PORT #3
450F                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
450F                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE DATA PATH FROM TAPE UNIT PORT 3 TO THE READ
                                ;*CHANNELS.
450F                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #3 SELECTED BY USER
                                ;*        THEN-CONTINUE
                                ;*        ELSE-NEXT TEST
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD TU PORT #3 SELECT CODE
                                ;*    CALL SUBROUTINE RDSET
                                ;*ENDTST
450F                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM2 MICRO TEST 07
                                ;*RPM2 MICRO ERROR 07
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 0 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 07
                                ;*RPM2 MICRO ERROR 10
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 1 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 07
                                ;*RPM2 MICRO ERROR 11
                                ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
                                ;*M8950, M8953, M8955'S
                                ;*CHANNEL 2 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*RPM2 MICRO TEST 07
```

```
2213                            ;*RPM2 MICRO ERROR 12
2214                            ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
2215                            ;*M8950, M8953, M8955'S
2216                            ;*CHANNEL 3 FAILED
2217                            ;*ACTUAL = NNNN
2218                            ;*EXPECTED = NNNN
2219                            ;*
2220                            ;*RPM2 MICRO TEST 07
2221                            ;*RPM2 MICRO ERROR 13
2222                            ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
2223                            ;*M8950, M8953, M8955'S
2224                            ;*CHANNEL 4 FAILED
2225                            ;*ACTUAL = NNNN
2226                            ;*EXPECTED = NNNN
2227                            ;*
2228                            ;*RPM2 MICRO TEST 07
2229                            ;*RPM2 MICRO ERROR 14
2230                            ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
2231                            ;*M8950, M8953, M8955'S
2232                            ;*CHANNEL 5 FAILED
2233                            ;*ACTUAL = NNNN
2234                            ;*EXPECTED = NNNN
2235                            ;*
2236                            ;*RPM2 MICRO TEST 07
2237                            ;*RPM2 MICRO ERROR 15
2238                            ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
2239                            ;*M8950, M8953, M8955'S
2240                            ;*CHANNEL 6 FAILED
2241                            ;*ACTUAL = NNNN
2242                            ;*EXPECTED = NNNN
2243                            ;*
2244                            ;*RPM2 MICRO TEST 07
2245                            ;*RPM2 MICRO ERROR 16
2246                            ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
2247                            ;*M8950, M8953, M8955'S
2248                            ;*CHANNEL 7 FAILED
2249                            ;*ACTUAL = NNNN
2250                            ;*EXPECTED = NNNN
2251                            ;*
2252                            ;*RPM2 MICRO TEST 07
2253                            ;*RPM2 MICRO ERROR 17
2254                            ;*RPM2-READ DATA PATH TEST-FROM TU PORT #3
2255                            ;*M8950, M8953, M8955'S
2256                            ;*CHANNEL P FAILED
2257                            ;*ACTUAL = NNNN
2258                            ;*EXPECTED = NNNN
2259                            ;*BYTE/SCLK COUNT NUMBER = LLL
2260  450F                      S
 (1)                            ; ***************************************************************************************
2261
```

```
2263  450F                                  TEST7:  TESTX   @007            ;INITIALIZE THE TEST
 (1)  450F  3E  07              7.0                      MVI     A,@007          ;DEFINE THE TEST NUMBER
 (1)  4511  CD  03  28         18.0                      CALL    TSET            ;SETUP THE TEST
2264                                   ;%RPM2-READ DATA PATH TEST-FROM TU PORT #3
2265                                   ;&M8950, M8953, M8955'S
2266  4514  3A  47  48         13.0    TST07X: LDA     UNITMP          ;GET THE UNIT MAP BYTE
2267  4517  F6  10               7.0           ANI     $010            ;TEST FOR TU PORT #3
2268  4519  CA  38  45         10.0           JZ      TEST10          ;NO-EXIT TEST
2269  451C  CD  1E  47         18.0           CALL    CLEAR
2270  451F  3E  03               7.0           MVI     A,@3
2271  4521  CD  9B  46         18.0           CALL    RDSET           ;GO SET UP THE READ CHANNELS
2272  4524                                   ENDTST  TST07X
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4524                             REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4524  CD  06  28         18.0                      CALL    REQST
 (2)  4527  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  4528  00  00                                       .WORD                   ;SYSTEM '*' COUNT
 (2)  452A  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  452C  00                                           .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  452D  07                                           .BYTE   7               ;REQUEST CODE
 (1)  452E  3A  9A  4F         13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1)  4531  3D               4.0           DCR     A               ;DOWNCOUNT
 (1)  4532  32  9A  4F         13.0           STA     ITERA           ;SAVE COUNT
 (1)  4535  F2  14  45         10.0           JP      TST07X          ;DO TEST UNTIL TILL = 0
2273
2274
```

```
                                .SBTTL  TEST 10 - VELOCITY BIT JAM TEST

                                ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*VELOCITY BIT JAM TEST
                                SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST ASSURES THAT THE VELOCITY OK BIT CAN BE SET BY WRITING THE
                                ;*VELOCITY OK JAM BIT IN THE READ PATH CONTROL WORD.
                                SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    STOP THE READ PATH AND READ CHANNELS
                                ;*    SET THE JAM VELOCITY OK BIT
                                ;*    INPUT THE VELOCITY OK STATUS BIT
                                ;*      IF VELOCITY OK STATUS BIT=1
                                ;*          THEN-CONTINUE
                                ;*          ELSE-ERROR
                                ;*      ENDIF
                                ;*    CLEAR THE JAM VELOCITY OK BIT
                                ;*    DELAY 50 MICRO SECONDS
                                ;*    INPUT THE VELOCITY OK STATUS BIT
                                ;*      IF VELOCITY OK STATUS BIT=0
                                ;*          THEN-CONTINUE
                                ;*          ELSE-ERROR
                                ;*      ENDIF
                                ;*ENDTST
                                SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM2 MICRO TEST 10
                                ;*RPM2 MICRO ERROR 02
                                ;*RPM2-VELOCITY LOGIC TEST
                                ;*M8953
                                ;*VELOCITY "OK" BIT NOT SET WHEN JAMMED SET
                                ;*
                                ;*RPM2 MICRO TEST 10
                                ;*RPM2 MICRO ERROR 03
                                ;*RPM2-VELOCITY LOGIC TEST
                                ;*M8953
                                ;*VELOCITY "OK" BIT STILL SET WHEN JAMMED RESET
                                S
                                ; ***************************************************************************
```

```
2315  4538                                  TEST10: TESTX   @10              ;INITIALIZE THE TEST
 (1)  4538  3E  08              7.0                    MVI     A,@10           ;DEFINE THE TEST NUMBER
 (1)  453A  CD  03  28         18.0                    CALL    TSET            ;SETUP THE TEST
2316                                  ;%RPM2-VELOCITY LOGIC TEST
2317                                  ;&M8953
2318  453D  3E  10              7.0  TST10X: MVI     A,RDCLK         ;SET THE NORMAL READ PATH CLOCK
2319  453F  D3  F0             10.0          OUT     CLKCTL          ;
2320  4541  3E  05              7.0  VOKLP0: MVI     A,R.STPC.R.JVOK ;STOP THE READ MACHINE AND SET
2321  4543  D3  09             10.0          OUT     RPCTL           ;THE VELOCITY OK BIT.
2322  4545  00                  4.0          NOP
2323  4546  DB  01             10.0          IN      RPCHI           ;GET THE VELOCITY OK BIT
2324  4548  E6  80              7.0          ANI     R.VOK           ;REMOVE UNWANTED BITS
2325  454A  C2  52  45          10.0          JNZ     VOKCN0
2326  454D                                  ERR     VOKLP0,VOKCN0
 (1)                                  ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  454D  CD  09  28         18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                               MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4550  02                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4551  00                                .BYTE
 (1)  4552  CD  15  28         18.0          VOKCN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4555  DA  41  45         10.0                    JC      VOKLP0          ;LOOP ADDRESS IF LOOP SPECIFIED
2327                                  ;>VELOCITY "OK" BIT NOT SET WHEN JAMMED SET
2328
2329  4558  3E  01              7.0  VOKLP1: MVI     A,R.STPC                ;RESET THE VELOCITY OK
2330  455A  D3  09             10.0          OUT     RPCTL           ;JAM BIT
2331  455C  3E  0A              7.0          MVI     A,10            ;LOAD A 50 MICRO SECOND TIMEOUT
2332  455E  3D                  4.0  1$:     DCR     A               ;DECREMENT THE COUNTER
2333  455F  C2  5E  45          10.0          JNZ     1$              ;CONTINUE UNTIL DONE
2334  4562  DB  01             10.0          IN      RPCHI           ;GET THE VELOCITY OK CHECK BYTE
2335  4564  E6  80              7.0          ANI     R.VOK           ;VELOCITY OK?
2336  4566  CA  6E  45          10.0          JZ      VOKCN1          ;YES - CONTINUE
2337  4569                                  ERR     VOKLP1,VOKCN1
 (1)                                  ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4569  CD  09  28         18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0003                               MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  456C  03                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  456D  00                                .BYTE
 (1)  456E  CD  15  28         18.0          VOKCN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4571  DA  58  45         10.0                    JC      VOKLP1          ;LOOP ADDRESS IF LOOP SPECIFIED
2338                                  ;>VELOCITY "OK" BIT STILL SET WHEN JAMMED RESET
2339  4574                                  ENDTST  TST10X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4574                                  REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4574  CD  06  28         18.0                    CALL    REQST
 (2)  4577  00                                .BYTE                   ;DATA PATTERN NUMBER
 (2)  4578  00  00                            .WORD                   ;SYSTEM "" COUNT
 (2)  457A  00  00                            .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  457C  00                                .BYTE                   ;DATA COMPARE FLAG IF  1
 (2)  457D  07                                .BYTE   7               ;REQUEST CODE
 (1)  457E  3A  9A  4F         13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4581  3D                  4.0          DCR     A               ;DOWNCOUNT
 (1)  4582  32  9A  4F         13.0          STA     ITERA           ;SAVE COUNT
 (1)  4585  F2  3D  45         10.0          JP      TST10X          ;DO TEST UNTIL TILL = 0
2340
```

```
                                .SBTTL  TEST 11 - VELOCITY TEST - FROM TL PORT #0
 4588                           ST
  (1)                           ; ***************************************************************************
  (1)                           ;*TEST TITLE
  (1)                           ;*---- -----
                                ;*VELOCITY TEST - FROM TU PORT #0
 4588                           SD
  (1)                           ; ***************************************************************************
  (1)                           ;*DESCRIPTION
  (1)                           ;*-----------
                                ;*THIS TEST CHECKS THE VELOCITY LOGIC BY SIMULATING VELOCITY OK AND
                                ;*VELOCITY NOT OK CONDITIONS.  THE SIMULATION FOR THIS TEST IS DONE BY
                                ;*DRIVING THE 'TACH' SIGNAL FROM TAPE UNIT PORT 0.
 4588                           SP
  (1)                           ; ***************************************************************************
  (1)                           ;*PROCEDURE
  (1)                           ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #0 SELECTED BY USER
                                ;*      THEN-CONTINUE
                                ;*      ELSE-NEXT TEST
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD THE PORT #0 CONTROL ADDRESS
                                ;*    CALL SUBROUTINE VELCTL
                                ;*ENDTST
 4588                           SE
  (1)                           ; ***************************************************************************
  (1)                           ;*ERRORS
  (1)                           ;*------
                                ;*RPM2 MICRO TEST 11
                                ;*RPM2 MICRO ERROR 04
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #0
                                ;*M8953, M8955'S
                                ;*VELOCITY OK SET AT -25%
                                ;*
                                ;*RPM2 MICRO TEST 11
                                ;*RPM2 MICRO ERROR 05
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #0
                                ;*M8953, M8955'S
                                ;*VELOCITY OK NOT SET AT NOMINAL
                                ;*
                                ;*RPM2 MICRO TEST 11
                                ;*RPM2 MICRO ERROR 06
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #0
                                ;*M8953, M8955'S
                                ;*VELOCITY OK SET AT +25%
 4588                           S
  (1)                           ; ***************************************************************************
```

```
2379  4588                                  TEST11: TESTX   @11              ;INITIALIZE THE TEST
 (1)  4588  3E  09              7.0                      MVI     A,@11          ;DEFINE THE TEST NUMBER
 (1)  458A  CD  03  28         18.0                      CALL    TSET           ;SETUP THE TEST
2380                                  ;%RPM2-VELOCITY TEST-FROM TU PORT #0
2381                                  ;@M8953, M8955'S
2382  458D  3A  47  48         13.0  TST11X: LDA     UNITMP          ;GET THE UNIT MAP BYTE
2383  4590  E6  01                7.0          ANI     $01             ;TEST TU PORT #0
2384  4592  CA  B1  45         10.0          JZ      TEST12          ;NO-GO TO NEXT TEST
2385  4595  CD  1E  47         18.0          CALL    CLEAR           ;CLEAR ALL TU PORTS
2386  4598  3E  00                7.0          MVI     A,@0
2387  459A  CD  50  46         18.0          CALL    VELCTL
2388  459D                                  ENDTST  TST11X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  459D                                  REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  459D  CD  06  28         18.0                      CALL    REQST
 (2)  45A0  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  45A1  00  00                                       .WORD                   ;SYSTEM "" COUNT
 (2)  45A3  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  45A5  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  45A6  07                                           .BYTE   7               ;REQUEST CODE
 (1)  45A7  3A  9A  4F         13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  45AA  3D                 4.0          DCR     A               ;DOWNCOUNT
 (1)  45AB  32  9A  4F         13.0          STA     ITERA           ;SAVE COUNT
 (1)  45AE  F2  8D  45         10.0          JP      TST11X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 12 - VELOCITY TEST - FROM TU PORT #1
                               ST
                               ; ************************************************************************
                               ;*TEST TITLE
                               ;*---- -----
                               ;*RPM2-VELOCITY TEST-FROM TU PORT #1
                               SD
                               ; ************************************************************************
                               ;*DESCRIPTION
                               ;*-----------
                               ;*THIS TEST CHECKS THE VELOCITY LOGIC BY SIMULATING VELOCITY OK AND
                               ;*VELOCITY NOT OK CONDITIONS.  THE SIMULATION FOR THIS TEST IS DONE BY
                               ;*DRIVING THE 'TACH' SIGNAL FROM TAPE UNIT PORT 1.
                               SP
                               ; ************************************************************************
                               ;*PROCEDURE
                               ;*---------
                               ;*BGNTST
                               ;*    IF TU PORT #1 SELECTED BY USER
                               ;*      THEN-CONTINUE
                               ;*      ELSE-NEXT TEST
                               ;*    ENDIF
                               ;*    CALL SUBROUTINE CLEAR
                               ;*    LOAD THE PORT #1 CONTROL ADDRESS
                               ;*    CALL SUBROUTINE VELCTL
                               ;*ENDTST
                               SE
                               ; ************************************************************************
                               ;*ERRORS
                               ;*------
                               ;*RPM2 MICRO TEST 12
                               ;*RPM2 MICRO ERROR 04
                               ;*RPM2-VELOCITY TEST-FROM TU PORT #1
                               ;*M8953, M8955'S
                               ;*VELOCITY OK SET AT -25%
                               ;*
                               ;*RPM2 MICRO TEST 12
                               ;*RPM2 MICRO ERROR 05
                               ;*RPM2-VELOCITY TEST-FROM TU PORT #1
                               ;*M8953, M8955'S
                               ;*VELOCITY OK NOT SET AT NOMINAL
                               ;*
                               ;*RPM2 MICRO TEST 12
                               ;*RPM2 MICRO ERROR 06
                               ;*RPM2-VELOCITY TEST-FROM TU PORT #1
                               ;*M8953, M8955'S
                               ;*VELOCITY OK SET AT +25%
                               S
                               ; ************************************************************************
```

```
2427  45B1                                TEST12: TESTX   @12              ;INITIALIZE THE TEST
 (1)  45B1  3E  0A              7.0                      MVI     A,@12      ;DEFINE THE TEST NUMBER
 (1)  45B3  CD  03  28         18.0                      CALL    TSET       ;SETUP THE TEST
2428                                ;%RPM2-VELOCITY TEST-FROM TU PORT #1
2429                                ;&M8953, M8955'S
2430  45B6  3A  47  48         13.0 TST12X: LDA     UNITMP                 ;GET THE UNIT MAP BYTE
2431  45B9  E6  02                7.0          ANI     $02                ;TEST TU PORT #1
2432  45BB  CA  DA  45          10.0          JZ      TEST13             ;NO-GO TO NEXT TEST
2433  45BE  CD  1E  47          18.0          CALL    CLEAR              ;CLEAR ALL TU PORTS
2434  45C1  3E  01                7.0          MVI     A,@1
2435  45C3  CD  50  46          18.0          CALL    VELCTL
2436  45C6                                    ENDTST  TST12X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  45C6                                    REQ     7                  ;FAKE CALL TO KEEP TEST ALIVE
 (2)  45C6  CD  06  28         18.0                      CALL    REQST
 (2)  45C9  00                                           .BYTE              ;DATA PATTERN NUMBER
 (2)  45CA  00  00                                       .WORD              ;SYSTEM "" COUNT
 (2)  45CC  00  00                                       .WORD              ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  45CE  00                                           .BYTE              ;DATA COMPARE FLAG IF =1
 (2)  45CF  07                                           .BYTE   7          ;REQUEST CODE
 (1)  45D0  3A  9A  4F         13.0          LDA     ITERA              ;GET ITERATION COUNT
 (1)  45D3  3D                 4.0          DCR     A                  ;DOWNCOUNT
 (1)  45D4  32  9A  4F         13.0          STA     ITERA              ;SAVE COUNT
 (1)  45D7  F2  B6  45         10.0          JP      TST12X             ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 13 - VELOCITY TEST - FROM TU PORT #2
45DA                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #2
45DA                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE VELOCITY LOGIC BY SIMULATING VELOCITY OK AND
                                ;*VELOCITY NOT OK CONDITIONS.  THE SIMULATION FOR THIS TEST IS DONE BY
                                ;*DRIVING THE 'TACH' SIGNAL FROM TAPE UNIT PORT 2.
45DA                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #2 SELECTED BY USER
                                ;*      THEN-CONTINUE
                                ;*      ELSE-NEXT TEST
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD THE PORT #2 CONTROL ADDRESS
                                ;*    CALL SUBROUTINE VELCTL
                                ;*ENDTST
45DA                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*RPM2 MICRO TEST 13
                                ;*RPM2 MICRO ERROR 04
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #2
                                ;*M8953, M8955'S
                                ;*VELOCITY OK SET AT -25%
                                ;*
                                ;*RPM2 MICRO TEST 13
                                ;*RPM2 MICRO ERROR 05
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #2
                                ;*M8953, M8955'S
                                ;*VELOCITY OK NOT SET AT NOMINAL
                                ;*
                                ;*RPM2 MICRO TEST 13
                                ;*RPM2 MICRO ERROR 06
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #2
                                ;*M8953, M8955'S
                                ;*VELOCITY OK SET AT +25%
45DA                            S
 (1)                            ; ************************************************************************
```

```
2475  45DA                                  TEST13: TESTX   @13              ;INITIALIZE THE TEST
 (1)  45DA  3E  0B           7.0                          MVI     A,@13       ;DEFINE THE TEST NUMBER
 (1)  45DC  CD  03  28      18.0                          CALL    TSET        ;SETUP THE TEST
2476                                   ;%RPM2-VELOCITY TEST-FROM TU PORT #2
2477                                   ;&M8953, M8955'S
2478
2479  45DF  3A  47  48      13.0  TST13X: LDA     UNITMP          ;GET THE UNIT MAP BYTE
2480  45E2  E6  04           7.0           ANI     $04             ;TEST TU PORT #2
2481  45E4  CA  03  46      10.0           JZ      TEST14          ;NO-GO TO NEXT TEST
2482  45E7  CD  1E  47      18.0           CALL    CLEAR           ;CLEAR ALL TU PORTS
2483  45EA  3E  02           7.0           MVI     A,@2
2484  45EC  CD  50  46      18.0           CALL    VELCTL
2485  45EF                                 ENDTST  TST13X
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  45EF                             REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  45EF  CD  06  28      18.0                          CALL    REQST
 (2)  45F2  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2)  45F3  00  00                                        .WORD                   ;SYSTEM '"' COUNT
 (2)  45F5  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  45F7  00                                            .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  45F8  07                                            .BYTE   7               ;REQUEST CODE
 (1)  45F9  3A  9A  4F      13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1)  45FC  3D               4.0           DCR     A               ;DOWNCOUNT
 (1)  45FD  32  9A  4F      13.0           STA     ITERA           ;SAVE COUNT
 (1)  4600  F2  DF  45      10.0           JP      TST13X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 14 - VELOCITY TEST - FROM TU PORT #3
4603                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #3
4603                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE VELOCITY LOGIC BY SIMULATING VELOCITY OK AND
                                ;*VELOCITY NOT OK CONDITIONS.  THE SIMULATION FOR THIS TEST IS DONE BY
                                ;*DRIVING THE 'TACH' SIGNAL FROM TAPE UNIT PORT 3.
4603                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    IF TU PORT #3 SELECTED BY USER
                                ;*      THEN-CONTINUE
                                ;*      ELSE-NEXT TEST
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    LOAD THE PORT #3 CONTROL ADDRESS
                                ;*    CALL SUBROUTINE VELCTL
                                ;*ENDTST
4603                            SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM2 MICRO TEST 14
                                ;*RPM2 MICRO ERROR 04
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #3
                                ;*M8953, M8955'S
                                ;*VELOCITY OK SET AT -25%
                                ;*
                                ;*RPM2 MICRO TEST 14
                                ;*RPM2 MICRO ERROR 05
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #3
                                ;*M8953, M8955'S
                                ;*VELOCITY OK NOT SET AT NOMINAL
                                ;*
                                ;*RPM2 MICRO TEST 14
                                ;*RPM2 MICRO ERROR 06
                                ;*RPM2-VELOCITY TEST-FROM TU PORT #3
                                ;*M8953, M8955'S
                                ;*VELOCITY OK SET AT +25%
4603                            S
                                ; ***************************************************************************
```

```
2524  4603                                  TEST14: TESTX   @14              ;INITIALIZE THE TEST
 (1)  4603  3E  0C            7.0                        MVI     A,@14       ;DEFINE THE TEST NUMBER
 (1)  4605  CD  03  28       18.0                        CALL    TSET        ;SETUP THE TEST
2525                                  ;%RPM2-VELOCITY TEST-FROM TU PORT #3
2526                                  ;&M8953, M8955'S
2527  4608  3A  47  48       13.0  TST14X: LDA     UNITMP          ;GET THE UNIT MAP BYTE
2528  460B  E6  10             7.0         ANI     $10             ;TEST TU PORT #3
2529  460D  CA  2C  46         10.0         JZ      EXIT            ;NO-GO TO NEXT TEST
2530  4610  CD  1E  47         18.0         CALL    CLEAR           ;CLEAR ALL TU PORTS
2531  4613  3E  03             7.0         MVI     A,@3
2532  4615  CD  50  46         18.0         CALL    VELCTL
2533  4618                               ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (1)                                       ENDTST  TST14X
 (2)  4618                                 REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4618  CD  06  28       18.0                        CALL    REQST
 (2)  461B  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  461C  00  00                                       .WORD                   ;SYSTEM '"' COUNT
 (2)  461E  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4620  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4621  07                                           .BYTE   7               ;REQUEST CODE
 (1)  4622  3A  9A  4F       13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1)  4625  3D                4.0         DCR     A               ;DOWNCOUNT
 (1)  4626  32  9A  4F       13.0         STA     ITERA           ;SAVE COUNT
 (1)  4629  F2  08  46       10.0         JP      TST14X          ;DO TEST UNTIL TILL = 0
2534
2535  462C  CD  1E  47         18.0  EXIT:   CALL    CLEAR           ;CLEAR ALL PORTS BEFORE LEAVING
2536  462F  C3  18  28         10.0         JMP     TSTEND          ;END OF TEST
```

```
                                .SBTTL  SUBROUTINE TACH PULSE GENERATOR
                                SSUB
                                ; ***************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*TACH PULSE GENERATOR
                                SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE ISSUES 4 TACH PULSES FROM THE TAPE UNIT PORT AT A
                                ;*PULSE RATE CALCUALTED BY THE INTEGER IN MEMORY LOCATION ''TICNM'' TIMES
                                ;*5 MICROSECONDS.
                                SP
                                ; ******* ***************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    LOAD THE NUMBER OF TACH PULSES TO BE GENERATED
                                ;*    BGNDO
                                ;*       :    OUTPUT THE TACH BIT=1 FROM THE SPECIFIED TU PORT
                                ;*       :    OUTPUT THE TACH BIT=0 FROM THE SPECIFIED TU PORT
                                ;*       :    DELAY FOR THE TIME SPECIFIED BY ''TICNM'' X 5 MICRO SECONDS
                                ;*       :    DECREMENT THE TACH PULSE COUNTER
                                ;*       :    DO UNTIL THE TACH PULSE COUNTER = ZERO
                                ;*    ENDDO
                                ;*ENDTST
                                S
                                ; ***************************************************************************

                                ;
 4632 06 04          7.0  VELOK:  MVI   B,4              ;LOAD THE TACH TICK COUNT
 4634 3E 04          7.0          MVI   A,P.RPEN
 4636 D3 4C         10.0          OUT   PENAB
 4638 3E 28          7.0  1$:     MVI   A,P.LWR!P.TACH   ;LOAD THE TACH BIT
 463A D3 48         10.0          OUT   PDIAG            ;SET THE TACH BIT
 463C 3E 20          7.0          MVI   A,P.LWR          ;CLEAR THE TACH PULSE BIT
 463E D3 48         10.0          OUT   PDIAG

                                ;PERFORM A TIMEOUT

 4640 3A 4A 48      13.0          LDA   TICNM            ;GET THE TICK COUNTER

 4643 3D             4.0  2$:     DCR   A                ;WAIT UNTIL ZERO
 4644 C2 43 46      10.0          JNZ   2$               ;
                                                         ;DECREMENT THE TICK COUNT
 4647 05             4.0          DCR   B                ;CONTINUE UNTIL ALL
 4648 C2 38 46      10.0          JNZ   1$               ;TACKS ISSUED.
                                                         ;
 464B DB 01         10.0          IN    RPCHI            ;GET THE VELOCITY CHECK WORD
 464D E6 80          7.0          ANI   R.VOK            ;VELOCITY OK?
 464F C9            10.0          RET                    ;RETURN TO CALLER
```

```
                                .SBTTL  SUBROUTINE VELCTL
                               SSUB
                               ; ***************************************************************************
                               ;*SUBROUTINE TITLE
                               ;*---------- -----
                               ;*SUBROUTINE VELOCITY CONTROL
                               SD
                               ; ***************************************************************************
                               ;*DESCRIPTION
                               ;*-----------
                               ;*THIS SUBROUTINE SIMULATES TACH PULSES FROM THE TAPE TRANSPORT AT -25%
                               ;*OF NOMINAL, NOMINAL (40 MICRO SECONDS), AND +25% OF NOMINAL.  THE VEL-
                               ;*OCITY OK STATUS BIT IS CHECKED AND THE FOLLOWING RESULTS ARE EXPECTED.
                               ;*FAILS (VELOCITY OK = 0) AT -25% OF NOMINAL
                               ;*PASSES (VELOCITY OK = 1) AT NOMINAL
                               ;*FAILS (VELOCITY OK = 0) AT +25% OF NOMINAL
                               SP
                               ; ***************************************************************************
                               ;*PROCEDURE
                               ;*---------
                               ;*BGNSUB
                               ;*    LOAD THE 30 MICRO SECOND TICK RATE VARIABLE
                               ;*    CALL SUBROUTINE "VELOK"
                               ;*      IF VELOCITY=OK
                               ;*      :   THEN-ERROR
                               ;*      :   ELSE-CONTINUE
                               ;*      ENDIF
                               ;*    LOAD THE 40 MICRO SECOND TICK RATE VARIABLE
                               ;*    CALL SUBROUTINE "VELOK"
                               ;*      IF VELOCITY=OK
                               ;*      :   THEN-CONTINUE
                               ;*      :   ELSE-ERROR
                               ;*      ENDIF
                               ;*    LOAD THE 50 MICRO SECOND TICK RATE VARIABLE
                               ;*    CALL SUBROUTINE "VELOK"
                               ;*      IF VELOCITY=OK
                               ;*      :   THEN-ERROR
                               ;*      :   ELSE-CONTINUE
                               ;*      ENDIF
                               ;*ENDSUB
                               S
                               ; ***************************************************************************

4650  47              4.0  VELCTL: MOV     B,A
4651  DB  E0         10.0         IN      INTSTA
4653  E6  80          7.0         ANI     BIT7
4655  B0              4.0         ORA     B
4656  D3  E0         10.0         OUT     MBSEL
4658  3E  02          7.0  TSTVLO: MVI     A,2             ;LOAD THE 30 MICRO SECOND TICK RATE
465A  32  4A  48      13.0         STA     TICNM           ;SAVE IT
465D  CD  32  46      18.0         CALL    VELOK           ;CALL THE VELOCITY ROUTINE
4660  CA  68  46      10.0         JZ      VELCNO          ;CONTINUE IF VELOCITY OK RESET
```

```
2623  4663                                    ERR     TSTVL0,VELCN0
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4663  CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0004                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4666  04                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4667  00                                       .BYTE
 (1)  4668  CD  15  28        18.0               VELCN0::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  466B  DA  58  46        10.0                       JC      TSTVL0          ;LOOP ADDRESS IF LOOP SPECIFIED
2624                                  ;>VELOCITY OK SET AT -25%
2625  466E  3E  04              7.0  TSTVL1: MVI     A,4             ;LOAD A 40 MICRO SECOND TICK RATE
2626  4670  32  4A  48        13.0          STA     TICNM           ;SAVE IT
2627  4673  CD  32  46        18.0          CALL    VELOK           ;CALL THE VELOCITY ROUTINE
2628  4676  C2  7E  46        10.0          JNZ     VELCN1          ;CONTINUE IF VELOCITY OK SET
2629  4679                                  ERR     TSTVL1,VELCN1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4679  CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0005                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  467C  05                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  467D  00                                       .BYTE
 (1)  467E  CD  15  28        18.0               VELCN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4681  DA  6E  46        10.0                       JC      TSTVL1          ;LOOP ADDRESS IF LOOP SPECIFIED
2630                                  ;>VELOCITY OK NOT SET AT NOMINAL
2631
2632
2633  4684  3E  07              7.0  TSTVL2: MVI     A,7             ;LOAD A 50 MICROSECOND TICK RATE
2634  4686  32  4A  48        13.0          STA     TICNM           ;SAVE IT
2635  4689  CD  32  46        18.0          CALL    VELOK           ;CALL VELOCITY ROUTINE
2636  468C  CA  94  46        10.0          JZ      VELCN2          ;CONTINUE IF VELOCITY OK RESET
2637  468F                                  ERR     TSTVL2,VELCN2
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  468F  CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0006                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4692  06                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4693  00                                       .BYTE
 (1)  4694  CD  15  28        18.0               VELCN2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4697  DA  84  46        10.0                       JC      TSTVL2          ;LOOP ADDRESS IF LOOP SPECIFIED
2638                                  ;>VELOCITY OK SET AT +25%
2639  469A  C9                10.0          RET
2640
2641
```

```
                                .SBTTL  SUBROUTINE SET UP THE READ PATH FOR FIFO TEST
469B                            SSUB
 (1)                            ; ************************************************************************
 (1)                            ;*SUBROUTINE TITLE
 (1)                            ;*---------- -----
                                ;*SET UP READ PATH FOR FIFO TEST - FROM PORT
469B                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS SUBROUTINE SETS THE READ CHANNELS UP FOR THE FIFO DATA TEST
                                ;*FROM THE PORTS.
469B                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNSUB
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    SET PLO BYPASS MODE 2
                                ;*    CLOCK THE FIFO'S
                                ;*    ISSUE CLEAR ALL FIFO'S COMMAND
                                ;*    WAIT 5 MICROSECONDS
                                ;*    STOP THE READ PATH AND READ CHANNELS
                                ;*    CLEAR THE EXPECTED DATA
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    DISABLE THE TRANSLATOR OUTPUTS
                                ;*    ISSUE TU PORT CONTROL COMMAND (DPEN, LCS, LWR)
                                ;*    BGNDO
                                ;*       :    WRITE EXPECTED DATA TO THE READ DATA LINES
                                ;*       :    CALCULATE THE PARITY BIT FOR THE EXPECTED DATA
                                ;*       :    WRITE EXPECTED DATA PARITY TO THE READ DATA LIENS
                                ;*       :    CALL SUBROUTINE RDPLO2
                                ;*       :    CALL SUBROUTINE CKAMT
                                ;*       :    INCREMENT THE EXPECTED DATA BYTE
                                ;*       :    DO UNTIL THE EXPECTED DATA BYTE=ZERO
                                ;*    ENDDO
                                ;*ENDSUB
469B                            S
 (1)                            ; ************************************************************************

469B  47              4.0  RDSET:  MOV     B,A
469C  3E  10          7.0          MVI     A,RDCLK          ;GET NORMAL READ PATH CLOCK
469E  D3  F0         10.0          OUT     CLKCTL           ;SET UP THE CLOCK
46A0  3E  20          7.0          MVI     A,R.PLO1         ;GET PLO BYPASS MODE 2
46A2  D3  09         10.0          OUT     RPCTL            ;SET PLO BYPASS

46A4  D3  08         10.0          OUT     RFIFOL           ;CLOCK THE FIFO
46A6  3E  0D          7.0          MVI     A,RCLRT          ;LOAD ALL FIFO'S CLEAR COMMAND
46A8  D3  0B         10.0          OUT     RCMD             ;ISSUE FIFO CLEAR

46AA  00              4.0          NOP                      ;WAIT
46AB  00              4.0          NOP                      ;WAIT
46AC  00              4.0          NOP                      ;WAIT
46AD  00              4.0          NOP                      ;WAIT
```

```
2687
2688  46AE  3E 21          7.0          MVI   A,R.PLO1!R.STPC ;STOP THE RMC SO THAT
2689  46B0  D3 09         10.0          OUT   RPCTL           ;THE FIFO CAN BE LOADED
2690  46B2  DB E0         10.0          IN    INTSTA          ;
2691  46B4  E6 80          7.0          ANI   BIT7            ;
2692  46B6  B0             4.0          ORA   B               ;
2693  46B7  D3 E0         10.0          OUT   MBSEL           ;
2694  46B9  AF             4.0          XRA   A               ;CLEAR THE ACCUMULATOR
2695  46BA  32 43 48      13.0          STA   DATAE           ;STORE AS EXPECTED DATA
2696  46BD  D3 D2         10.0          OUT   TRKENA          ;DISABLE THE TRANSLATOR OUTPUTS
2697  46BF  D3 D2         10.0          OUT   TRKENA
2698  46C1                              ROUT  R05H
 (1)  46C1  D3 8B         10.0                OUT   R05H             ;WRITE AC INTO R05H
 (1)  46C3  7F            4.0                 MOV   A,A              ;RETRY LINK
2699  46C4  3E 60          7.0          MVI   A,P.LCS!P.LWR
2700  46C6  D3 48         10.0          OUT   PDIAG           ;LOOP THE 8085 TO READ PATH
2701  46C8  3E 04          7.0          MVI   A,P.RPEN
2702  46CA  D3 4C         10.0          OUT   PENAB
2703
2704  46CC  3A 43 48      13.0  3$:     LDA   DATAE           ;GET THE EXPECTED DATA
2705  46CF  D3 40         10.0          OUT   TCMD            ;PUT ON THE READ DATA LINES
2706  46D1                              ROUT  R05L            ;WRITE THE EXPECTED DATA TO CAS REGISTER 5
 (1)  46D1  D3 8A         10.0                OUT   R05L             ;WRITE AC INTO R05L
 (1)  46D3  7F            4.0                 MOV   A,A              ;RETRY LINK
2707
2708                                  ;CALCULATE EXPECTED DATA PARITY
2709  46D4  A7             4.0          ANA   A               ;SET CONDITION BITS
2710  46D5  E2 DD 46      10.0          JPO   1$              ;ODD NUMBER OF ONES
2711  46D8  3E 01          7.0          MVI   A,$01           ;ELSE-EVEN NUMBER OF ONES
2712  46DA  C3 DE 46      10.0          JMP   2$              ;PARITY BIT SHOULD BE ONE
2713
2714  46DD  AF             4.0  1$:     XRA   A               ;PARITY BIT SHOULD BE ZERO
2715  46DE  32 44 48      13.0  2$:     STA   DATAEP          ;SAVE EXPECTED DATA PARITY
2716  46E1  07             4.0          RLC                   ;SHIFT THE PARITY BIT TO THE RIGHT PLACE
2717  46E2  F6 60          7.0          ORI   P.LCS!P.LWR
2718  46E4  D3 48         10.0          OUT   PDIAG
2719
2720  46E6  CD FA 46      18.0          CALL  RDPLO2          ;GO CLOCK DATA THROUGH THE FIFO
2721  46E9  CD 49 47      18.0          CALL  CKAMT           ;GO CHECK FOR BAD READ CHANNELS
2722  46EC  DA CC 46      10.0          JC    3$              ;LOOP ON ERROR
2723
2724  46EF  3A 43 48      13.0          LDA   DATAE           ;GET EXPECTED DATA
2725  46F2  3C             4.0          INR   A               ;UPDATE THE EXPECTED DATA
2726  46F3  32 43 48      13.0          STA   DATAE           ;SAVE EXPECTED DATA
2727  46F6  C2 CC 46      10.0          JNZ   3$              ;CONTINUE UNTIL DONE
2728
2729  46F9  C9            10.0          RET
2730
2731
```

```
                                .SBTTL  SUBROUTINE READ THE FIFO'S
46FA                            SSUB
 (1)                            ; ***************************************************************************
 (1)                            ;*SUBROUTINE TITLE
 (1)                            ;*---------- -----
                                ;*READ THE FIFO'S SUBROUTINE
46FA                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*BGNSUB
                                ;*    CLOCK THE DATA FROM THE TU PORT INTO THE FIFO'S
                                ;*    PUT THE TM78 SUBSYSTEM IN READ PE MODE
                                ;*    ISSUE THE FIFO READ COMMAND
                                ;*    ENABLE THE READ PATH AND READ CHANNEL CLOCK
                                ;*    DELAY 5 MICROSECONDS
                                ;*    DISABLE THE READ PATH AND READ CHANNEL CLOCK
                                ;*    READ THE FIFO DATA (9 BITS)
                                ;*ENDSUB
46FA                            S
 (1)                            ; ***************************************************************************

46FA  D3  08          10.0  RDPL02: OUT     RFIFOL          ;CLOCK THE DATA INTO THE FIFO

46FC  AF                     4.0        XRA     A               ;CLEAR A
46FD  D3  D3                10.0        OUT     WMCCTL          ;PUT THE MACHINE IN PE MODE

46FF  3E  6A                 7.0        MVI     A,FIFORD        ;LOAD THE FIFO READ COMMAND
4701  D3  0B                10.0        OUT     RCMD

4703  3E  20                 7.0        MVI     A,R.PL01        ;LET THE READ MACHINE RUN
4705  D3  09                10.0        OUT     RPCTL

4707  00                     4.0        NOP                     ;WAIT
4708  00                     4.0        NOP                     ;WAIT
4709  00                     4.0        NOP                     ;WAIT
470A  00                     4.0        NOP                     ;WAIT

470B  3E  21                 7.0        MVI     A,R.PL01!R.STPC ;STOP THE READ MACHINE
470D  D3  09                10.0        OUT     RPCTL

470F  DB  17                10.0        IN      RDATA           ;READ THE FIFO DATA
4711  32  45  48            13.0        STA     DATAA           ;STORE ACTUAL DATA
4714  DB  15                10.0        IN      RPSTA           ;READ THE FIFO DATA PARITY
4716  E6  40                 7.0        ANI     R.DATA          ;REMOVE UNWANTED BITS
4718  07                     4.0        RLC                     ;JUSTIFY THE ACTUAL
4719  07                     4.0        RLC                     ;DATA PARITY
471A  32  46  48            13.0        STA     DATAAP          ;STORE ACTUAL DATA PARITY
471D  C9                    10.0        RET
```

```
2777                                    .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
2778  471E                              SSUB
 (1)                                    ; ***************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
2779                                    ;*CLEAR ALL TU PORTS
2780  471E                              SD
 (1)                                    ; ***************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
2781                                    ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
2782                                    ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
2783                                    ;*AND LOOP MODES.
2784  471E                              SP
 (1)                                    ; ***************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
2785                                    ;*BGNSUB
2786                                    ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
2787                                    ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
2788                                    ;*    CLEAR PORT SELECT FOR TRANSPORT
2789                                    ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
2790                                    ;*    CLEAR PORT DIAGNOSTIC CONTROL
2791                                    ;*    CLEAR PORT AMTIE WORD
2792                                    ;*ENDSUB
2793  471E                              S
 (1)                                    ; ***************************************************************************
2794  471E  F5              12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
2795  471F  C5              12.0          PUSH    B
2796  4720  06  00           7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
2797  4722  DB  E0          10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
2798  4724  E6  80           7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
2799  4726  B0               4.0          ORA     B               ;ADD IN THE SELECTED PORT #
2800  4727  D3  E0          10.0          OUT     MBSEL           ;RESET TO THIS PORT
2801  4729  3E  80           7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
2802  472B  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
2803  472D  AF               4.0          XRA     A               ;CLEAR TU COMMAND A
2804  472E  D3  40          10.0          OUT     TCMD            ;
2805  4730  3E  81           7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
2806  4732  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
2807  4734  3E  00           7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
2808  4736  D3  40          10 0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
2809  4738  AF               4.0          XRA     A
2810  4739  D3  44          10.0          OUT     TAMT            ;CLEAR AMTIE WORD
2811  473B  D3  48          10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
2812  473D  D3  4C          10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
2813  473F  04               4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
2814  4740  78               4.0          MOV     A,B
2815  4741  FE  04           7.0          CPI     4               ;DONE?
2816  4743  C2  22  47       10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
2817  4746  C1              10.0          POP     B               ;RESET B & C
2818  4747  F1              10.0          POP     PSW             ;ALL DONE
2819  4748  C9              10.0          RET                     ;EXIT
2820
```

```
2822                                    .SBTTL  SUBROUTINE CHECK AMTIE
2823  4749                              SSUB
 (1)                                    ;*****************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
2824                                    ;*CHECK AMTIE SUBROUTINE
2825  4749                              SD
 (1)                                    ;*****************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
2826                                    ;*THIS SUBROUTINE IS USED TO ISOLATE A BIT FAILURE IN THE READ PATH DATA
2827                                    ;*OR AMTIE BYTES TO A SPECIFIC READ CHANNEL BY COMPARING THE ACTUAL AND
2828                                    ;*EXPECTED INPUTS TO THE SUBROUTINE ON A BIT BY BIT BASIS.
2829  4749                              SP
 (1)                                    ;*****************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
2830                                    ;*BGNSUB
2831                                    ;*     IF CHANNEL 0 ACTUAL INFORMATION NOT = CHANNEL 0 EXPECTED
2832                                    ;*       THEN-ERROR CHANNEL 0
2833                                    ;*       ELSE-CONTINUE
2834                                    ;*     ENDIF
2835                                    ;*     IF CHANNEL 1 ACTUAL INFORMATION NOT = CHANNEL 1 EXPECTED
2836                                    ;*       THEN-ERROR CHANNEL 1
2837                                    ;*       ELSE-CONTINUE
2838                                    ;*     ENDIF
2839                                    ;*     IF CHANNEL 2 ACTUAL INFORMATION NOT - CHANNEL 2 EXPECTED
2840                                    ;*       THEN-ERROR CHANNEL 2
2841                                    ;*       ELSE-CONTINUE
2842                                    ;*     ENDIF
2843                                    ;*     IF CHANNEL 3 ACTUAL INFORMATION NOT = CHANNEL 3 EXPECTED
2844                                    ;*       THEN-ERROR CHANNEL 3
2845                                    ;*       ELSE-CONTINUE
2846                                    ;*     ENDIF
2847                                    ;*     IF CHANNEL 4 ACTUAL INFORMATION NOT = CHANNEL 4 EXPECTED
2848                                    ;*       THEN-ERROR CHANNEL 4
2849                                    ;*       ELSE-CONTINUE
2850                                    ;*     ENDIF
2851                                    ;*     IF CHANNEL 5 ACTUAL INFORMATION NOT = CHANNEL 5 EXPECTED
2852                                    ;*       THEN-ERROR CHANNEL 5
2853                                    ;*       ELSE-CONTINUE
2854                                    ;*     ENDIF
2855                                    ;*     IF CHANNEL 6 ACTUAL INFORMATION NOT - CHANNEL 6 EXPECTED
2856                                    ;*       THEN-ERROR CHANNEL 6
2857                                    ;*       ELSE-CONTINUE
2858                                    ;*     ENDIF
2859                                    ;*     IF CHANNEL 7 ACTUAL INFORMATION NOT = CHANNEL 7 EXPECTED
2860                                    ;*       THEN-ERROR CHANNEL 7
2861                                    ;*       ELSE-CONTINUE
2862                                    ;*     ENDIF
2863                                    ;*     IF CHANNEL P ACTUAL INFORMATION NOT   CHANNEL P EXPECTED
2864                                    ;*       THEN-ERROR CHANNEL P
2865                                    ;*       ELSE-CONTINUE
2866                                    ;*     ENDIF
```

```
2867                                        ;*ENDSUB
2868  4749                                  $
 (1)                                        ; ************************************************************************
2869  4749  21  43  48      10.0  CKAMT:  LXI     H,DATAE          ;LOAD EXPECTED DATA ADDRESS
2870  474C  3A  45  48      13.0          LDA     DATAA            ;GET THE ACTUAL DATA
2871  474F                        ROUT    ADATA            ;WRITE THE ACTUAL DATA TO THE CAS
 (1)  474F  D3  94           10.0                OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  4751  7F                4.0                MOV     A,A              ;RETRY LINK
2872  4752  E6  01            7.0          ANI     $01              ;REMOVE UNWANTED BITS
2873  4754  47                4.0          MOV     B,A              ;SAVE IN B
2874  4755  7E                7.0          MOV     A,M              ;GET EXPECTED DATA
2875  4756                        ROUT    EDATA            ;WRITE THE EXPECTED DATA TO THE CAS
 (1)  4756  D3  95           10.0                OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  4758  7F                4.0                MOV     A,A              ;RETRY LINK
2876  4759  E6  01            7.0          ANI     $01              ;REMOVE UNWANTED BITS
2877  475B  B8                4.0          CMP     B                ;COMPARE
2878  475C  CA  64  47      10.0          JZ      CKAM0            ;CONTINUE IF EQUAL
2879  475F                        ERRB    OUT,CKAM0
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  475F  CD  12  28      18.0                CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)        0007                             MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4762  07                               .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  4763  00                               .BYTE                    ;PRINT ROUTINE NUMBER
 (1)  4764  CD  15  28      18.0        CKAM0:: CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4767  DA  2D  48      10.0                JC      OUT              ;LOOP ADDRESS IF LOOP SPECIFIED
2880                                    ;>CHANNEL 0 FAILED
2881
2882  476A  3A  45  48      13.0          LDA     DATAA            ;GET THE ACTUAL DATA
2883  476D  E6  02            7.0          ANI     $02              ;REMOVE UNWANTED BITS
2884  476F  47                4.0          MOV     B,A              ;SAVE IN B
2885  4770  7E                7.0          MOV     A,M              ;GET EXPECTED DATA
2886  4771  E6  02            7.0          ANI     $02              ;REMOVE UNWANTED BITS
2887  4773  B8                4.0          CMP     B                ;COMPARE
2888  4774  CA  7C  47      10.0          JZ      CKAM1            ;CONTINUF IF EQUAL
2889  4777                        ERRB    OUT,CKAM1
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4777  CD  12  28      18.0                CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)        0008                             MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  477A  08                               .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  477B  00                               .BYTE                    ;PRINT ROUTINE NUMBER
 (1)  477C  CD  15  28      18.0        CKAM1:: CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  477F  DA  2D  48      10.0                JC      OUT              ;LOOP ADDRESS IF LOOP SPECIFIED
2890                                    ;>CHANNEL 1 FAILED
```

```
2892
2893
2894  4782  3A 45 48   13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2895  4785  E6 04         7.0          ANI     $04             ;REMOVE UNWANTED BITS
2896  4787  47            4.0          MOV     B,A             ;SAVE IN B
2897  4788  7E            7.0          MOV     A,M             ;GET EXPECTED DATA
2898  4789  E6 04         7.0          ANI     $04             ;REMOVE UNWANTED BITS
2899  478B  B8            4.0          CMP     B               ;COMPARE
2900  478C  CA 94 47      10.0          JZ      CKAM2           ;CONTINUE IF EQUAL
2901  478F                              ERRB    OUT,CKAM2
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  478F  CD 12 28      18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0009                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4792  09                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4793  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4794  CD 15 28      18.0            CKAM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4797  DA 2D 48      10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2902                            ;>CHANNEL 2 FAILED
2903
2904  479A  3A 45 48   13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2905  479D  E6 08         7.0          ANI     $08             ;REMOVE UNWANTED BITS
2906  479F  47            4.0          MOV     B,A             ;SAVE IN B
2907  47A0  7E            7.0          MOV     A,M             ;GET EXPECTED DATA
2908  47A1  E6 08         7.0          ANI     $08             ;REMOVE UNWANTED BITS
2909  47A3  B8            4.0          CMP     B               ;COMPARE
2910  47A4  CA AC 47      10.0          JZ      CKAM3           ;CONTINUE IF EQUAL
2911  47A7                              ERRB    OUT,CKAM3
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  47A7  CD 12 28      18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000A                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47AA  0A                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47AB  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  47AC  CD 15 28      18.0            CKAM3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47AF  DA 2D 48      10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2912                            ;>CHANNEL 3 FAILED
2913
2914
2915  47B2  3A 45 48   13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2916  47B5  E6 10         7.0          ANI     $10             ;REMOVE UNWANTED BITS
2917  47B7  47            4.0          MOV     B,A             ;SAVE IN B
2918  47B8  7E            7.0          MOV     A,M             ;GET EXPECTED DATA
2919  47B9  E6 10         7.0          ANI     $10             ;REMOVE UNWANTED BITS
2920  47BB  B8            4.0          CMP     B               ;COMPARE
2921  47BC  CA C4 47      10.0          JZ      CKAM4           ;CONTINUE IF EQUAL
2922  47BF                              ERRB    OUT,CKAM4
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  47BF  CD 12 28      18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000B                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47C2  0B                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47C3  00                                    .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  47C4  CD 15 28      18.0            CKAM4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47C7  DA 2D 48      10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2923                            ;>CHANNEL 4 FAILED
```

```
2925
2926  47CA  3A  45  48   13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2927  47CD  E6  20        7.0          ANI     $20             ;REMOVE UNWANTED BITS
2928  47CF  47           4.0          MOV     B,A             ;SAVE IN B
2929  47D0  7E           7.0          MOV     A,M             ;GET EXPECTED DATA
2930  47D1  E6  20        7.0          ANI     $20             ;REMOVE UNWANTED BITS
2931  47D3  B8           4.0          CMP     B               ;COMPARE
2932  47D4  CA  DC  47   10.0          JZ      CKAM5           ;CONTINUE IF EQUAL
2933  47D7                             ERRB    OUT,CKAM5
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  47D7  CD  12  28   18.0                         CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000C                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47DA  0C                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47DB  00                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  47DC  CD  15  28   18.0                CKAM5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47DF  DA  2D  48   10.0                        JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2934                          ;>CHANNEL 5 FAILED
2935
2936
2937  47E2  3A  45  48   13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2938  47E5  E6  40        7.0          ANI     $40             ;REMOVE UNWANTED BITS
2939  47E7  47           4.0          MOV     B,A             ;SAVE IN B
2940  47E8  7E           7.0          MOV     A,M             ;GET EXPECTED DATA
2941  47E9  E6  40        7.0          ANI     $40             ;REMOVE UNWANTED BITS
2942  47EB  B8           4.0          CMP     B               ;COMPARE
2943  47EC  CA  F4  47   10.0          JZ      CKAM6           ;CONTINUE IF EQUAL
2944  47EF                             ERRB    OUT,CKAM6
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  47EF  CD  12  28   18.0                         CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000D                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47F2  0D                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47F3  00                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  47F4  CD  15  28   18.0                CKAM6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47F7  DA  2D  48   10.0                        JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2945                          ;>CHANNEL 6 FAILED
2946
2947  47FA  3A  45  48   13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2948  47FD  E6  80        7.0          ANI     $80             ;REMOVE UNWANTED BITS
2949  47FF  47           4.0          MOV     B,A             ;SAVE IN B
2950  4800  7E           7.0          MOV     A,M             ;GET EXPECTED DATA
2951  4801  E6  80        7.0          ANI     $80             ;REMOVE UNWANTED BITS
2952  4803  B8           4.0          CMP     B               ;COMPARE
2953  4804  CA  0C  48   10.0          JZ      CKAM7           ;CONTINUE IF EQUAL
2954  4807                             ERRB    OUT,CKAM7
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4807  CD  12  28   18.0                         CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000E                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  480A  0E                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  480B  00                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  480C  CD  15  28   18.0                CKAM7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  480F  DA  2D  48   10.0                        JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2955                          ;>CHANNEL 7 FAILED
```

```
2957
2958
2959  4812  21  44  48  10.0          LXI     H,DATAEP        ;GET THE EXPECTED DA~A PARITY ADDRESS
2960  4815  3A  46  48  13.0          LDA     DATAAP          ;GET THE ACTUAL DATA
2961  4818  E6  01       7.0          ANI     $01             ;REMOVE UNWANTED BITS
2962  481A  47           4.0          MOV     B,A             ;SAVE IN B
2963  481B  7E           7.0          MOV     A,M             ;GET EXPECTED DATA
2964  481C  E6  01       7.0          ANI     $01             ;REMOVE UNWANTED BITS
2965  481E  B8           4.0          CMP     B               ;COMPARE
2966  481F  CA  27  48  10.0          JZ      CKAMP           ;CONTINUE IF EQUAL
2967  4822                           ERRB    OUT,CKAMP,1
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4822  CD  12  28  18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000F                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4825  0F                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4826  01                                  .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4827  CD  15  28  18.0            CKAMP:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  482A  DA  2D  48  10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2968                           ;>CHANNEL P FAILED
2969
2970  482D  C9           10.0  OUT:    RET
```

```
2972                                    .SBTTL  TABLE EXPECTED FIFO DATA
2973
2974  482E  00          FIFOEX: .BYTE   $00          ;FIFO READ 01 DATA
2975  482F  00                  .BYTE   $00          ;FIFO READ 01 PARITY
2976  4830  FF                  .BYTE   $FF          ;FIFO READ 02 DATA
2977  4831  01                  .BYTE   $01          ;FIFO READ 02 PARITY
2978  4832  00                  .BYTE   $00          ;FIFO READ 03 DATA
2979  4833  00                  .BYTE   $00          ;FIFO READ 03 PARITY
2980  4834  FF                  .BYTE   $FF          ;FIFO READ 04 DATA
2981  4835  01                  .BYTE   $01          ;FIFO READ 04 PARITY
2982  4836  00                  .BYTE   $00          ;FIFO READ 05 DATA
2983  4837  00                  .BYTE   $00          ;FIFO READ 05 PARITY
2984  4838  FF                  .BYTE   $FF          ;FIFO READ 06 DATA
2985  4839  01                  .BYTE   $01          ;FIFO READ 06 PARITY
2986  483A  00                  .BYTE   $00          ;FIFO READ 07 DATA
2987  483B  00                  .BYTE   $00          ;FIFO READ 07 PARITY
2988  483C  FF                  .BYTE   $FF          ;FIFO READ 08 DATA
2989  483D  01                  .BYTE   $01          ;FIFO READ 08 PARITY
2990  483E  00                  .BYTE   $00          ;FIFO READ 09 DATA
2991  483F  00                  .BYTE   $00          ;FIFO READ 09 PARITY
2992  4840  FF                  .BYTE   $FF          ;FIFO READ 10 DATA
2993  4841  01                  .BYTE   $01          ;FIFO READ 10 PARITY
2994  4842  00          FIFOEN: .BYTE   0            ;END OF TABLE
```

```
2996                                    .SBTTL  PROGRAM VARIABLES
2997
2998  4843   00                         DATAE:  .BYTE   0               ;EXPECTED DATA BYTE
2999  4844   00                         DATAEP: .BYTE   0               ;EXPECTED DATA BYTE PARITY
3000  4845   00                         DATAA:  .BYTE   0               ;ACTUAL DATA BYTE
3001  4846   00                         DATAAP: .BYTE   0               ;ACTUAL DATA BYTE PARITY
3002  4847   00                         UNITMP: .BYTE   0               ;UNITS MAP
3003  4848   00                         PASCT:  .BYTE   0               ;PASS COUNTER
3004  4849   00                         PASFL:  .BYTE   0               ;FAILING PASS NUMBER
3005  484A   00                         TICNM:  .BYTE   0               ;TACK RATE COUNTER
3006
3007        0000                        .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE   4F9B          ATTCD   4F97          AXNUM   4F91
B      =%0000          BADST  = 0090          BIT0   = 0001          BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008          BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080          BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A          BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00           BSAVE   4F9C          BYTCNT= 00D4          BYTEH   4F24
BYTEL   4F23           C      =%0001          CASCT   4F21          CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1
CBYTH  = 008B          CBYTL  = 008A          CDG1H  = 0087          CDG1L  = 0086
CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095          CDG3L  = 0094
CDVTH  = 008D          CDVTL  = 008C          CHPTIE= 0028          CHOTIE= 0020
CH1TIE= 0021           CH2TIE= 0022          CH3TIE= 0023          CH4TIE= 0024
CH5TIE= 0025           CH6TIE= 0026          CH7TIE= 0027          CKAMP   4827  G
CKAMT   4749           CKAM0   4764  G       CKAM1   477C  G       CKAM2   4794  G
CKAM3   47AC  G        CKAM4   47C4  G       CKAM5   47DC  G       CKAM6   47F4  G
CKAM7   480C  G        CKLOP  = 2815          CLEAR   471E          CLKCTL= 00F0
CLOCK   4F26           CLRLP   4722          CMC0H  = C099          CMC0L  = 0098
CMC1H  = 009B          CMC1L  = 009A          CMC2H  = 009D          CMC2L  = 009C
CMC3H  = 009F          CMC3L  = 009E          CMINH  = 0097          CMINL  = 0096
CNTCTL= 00D7           CRCWRD= 0018          CSAVE   4F9D          CSRLH  = 0091
CSRLL  = 0090          CTCH   = 0085          CTCL   = 0084          CTSTH  = 008F
CTSTL  = 008E          CXCTH  = 0081          CXCTL  = 0080          CXINH  = 0083
CXINL  = 0082          C.     = 0001          C.AVAI= 0080          C.DP   = 0008
C.DSE  = 0010          C.DTU  = 0003          C.DVA  = 0008          C.FAIL= 00FC
C.FMT  = 0070          C.FNCT= 003E          C.GO   = 0001          C.INTC= 00FE
C.MAIN= 0020           C.NSA  = 0080          C.RCT  = 00FC          C.SER  = 0080
C.SHR  = 0040          C.SKPC= 000F          C.TAPE= 0040          C.WCS  = 0002
D      =%0002          DATAA   4845          DATAAP  4846          DATACT= 00D0
DATAE   4843           DATAEP  4844          DBUS    4F28          DBUSCT= 00C0
DBUSST= 00C0           DDRA   = 00D8          DDRAIN= 0010          DDRB   = 00D9
DDRBIN= 0002           DDRC   = 00DA          DDRCIN= 0001          DDRCO  = 0088
DDRCTL= 00DB           DIAFLG  4F22          DIAGPG= 4300          DIAGRM= 4F90
DIARD  = 000B          DONE1  = 0045          DONINT= 0010          DSAVE   4F9E
DSE    = 0006          DUMMY   4490  G       D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010           D.LAGC= 0020          D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008           D.WR4  = 0080          E      =%0003          ECCBAD= 0042
ECCCOR= 0019           ECCOK  = 0041          ECCSTA= 001A          ECCTST= 000E
EDATA  = 0095          EOTCLR= 0003          ERFLG   4F93          ERLP   = 2809
ERLPA  = 280F          ERLPB  = 2812          ERLPE  = 280C          ERNUM   4F90
ERRCNT= 00D6           ESAVE   4F9F          EXIT    462C          E.ACRC= 0010
E.AMT  = 0020          E.CDP  = 0080          E.CRC  = 0080          E.PNTR= 0008
E.RPE  = 0040          E.STEC= 0001          E.TTEC= 0002          E.UNC  = 0004
FIFOEN  4842           FIFOEX  482E          FIFOL1  4309          FIFOL2  436B
FIFORD= 006A           FORMAT  4F25          FWDTST= 0061          GCRID  = 0089
GCRSET= 0002           GOODTM= 0092          H      =%0004          HLSAVE  4FA0
IE     = 0008          INTSTA= 00E0          ITERA   4F9A          I.PWR  = 0020
I.RMPE= 0040           I5.5   = 0010          I6.5   = 0020          I7.5   = 0040
KCALL  = 005F          KCLR   = 007B          KDEP   = 003F          KENAB  = 0078
KEXAM  = 003E          KEYBRD= 00C8          KEY1   = 0078          KEY10  = 006D
KEY11  = 006E          KEY12  = 006F          KEY13  = 005C          KEY14  = 005D
KEY15  = 005E          KEY16  = 005F          KEY17  = 003C          KEY18  = 003D
KEY19  = 003E          KEY2   = 0079          KEY20  = 003F          KEY3   = 007A
KEY4   = 007B          KEY5   = 0074          KEY6   = 0075          KEY7   = 0076
```

```
KEY8   = 0077          KEY9   = 006C          KINTA = 006F          KLDAD = 003D
KN0    = 003C          KN1    = 005C          KN2   = 005D          KN3   = 005E
KN4    = 006C          KN5    = 006D          KN6   = 006E          KN7   = 0074
KN8    = 0075          KN9    = 0076          KU2   = 0079          KU3   = 007A
KU8    = 0077          L      =%0005          LBLANK= 000F          LCE   = 000B
LCH    = 000C          LCL    = 000D          LCP   = 000E          LC0   = 0000
LC1    = 0001          LC2    = 0002          LC3   = 0003          LC4   = 0004
LC5    = 0005          LC6    = 0006          LC7   = 0007          LC8   = 0008
LC9    = 0009          LDLEDA= 00CA           LDLEDB= 00CB          LDLEDC= 00CC
LDLEDD= 00CD           LDLEDE= 00CE           LDLEDF= 00CF          LKDIAG= 2800
LKKBD  = 004C          LKKEY  = 0049          LKLWMG= 0058          LKLWMP= 0055
LKLWPG= 0052           LKLWPP= 004F           LKMOD7= 0046          LKOPR = 0046
LPFLG   4F94           LPNUM   4F92           M     =%0006          MBSEL = 00E0
MB.A   = 0008          MB.B   = 0004          MEMTOP= 4FFF          MINUS = 000A
MM     = 8000          MSE    = 0008          MSGN  = 000F          MTACLR= 0000
MT.ARA= 0020           MT.CPE= 0080           MT.DSE= 0001          MT.FWD= 0040
MT.INH= 0008           MT.LWR= 0004           MT.MOT= 0002          MT.NWF= 0080
MT.PEC= 0040           MT.PSB= 0004           MT.PS0= 0001          MT.PS1= 0002
MT.REV= 0020           MT.WRT= 0010           MT.Z  = 0008          M.ATA = 0080
M.CAPE= 0020           M.CONT= 0080           M.DEM = 0020          M.EBL = 0010
M.EXC  = 0008          M.FAIL= 0008           M.ILR = 0010          M.INIT= 0010
M.OCC  = 0020          M.ONLI= 0001           M.PE  = 0040          M.PORT= 0080
M.RDEN= 0002           M.RDPE= 0008           M.RUN = 0004          M.SCLK= 0001
M.TRA  = 0040          M.UNIT= 0007           M.WCLK= 0040          M.WCLN= 0080
M.WREN= 0080           M5.5   = 0001          M6.5  = 0002          M7.5  = 0004
NOTCAP= 0088           OKAY   = 00FF          OPRRAM= 4300          OPSTRT= 0058
OPVER  = 0040          OUT     482D           PADCNT= 00D5          PADCRC= 0080
PASCT   4848           PASFL   4849           PDIAG = 0048          PEID  = 008A
PENAB  = 004C          PESET  = 0001          PL    = 00B1          PRDD  = 004C
PRENF  = 009C          PS     = 00B2          PSTAT = 0048          PSW   =%0009
P.AMTP= 0001           P.BCTC= 0040           P.CMDP= 0020          P.INTE= 0080
P.LCS  = 0040          P.LWR  = 0020          P.RDP  = 0002         P.RPEN= 0004
P.RPST= 0002           P.RPOE= 0020           P.RP1E= 0010          P.RP2E= 0020
P.RP3E= 0010           P.SING= 0080           P.STAT= 0002          P.STPE= 0080
P.TACH= 0008           P.TUPR= 0010           P.WCSP= 0004          P.WDS  = 0040
P.WFLP= 0001           P.WPEN= 0010           P.WPOE= 0008          P.WP1E= 0004
P.WP2E= 0008           P.WP3E= 0004           P.5VOK= 0002          QUE    = 281B
QUEM   = 281E          RAMT   = 0010          RARA  = 0006          RARAI = 0004
RCHBDO= 0048           RCHBD1= 0047           RCHOK = 0046          RCHTST= 000C
RCLRT  = 000D          RCMD   = 000B          RCMLP = 0003          RCONT = 0080
RDATA  = 0017          RDCLK  = 0010          RDON  = 0011          RDPL02  46FA
RDSET   469B           READG  = 0007          REND  = 0014          REQST = 2806
RESCHR= 00D1           REVTST= 0064           REWIND= 0004          RFIFOL= 0008
RGCLK  = 0002          RGCRI  = 0003          RIBG  = 0001          RILL  = 0012
RINST  = 000C          RMCTST= 0008           RMK2  = 0013          RNOP  = 0000
RPATH  = 0001          RPBAD  = 0044          RPCHI = 0001          RPCLK = 0003
RPCTL  = 0009          RPEI   = 0002          RPFAIL= 0000          RPF1  = 009D
RPF2   = 009E          RPOK   = 0043          RPOSTN= 0016          RPSTA = 0015
RRCMT  = 000A          RSTAT  = 0002          RTIEB = 000A          RTIER = 0030
RTM    = 0005          RUNKI  = 0009          RUPTST= 005E          RWDUNL= 0005
R.AMT  = 0001          R.BOP  = 0008          R.DATA= 0040          R.DON  = 0002
R.DRDY= 0010           R.END  = 0010          R.ILL  = 0004         R.JVOK= 0004
R.MK2  = 0008          R.PLOD= 0008           R.PLOO= 0010          R.PL01= 0020
R.POST= 0020           R.STNM= 0002           R.STOP= 0004          R.STPC= 0001
```

```
R.TBJN= 0080            R.TSTD= 0040            R.VOK = 0080            R00H  = 0081
R00L  = 0080            R01H  = 0083            R01L  = 0082            R02H  = 0085
R02L  = 0084            R03H  = 0087            R03L  = 0086            R04H  = 0089
R04L  = 0088            R05H  = 008B            R05L  = 008A            R06H  = 008D
R06L  = 008C            R07H  = 008F            R07L  = 008E            R10H  = 0091
R10L  = 0090            R11H  = 0093            R11L  = 0092            R12H  = 0095
R12L  = 0094            R13H  = 0097            R13L  = 0096            R14H  = 0099
R14L  = 0098            R15H  = 009B            R15L  = 009A            R16H  = 009D
R16L  = 009C            R17H  = 009F            R17L  = 009E            R7.5  = 0010
SELCLR= 0000            SETATA= 00A1            SID   = 0080            SOD   = 0080
SOE   = 0040            SP    =%0008            SSCLK = 0040            SSTEP = 0005
STACK = 4FFF            STATRM= 4F20            STPCT   4F20            STRSP = 5000
TADR00= 0080            TADR01= 0081            TADR02= 0082            TADR03= 0083
TADR04= 0084            TADR05= 0085            TADR06= 0086            TADR07= 0087
TADR10= 0088            TADR11= 0089            TADR12= 008A            TADR13= 008B
TAMT  = 0044            TASEL = 0080            TCMD  = 0040            TC.FWD= 0040
TC.INH= 0008            TC.LWR= 0004            TC.REV= 0020            TC.WRT= 0010
TEMP    4F99            TEST1   4300            TEST10  4538            TEST11  4588
TEST12  45B1            TEST13  45DA            TEST14  4603            TEST2   4362
TEST3   43C2            TEST4   4472            TEST5   44BD            TEST6   44E6
TEST7   450F            TICNM   484A            TMF   = 0099            TMRDY = 0040
TRKENA= 00D2            TSET  = 2803            TSTEND= 2818            TSTS  = 0040
TSTVL0  4658            TSTVL1  466E            TSTVL2  4684            TST01X  4305
TST02X  4367            TST03X  43C7            TST04X  4499            TST05X  44C2
TST06X  44EB            TST07X  4514            TST10X  453D            TST11X  458D
TST12X  45B6            TST13X  45DF            TST14X  4608            TUSEL0= 00D1
TUSEL1= 00D2            TU78  = 0010            T.ATH0= 0001            T.ATH1= 0002
T.BOT = 0004            T.EOT = 0002            T.FPT = 0001            T.NTHR= 0004
T.ONL = 0020            T.PES = 0008            T.PSBJ= 0020            T.PS0J= 0008
T.PS1J= 0010            T.RDY = 0080            T.RDY0= 0040            T.RWD = 0010
T.SCLK= 0002            UIBG  = 00A1            UNITMP  4847            VALFC   4F98
VALTB   4F95            VELCN0  4668  G         VELCN1  467E  G         VELCN2  4694  G
VELCIL  4650            VELOK   4632            VELTST= 005B            VOKCN0  4552  G
VOKCN1  456E  G         VOKLP0  4541            VOKLP1  4558            WDR.P = 0010
WMCCTL= 00D3            WMCERR= 00DA            WMCSTA= 00D0            WRTCLK= 0000
WRTDAT= 00D3            W.ACRC= 0004            W.CRC = 0008            W.DIAG= 0002
W.DONN= 0040            W.ECC = 0010            W.ENAB= 0080            W.ERR = 0020
W.FMT = 0070            W.GCR = 0010            W.LEFT= 0004            W.ONES= 0020
W.RESI= 0002            W.REV = 0004            W.ROME= 0010            W.RST = 0001
W.SKIP= 000F            W.WRIT= 0008            W.XFER= 0020            X     =%000A
X.DONN= 0080            X.ENAB= 0040            X.PEPE= 0002            X.ROME= 0001
X.WCLK= 0001            Y     =%000B            .     = 484B


ERRORS DETECTED:  0

*RPM2.A78/PTP,RPM2=NLIST,PARAM,MACRO,LIST,RPM2
RUN-TIME: 5 7 0 SECONDS
CORE USED:  10K
```

     3   "MEMORY MAP" EXTENDED ADDRESS BIT
    19   READ MICRO CONTROLLER CONTROL & STATUS WORDS
   139   TRANSPORT/PORT COMMAND + STATUS WORDS
   380   COMMON ADDRESS SPACE REGISTERS
   444   MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
   485   MASSBUS CONTROL WORD DEFINITIONS
   557   WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
   709   KEYBOARD/DISPLAY CONTROL + STATUS WORD
   797   MISCELLANEOUS VARIABLE DEFINITIONS
   801   RAM ADDRESS DEFINITIONS
   819   LINKAGE TABLE ADDRESSES
   842   BIT DEFINITIONS
   856   RIM AND SIM INSTRUCTION BIT DEFINITIONS
   877   INTERRUPTABLE INSTRUCTION MACROS
   924   DIAGNOSTIC VARIABLES
   945   COMMON VARIABLE STORAGE
   959   MACRO CALL DEFINITIONS
  1047   TM78 MICRO-DIAGNOSTIC MACRO CALLS
  1101   8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
  1333   TEST 01 - IBG DETECT TEST
  1505   TEST 02 - PE ID BURST TEST
  1642   TEST 03 - GCR ID BURST TEST
  1783   TEST 04 - GCR ARA BURST TEST
  1915   TEST 05 - GCR ARA ID CHARACTER TEST
  2060   TEST 06 - TAPE MARK TEST
  2195   TEST 07 - 5 X 4 TRANSLATION TEST (FORWARD)
  2482   SUBROUTINE CLEAR ALL TU PORTS
  2526   SUBROUTINE CHECK 5X4
  2669   TABLE UNTRANSLATED INPUT DATA
  2851   TABLE TRANSLATED OUTPUT DATA
  3001   PROGRAM VARIABLES

```
                ;*************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;                "BYTE/SCLK COUNT NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                MM = DATA FORMAT FROM CAS REGISTER 2
                ;                NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;                BYTE-SCLK COUNT = LLL
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                LLL = AS ABOVE
                ;                MM  = AS ABOVE
                ;                NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;                TRANSITION COUNT = LLL
                ;                WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;                EXPECTED 18 BITS =      E EEEEEE
                ;                ACTUAL 18 BITS =        A AAAAAA
                ;
                ;                WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;                BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;                OF CAS REG 15 LOW BYTE.
                ;
                ;                ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;                AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;                17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;                TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;                AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;                "SUBGROUP NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;*************************************************************************
```

```
1016                    ; ************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ; ************************************************************************
1035                    ; ************************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ; ************************************************************************
1329    4300                    .       DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  RPM3 - READ PATH MICRO CONTROLLER PART #3
                                .SBTTL  TEST 01 - IBG DETECT TEST
                                ;ID     RPM3-READ PATH MICRO CONTROLLER PART #3

1336  4300                      ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1337                            ;*RPM - IBG DETECT TEST
1338  4300                      SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1339                            ;*THIS TEST IS DESIGNED TO CHECK THE READ PATHS ABILITY TO DETECT AN
1340                            ;*INTERBLOCK GAP AND AN INTERBLOCK GAP FAILURE.
1341  4300                      SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1342                            ;*BGNTST
1343                            ;*  SET NORMAL READ PATH CLOCK
1344                            ;*  CALL SUBROUTINE CLEAR
1345                            ;*  ENABLE THE READ MICRO CONTROLLER
1346                            ;*  INIT THE AMTIE MASK TO ZERO (9 BITS WIDE)
1347                            ;*  BGNDO
1348                            ;*      :   SET ALL 9 AMTIES FROM THE USER SPECIFIED PORT
1349                            ;*      :   DELAY 50 MICROSECONDS
1350                            ;*      :   ISSUE THE IBG COMMAND
1351                            ;*      :   DELAY 50 MICROSECONDS
1352                            ;*      :   READ THE READ MICRO STATUS
1353                            ;*      :       IF STATUS = 0
1354                            ;*      :       :   THEN-CONTINUE
1355                            ;*      :       :   ELSE-ERROR
1356                            ;*      :       ENDIF
1357                            ;*      :   ISSUE THE AMTIE LINE AS INDICATED BY THE AMTIE MASK
1358                            ;*      :   DELAY 50 MICROSECONDS
1359                            ;*      :   READ THE READ MICRO STATUS
1360                            ;*      :       IF STATUS = "BADST"
1361                            ;*      :       :   THEN-CONTINUE
1362                            ;*      :       :   ELSE-ERROR
1363                            ;*      :       ENDIF
1364                            ;*      :   INCREMENT THE AMTIE MASK BY 1
1365                            ;*      :   DO UNTIL AMTIE MASK = 777(8)
1366                            ;*  ENDDO
1367                            ;*ENDTST
1368  4300                      SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1369                            ;*RPM3 MICRO TEST 01
1370                            ;*RPM3 MICRO ERROR 01
1371                            ;*RPM3-READ PATH-IBG DETECT
1372                            ;*M8955, M8950'S
1373                            ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
```

```
                                        ;*FATAL ERROR - TEST ABORTED
                                        ;*
                                        ;*RPM3 MICRO TEST 01
                                        ;*RPM3 MICRO ERROR 02
                                        ;*RPM3-READ PATH-IBG DETECT
                                        ;*M8953, M8950'S
                                        ;*READ PATH STATUS INCORRECT WHEN MONITORING A GOOD IBG
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM3 MICRO TEST 01
                                        ;*RPM3 MICRO ERROR 03
                                        ;*RPM3-READ PATH-IBG DETECT
                                        ;*M8953, M8950'S
                                        ;*READ PATH STATUS INCORRECT WHEN MONITORING A BAD IBG
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*EXPECTED 18 BITS =    E EEEEEE
                                        ;*ACTUAL 18 BITS =      A AAAAAA
                                        ;*EXTENDED ACTUAL PRINTOUT = AMTIE LINES
1394  4300                              S
 (1)                                    ; ***************************************************************************
1395
1396  4300                              TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1)  4300  3E  01            7.0                MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                CALL    TSET            ;SETUP THE TEST
1397                                    ;%RPM3-READ PATH-IBG DETECT
1398                                    ;&M8953, M8950'S
1399  4305                                      REQ     @7,0,0,0,0
 (1)  4305  CD  06  28       18.0                         CALL    REQST
 (1)  4308  00                                            .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4309  00  00                                        .WORD   0               ;SYSTEM "0" COUNT
 (1)  430B  00  00                                        .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                            .BYTE   0               ;DATA COMPARE FLAG IF  1
 (1)  430E  07                                            .BYTE   @7              ;REQUEST CODE
1400  430F                                      RIN     R12L
 (1)  430F  DB  94            10.0                       IN      R12L            ;READ R12L INTO AC
 (1)  4311  7F                 4.0                       MOV     A,A             ;RETRY LINK
1401  4312  32  FD  4A         13.0              STA     UNITMP
1402  4315  A7                  4.0              ANA     A
1403  4316  C2  27  43         10.0              JNZ     UNITSL
1404  4319                                      ERR     EXIT,DUMMY
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28       18.0                         CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0001                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                           .BYTE
 (1)  431E  CD  15  28       18.0                DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  CA  48       10.0                        JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1405                                    ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1406                                    ;>FATAL ERROR - TEST ABORTED
1407  4324  C3  CA  48         10.0              JMP     EXIT
1408
1409  4327  06  00               7.0  UNITSL: MVI     B,0             ;
```

```
1410  4329  3A  FD  4A      13.0          LDA     UNITMP          ;DID THE USER SPECIFY T.
1411  432C  E6  01            7.0          ANI     @01             ;PORT 0?
1412  432E  C2  44  43       10.0          JNZ     FOUND           ;YES-GO USE PORT #0
1413  4331  04                4.0          INR     B               ;NO-UPDATE POINTER TO PORT #1
1414  4332  3A  FD  4A      13.0          LDA     UNITMP          ;DID THE USER SPECIFY TU
1415  4335  E6  02            7.0          ANI     @02             ;PORT 1?
1416  4337  C2  44  43       10.0          JNZ     FOUND           ;YES-GO USE PORT #1
1417  433A  04                4.0          INR     B               ;NO-UPDATE POINTER TO PORT #2
1418  433B  3A  FD  4A      13.0          LDA     UNITMP          ;DID THE USER SPECIFY TU
1419  433E  E6  04            7.0          ANI     @04             ;PORT 2?
1420  4340  C2  44  43       10.0          JNZ     FOUND           ;YES-GO USE PORT #2
1421  4343  04                4.0          INR     B               ;NO-ASSUME PORT #3
1422  4344  CD  CD  48      18.0  FOUND:  CALL    CLEAR           ;CLEAL ALL TU PORTS
1423  4347  DB  E0           10.0          IN      INTSTA          ;GET THE MASS BUS PORT
1424  4349  E6  80            7.0          ANI     BIT7            ;
1425  434B  B0                4.0          ORA     B               ;OR IN THE TU PORT
1426  434C  D3  E0           10.0          OUT     MBSEL           ;SELECT DESIRED PORT
1427  434E  32  FE  4A      13.0          STA     SUNIT           ;SAVE FOR LATER
1428  4351  3E  10            7.0  TST01X: MVI     A,RDCLK         ;LOAD THE NORMAL READ PATH CLOCK
1429  4353  D3  F0           10.0          OUT     CLKCTL          ;SET NORMAL CLOCKS
1430
1431  4355  3E  80            7.0          MVI     A,R.TBJN                ;LOAD THE READ PATH ENABLE BIT
1432  4357  D3  09           10.0          OUT     RPCTL           ;ENABLE THE READ PATH
1433
1434  4359  21  00  00       10.0          LXI     H,@0            ;INITIALIZE THE AMTIE MASK TO ZERO
1435  435C  22  FF  4A      16.0          SHLD    AMTMSK
1436
1437  435F  3E  04            7.0  AMDLP1: MVI     A,P.RPEN
1438  4361  D3  4C           10.0          OUT     PENAB
1439  4363  3E  21            7.0          MVI     A,P.LWR!@1      ;SET DP ENABLE AND LOOP W/R AND AMTIE P
1440  4365  D3  48           10.0          OUT     PDIAG           ;SET THE PARITY AMTIE
1441  4367  3E  FF            7.0          MVI     A,@377          ;SET AMTIES 7:0
1442  4369  D3  44           10.0          OUT     TAMT
1443  436B  06  0A            7.0          MVI     B,10            ;PERFORM A 50 MICROSECOND
1444  436D  05                4.0  2$:     DCR     B               ;TIMEOUT
1445  436E  C2  6D  43       10.0          JNZ     2$
1446
1447  4371  3E  01            7.0          MVI     A,RIBG          ;LOAD THE READ INTERBLOCK GAP CMD.
1448  4373  D3  0B           10.0          OUT     RCMD            ;ISSUE THE COMMAND
1449
1450  4375  06  0A            7.0          MVI     B,10            ;PERFORM A 50 MICROSECOND
1451  4377  05                4.0  1$:     DCR     B               ;TIMEOUT
1452  4378  C2  77  43       10.0          JNZ     1$
1453
1454  437B  DB  02           10.0          IN      RSTAT           ;GET THE READ PATH STATUS
1455  437D  A7                4.0          ANA     A               ;SET CONDITION BITS
1456  437E  CA  8D  43       10.0          JZ      AMDCN1          ;CONTINUE IF ZERO
1457  4381                                 ROUT    ADATA           ;SAVE THE ACTUAL STATUS
 (1)  4381  D3  94           10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4383  7F                4.0                  MOV     A,A             ;RETRY LINK
1458  4384  AF                4.0          XRA     A               ;CLEAR THE ACCUMULATOR
1459  4385                                 ROUT    EDATA           ;STORE AS EXPECTED STATUS
 (1)  4385  D3  95           10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4387  7F                4.0                  MOV     A,A             ;RETRY LINK
```

```
1460  4388                                          ERRB    AMDLP1,AMDCN1
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4388  CD  12  28      18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0002                                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  438B  02                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  438C  00                                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  438D  CD  15  28      18.0                    AMDCN1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4390  DA  5F  43      10.0                            JC      AMDLP1          ;LOOP ADDRESS IF LOOP SPECIFIED
1461                                        ;>READ PATH STATUS INCORRECT WHEN MONITORING A GOOD IBG
1462  4393  3A  00  4B      13.0  AMDLP2:   LDA     AMTMSK+1        ;SET THE AMTIE PARITY BIT
1463  4396  F6  20             7.0          ORI     P.LWR           ;SET LOOP W/R
1464  4398  D3  48            10.0          OUT     PDIAG           ;AS PER THE MASK
1465  439A  3A  FF  4A        13.0          LDA     AMTMSK          ;SET THE AMTIE BITS 7:0
1466  439D  D3  44            10.0          OUT     TAMT            ;AS PER THE MASK
1467
1468  439F  06  0A             7.0          MVI     B,10            ;PERFORM A 50 MICROSECOND
1469  43A1  05                 4.0  2$:     DCR     B               ;TIMEOUT
1470  43A2  C2  A1  43        10.0          JNZ     2$
1471
1472  43A5  DB  02            10.0          IN      RSTAT           ;GET THE READ PATH STATUS
1473  43A7  FE  90             7.0          CPI     BADST           ;BAD STATUS?
1474  43A9  CA  CE  43        10.0          JZ      AMDCN2          ;YES-CONTINUE
1475  43AC                                  ROUT    ADATA           ;NO-STORE ACTUAL STATUS
 (1)  43AC  D3  94            10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43AE  7F                 4.0                  MOV     A,A             ;RETRY LINK
1476  43AF  3E  90             7.0          MVI     A,BADST         ;GET EXPECTED STATUS
1477  43B1                                  ROUT    EDATA           ;STORE EXPECTED STATUS
 (1)  43B1  D3  95            10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  43B3  7F                 4.0                  MOV     A,A             ;RETRY LINK
1478
1479  43B4  3A  FF  4A        13.0          LDA     AMTMSK          ;GET AMTIE BITS 7:0
1480  43B7                                  ROUT    R16L            ;STORE IN CAS
 (1)  43B7  D3  9C            10.0                  OUT     R16L            ;WRITE AC INTO R16L
 (1)  43B9  7F                 4.0                  MOV     A,A             ;RETRY LINK
1481  43BA  3A  00  4B        13.0          LDA     AMTMSK+1        ;GET AMTIE PARITY BIT
1482  43BD                                  ROUT    R16H            ;STORE IN CAS
 (1)  43BD  D3  9D            10.0                  OUT     R16H            ;WRITE AC INTO R16H
 (1)  43BF  7F                 4.0                  MOV     A,A             ;RETRY LINK
1483  43C0  3E  80             7.0          MVI     A,@200          ;SET THE ACTUAL DATA PRINT FLAG
1484  43C2                                  ROUT    R17L            ;STORE IN CAS
 (1)  43C2  D3  9E            10.0                  OUT     R17L            ;WRITE AC INTO R17L
 (1)  43C4  7F                 4.0                  MOV     A,A             ;RETRY LINK
1485  43C5  AF                 4.0          XRA     A               ;CLEAR THE EXPECTED DATA
1486  43C6                                  ROUT    R15L            ;PRINT FLAG
 (1)  43C6  D3  9A            10.0                  OUT     R15             ;WRITE AC INTO R15L
 (1)  43C8  7F                 4.0                  MOV     A,A             ;RETRY LINK
```

```
1488  43C9                                        ERRB    AMDLP2,AMDCN2,@5
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  43C9   CD   12   28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0003                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43CC   03                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43CD   05                                           .BYTE   @5              ;PRINT ROUTINE NUMBER
 (1)  43CE   CD   15   28        18.0                AMDCN2::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43D1   DA   93   43        10.0                        JC      AMDLP2          ;LOOP ADDRESS IF LOOP SPECIFIED
1489                                        ;>READ PATH STATUS INCORRECT WHEN MONITORING A BAD IBG
1490                                        ;<EXTENDED ACTUAL PRINTOUT = AMTIE LINES
1491  43D4   2A   FF   4A        16.0            LHLD    AMTMSK          ;GET THE AMTIE MASK
1492  43D7   23                    6.0            INX     H               ;INCREMENT BY 1
1493  43D8   22   FF   4A        16.0            SHLD    AMTMSK          ;SAVE THE AMTIE MASK
1494  43DB   3A   FF   4A        13.0            LDA     AMTMSK          ;GET LS 8 BITS OF AMTIE MASK
1495  43DE   FE   FF               7.0            CPI     @377            ;ALL ONES?
1496  43E0   C2   5F   43        10.0            JNZ     AMDLP1          ;NO-CONTINUE
1497  43E3   3A   00   4B        13.0            LDA     AMTMSK+1        ;YES-GET AMTIE PARITY BIT
1498  43E6   FE   01               7.0            CPI     @1              ;END OF TEST
1499  43E8   C2   5F   43        10.0            JNZ     AMDLP1          ;NO-CONTINUE
1500  43EB                                        ENDTST  TST01X          ;YES-END IT
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43EB                                  REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43EB   CD   06   28        18.0                        CALL    REQST
 (2)  43EE   00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  43EF   00   00                                      .WORD                   ;SYSTEM "" COUNT
 (2)  43F1   00   00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43F3   00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  43F4   07                                           .BYTE   7               ;REQUEST CODE
 (1)  43F5   3A   9A   4F        13.0            LDA     ITERA           ;GET ITERATION COUNT
 (1)  43F8   3D                   4.0            DCR     A               ;DOWNCOUNT
 (1)  43F9   32   9A   4F        13.0            STA     ITERA           ;SAVE COUNT
 (1)  43FC   F2   51   43        10.0            JP      TST01X          ;DO TEST UNTIL TILL = 0
1501  43FF   CD   CD    8        18.0            CALL    CLEAR
1502
1503
```

```
                                .SBTTL  TEST 02 - PE ID BURST TEST

                                ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*PE ID BURST TEST
                                SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST IS DESIGNED TO CHECK THE READ PATHS ABILITY TO DETECT A
                                ;*CORRECT PE IDENTIFICATION BURST AND TO REJECT INCORRECT ONES.
                                SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    ENABLE THE READ MICRO CONTROLLER
                                ;*    INIT THE AMTIE MASK TO ZERO (9 BITS WIDE)
                                ;*    BGNDO
                                ;*      :   SET UP THE AMTIES FOR A GOOD ID BURST
                                ;*      :   DELAY 50 MICROSECONDS
                                ;*      :   ISSUE THE ID BURST COMMAND
                                ;*      :   DELAY 50 MICROSECONDS
                                ;*      :   READ THE READ MICRO STATUS
                                ;*      :       IF THE STATUS = 0
                                ;*      :       :   THEN-CONTINUE
                                ;*      :       :   ELSE-ERROR
                                ;*      :       ENDIF
                                ;*      :   ISSUE THE AMTIE LINES AS PER THE AMTIE MASK
                                ;*      :   DELAY 50 MICROSECONDS
                                ;*      :   READ THE READ MICRO STATUS
                                ;*      :       IF STATUS = "BADST"
                                ;*      :       :   THEN-CONTINUE
                                ;*      :       :   ELSE-ERROR
                                ;*      :       ENDIF
                                ;*      :   INCREMENT THE AMTIE MASK BY ONE
                                ;*      :       IF AMTIE MASK = 377(8)
                                ;*      :       :   THEN-INCREMENT THE AMTIE MASK BY ONE
                                ;*      :       :   ELSE-CONTINUE
                                ;*      :       ENDIF
                                ;*      :   DO UNTIL AMTIE MASK = 1000(8)
                                ;*    ENDDO
                                ;*ENDTST
                                SE
                                ;*********************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM3 MICRO TEST 02
                                ;*RPM3 MICRO ERROR 04
                                ;*RPM3-READ PATH-PE ID BURST DETECT
```

```
1547                                    ;*M8953, M8950'S
1548                                    ;*READ PATH STATUS INCORRECT WHEN MONITORING A GOOD PE ID
1549                                    ;*ACTUAL = NNNN
1550                                    ;*EXPECTED = NNNN
1551                                    ;*
1552                                    ;*RPM3 MICRO TEST 02
1553                                    ;*RPM3 MICRO ERROR 05
1554                                    ;*RPM3-READ PATH-PE ID BURST DETECT
1555                                    ;*M8953, M8950'S
1556                                    ;*READ PATH STATUS INCORRECT WHEN MONITORING A BAD PE ID
1557                                    ;*ACTUAL = NNNN
1558                                    ;*EXPECTED = NNNN
1559                                    ;*EXPECTED 18 BITS =    E EEEEEE
1560                                    ;*ACTUAL 18 BITS =      A AAAAAA
1561                                    ;*EXTENDED ACTUAL PRINTOUT = AMTIE LINES
1562  4402                              S
 (1)                                    ; ************************************************************************************
1563
1564  4402                              TEST2:  TESTX   @2              ;INITIALIZE THE TEST
 (1)  4402  3E  02          7.0                  MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4404  CD  03  28     18.0                  CALL    TSET            ;SETUP THE TEST
1565                                    ;%RPM3-READ PATH-PE ID BURST DETECT
1566                                    ;&M8953, M8950'S
1567  4407  3E  10          7.0  TST02X: MVI     A,RDCLK         ;LOAD THE NORMAL READ PATH CLOCK
1568  4409  D3  F0         10.0         OUT     CLKCTL          ;SET NORMAL CLOCKS
1569  440B  CD  CD  48      18.0         CALL    CLEAR           ;CLEAR ALL TU PORTS
1570  440E  3E  80          7.0         MVI     A,R.TBJN        ;LOAD THE READ PATH ENABLE BIT
1571  4410  D3  09         10.0         OUT     RPCTL           ;ENABLE THE READ PATH
1572  4412  21  00  00      10.0         LXI     H,@0            ;INITIALIZE THE AMTIE MASK TO ZERO
1573  4415  22  FF  4A      16.0         SHLD    AMTMSK
1574  4418  3A  FE  4A      13.0  AMDLP7: LDA     SUNIT
1575  441B  D3  E0         10.0         OUT     MBSEL
1576  441D  3E  04          7.0         MVI     A,P.RPEN
1577  441F  D3  4C         10.0         OUT     PENAB
1578  4421  3E  20          7.0         MVI     A,P.LWR         ;CLEAR THE PARITY AMTIE
1579  4423  D3  48         10.0         OUT     PDIAG           ;SET THE PARITY AMTIE
1580  4425  3E  FF          7.0         MVI     A,@377          ;SET AMTIES 7:0
1581  4427  D3  44         10.0         OUT     TAMT
1582  4429  06  0A          7.0         MVI     B,10            ;PERFORM A 50 MICROSECOND
1583  442B  05              4.0  2$:     DCR     B               ;TIMEOUT
1584  442C  C2  2B  44      10.0         JNZ     2$
1585  442F  3E  02          7.0         MVI     A,RPEI          ;LOAD THE READ PE ID BURST CMD.
1586  4431  D3  0B         10.0         OUT     RCMD            ;ISSUE THE COMMAND
1587  4433  06  0A          7.0         MVI     B,10            ;PERFORM A 50 MICROSECOND
1588  4435  05              4.0  1$:     DCR     B               ;TIMEOUT
1589  4436  C2  35  44      10.0         JNZ     1$
1590  4439  DB  02         10.0         IN      RSTAT           ;GET THE READ PATH STATUS
1591  443B  A7              4.0         ANA     A               ;SET CONDITION BITS
1592  443C  CA  4B  44      10.0         JZ      AMDCN7          ;CONTINUE IF ZERO
1593  443F                               ROUT    ADATA           ;SAVE THE ACTUAL STATUS
 (1)  443F  D3  94         10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4441  7F              4.0                 MOV     A,A             ;RETRY LINK
1594  4442  AF              4.0         XRA     A               ;CLEAR THE ACCUMULATOR
1595  4443                               ROUT    EDATA           ;STORE AS EXPECTED STATUS
```

```
 (1)  4443  D3  95        10.0                OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4445  7F             4.0                MOV     A,A             ;RETRY LINK
1596  4446                                ERRB    AMDLP7,AMDCN7
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4446  CD  12  28    18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0004                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4449  04                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  444A  00                              .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  444B  CD  15  28    18.0        AMDCN7::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  444E  DA  18  44    10.0                JC      AMDLP7          ;LOOP ADDRESS IF LOOP SPECIFIED
1597                            ;>READ PATH STATUS INCORRECT WHEN MONITORING A GOOD PE ID
1598  4451  3A  00  4B    13.0  AMDLP8: LDA     AMTMSK+1        ;SET THE AMTIE PARITY BIT
1599  4454  F6  20          7.0        ORI     P.LWR           ;SET LOOP W/R
1600  4456  D3  48         10.0        OUT     PDIAG           ;AS PER THE MASK
1601  4458  3A  FF  4A    13.0        LDA     AMTMSK          ;SET THE AMTIE BITS 7:0
1602  445B  D3  44         10.0        OUT     TAMT            ;AS PER THE MASK
1603  445D  06  0A          7.0        MVI     B,10            ;PERFORM A 50 MICROSECOND
1604  445F  05              4.0  2$:    DCR     B               ;TIMEOUT
1605  4460  C2  5F  44     10.0        JNZ     2$
1606  4463  DB  02         10.0        IN      RSTAT           ;GET THE READ PATH STATUS
1607  4465  FE  90          7.0        CPI     BADST           ;BAD STATUS?
1608  4467  CA  8C  44     10.0        JZ      AMDCN8          ;YES-CONTINUE
1609  446A                            ROUT    ADATA           ;NO-STORE ACTUAL STATUS
 (1)  446A  D3  94         10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  446C  7F              4.0                MOV     A,A             ;RETRY LINK
1610  446D  3E  90          7.0        MVI     A,BADST         ;GET EXPECTED STATUS
1611  446F                            ROUT    EDATA           ;STORE EXPECTED STATUS
 (1)  446F  D3  95         10.0                OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4471  7F              4.0                MOV     A,A             ;RETRY LINK
1612  4472  3A  FF  4A    13.0        LDA     AMTMSK          ;GET AMTIE BITS 7:0
1613  4475                            ROUT    R16L            ;STORE IN CAS
 (1)  4475  D3  9C         10.0                OUT     R16L            ;WRITE AC INTO R16L
 (1)  4477  7F              4.0                MOV     A,A             ;RETRY LINK
1614  4478  3A  00  4B    13.0        LDA     AMTMSK+1        ;GET AMTIE PARITY BIT
1615  447B                            ROUT    R16H            ;STORE IN CAS
 (1)  447B  D3  9D         10.0                OUT     R16H            ;WRITE AC INTO R16H
 (1)  447D  7F              4.0                MOV     A,A             ;RETRY LINK
1616  447E  3E  80          7.0        MVI     A,@200          ;SET THE ACTUAL DATA PRINT FLAG
1617  4480                            ROUT    R17L            ;STORE IN CAS
 (1)  4480  D3  9E         10.0                OUT     R17L            ;WRITE AC INTO R17L
 (1)  4482  7F              4.0                MOV     A,A             ;RETRY LINK
1618  4483  AF              4.0        XRA     A               ;CLEAR THE EXPECTED DATA
1619  4484                            ROUT    R15L            ;PRINT FLAG
 (1)  4484  D3  9A         10.0                OUT     R15L            ;WRITE AC INTO R15L
 (1)  4486  7F              4.0                MOV     A,A             ;RETRY LINK
```

```
1621  4487                                          ERRB    AMDLP8,AMDCN8,@5
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4487  CD  12  28      18.0                    CALL    ERLPR           ;PROCESS ERROR - DO 2.3
 (1)       0005                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  448A  05                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  448B  05                                      .BYTE   @5              ;PRINT ROUTINE NUMBER
 (1)  448C  CD  15  28      18.0            AMDCN8::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  448F  DA  51  44      10.0                    JC      AMDLP8          ;LOOP ADDRESS IF LOOP SPECIFIED
1622                                        ;>READ PATH STATUS INCORRECT WHEN MONITORING A BAD PE ID
1623                                        ;<EXTENDED ACTUAL PRINTOUT = AMTIE LINES
1624  4492  2A  FF  4A       *6.0           3$:     LHLD    AMTMSK          ;GET THE AMTIE MASK
1625  4495  23                  6.0                 INX     H               ;INCREMENT
1626  4496  22  FF  4A        16.0                 SHLD    AMTMSK          ;SAVE THE AMTIE MASK
1627  4499  3A  00  4B        13.0                 LDA     AMTMSK+1        ;GET THE AMTIE MASK PARITY BIT
1628  449C  FE  02                7.0             CPI     @2              ;DONE?
1629  449E  CA  B1  44          10.0             JZ      AMDLP9          ;YES-END OF TEST
1630  44A1  FE  01                7.0             CPI     @1              ;RESULT > 377?
1631  44A3  CA  18  44          10.0             JZ      AMDLP7          ;YES-CONTINUE
1632  44A6  3A  FF  4A          13.0             LDA     AMTMSK          ;NO-GET THE AMTIE DATA MASK
1633  44A9  FE  FF                7.0             CPI     @377            ;=377?
1634  44AB  CA  92  44          10.0             JZ      3$              ;YES-INCREMENT AGAIN
1635  44AE  C3  18  44          10.0             JMP     AMDLP7          ;NO-CONTINUE
1636
1637  44B1                                  AMDLP9: ENDTST  TST02X
 (1)                                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44B1                                          REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44B1  CD  06  28      18.0                            CALL    REQST
 (2)  44B4  00                                              .BYTE                   ;DATA PATTERN NUMBER
 (2)  44B5  00  00                                          .WORD                   ;SYSTEM '"' COUNT
 (2)  44B7  00  00                                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  44B9  00                                              .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  44BA  07                                              .BYTE   7               ;REQUEST CODE
 (1)  44BB  3A  9A  4F      13.0                    LDA     ITERA           ;GET ITERATION COUNT
 (1)  44BE  3D                4.0                   DCR     A               ;DOWNCOUNT
 (1)  44BF  32  9A  4F      13.0                    STA     ITERA           ;SAVE COUNT
 (1)  44C2  F2  07  44      10.0                    JP      TST02X          ;DO TEST UNTIL TILL = 0
1638  44C5  CD  CD  48      18.0                    CALL    CLEAR
1639
1640
```

```
                                .SBTTL  TEST 03 - GCR ID BURST TEST

1644  44C8                      ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1645                            ;*GCR ID BURST TEST
1646  44C8                      SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1647                            ;*THIS TEST IS DESIGNED TO CHECK THE READ PATHS ABILITY TO DETECT A
1648                            ;*CORRECT GCR IDENTIFICATION BURST AND TO REJECT INCORRECT ONES.
1649  44C8                      SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1650                            ;*BGNTST
1651                            ;*    SET NORMAL READ PATH CLOCK
1652                            ;*    CALL SUBROUTINE CLEAR
1653                            ;*    ENABLE THE READ MICRO CONTROLLER
1654                            ;*    INIT THE AMTIE MASK TO ZERO (9 BITS WIDE)
1655                            ;*    BGNDO
1656                            ;*      :   SET UP THE AMTIES FOR A GOOD GCR ID BURST
1657                            ;*      :   DELAY 50 MICROSECONDS
1658                            ;*      :   ISSUE THE ID BURST COMMAND
1659                            ;*      :   DELAY 50 MICROSECONDS
1660                            ;*      :   READ THE READ MICRO STATUS
1661                            ;*      :       IF THE STATUS = 0
1662                            ;*      :       :   THEN-CONTINUE
1663                            ;*      :       :   ELSE-ERROR
1664                            ;*      :       ENDIF
1665                            ;*      :   ISSUE THE AMTIE LINES AS PER THE AMTIE MASK
1666                            ;*      :   DELAY 50 MICROSECONDS
1667                            ;*      :   READ THE READ MICRO STATUS
1668                            ;*      :       IF STATUS = ''BADST''
1669                            ;*      :       :   THEN-CONTINUE
1670                            ;*      :       :   ELSE-ERROR
1671                            ;*      :       ENDIF
1672                            ;*      :   INCREMENT THE AMTIE MASK BY ONE
1673                            ;*      :       IF AMTIE MASK = 677(8)
1674                            ;*      :       :   THEN-INCREMENT THE AMTIE MASK BY ONE
1675                            ;*      :       :   ELSE-ERROR
1676                            ;*      :       ENDIF
1677                            ;*      :   DO UNTIL AMTIE MASK - 1000(8)
1678                            ;*    ENDDO
1679                            ;*ENDTST
1680  44C8                      SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1681                            ;*RPM3 MICRO TEST 03
1682                            ;*RPM3 MICRO ERROR 06
1683                            ;*RPM3-READ PATH-GCR ID BURST DETECT
```

```
1684                                    ;*M8953, M8950'S
1685                                    ;*READ PATH STATUS INCORRECT WHEN MONITORING A GOOD GCR ID
1686                                    ;*ACTUAL = NNNN
1687                                    ;*EXPECTED = NNNN
1688                                    ;*
1689                                    ;*RPM3 MICRO TEST 03
1690                                    ;*RPM3 MICRO ERROR 07
1691                                    ;*RPM3-READ PATH-GCR ID BURST DETECT
1692                                    ;*READ PATH STATUS INCORRECT WHEN MONITORING A BAD PE ID
1693                                    ;*ACTUAL = NNNN
1694                                    ;*EXPECTED = NNNN
1695                                    ;*EXPECTED 18 BITS =    E EEEEEE
1696                                    ;*ACTUAL 18 BITS =      A AAAAAA
1697                                    ;*EXTENDED ACTUAL PRINTOUT = AMTIE LINES
1698  44C8                              $
 (1)                                    ; **********-**********-**********-**********-**********-**********-**********-**********
1699
1700  44C8                              TEST3:  TESTX   @3              ;INITIALIZE THE TEST
 (1)  44C8  3E  03          7.0                  MVI     A,@3            ;DEFINE THE TEST NUMBER
 (1)  44CA  CD  03  28     18.0                  CALL    TSET            ;SETUP THE TEST
1701                                    ;%RPM3-READ PATH-GCR ID BURST DETECT
1702                                    ;&M8953, M8950'S
1703
1704  44CD  3E  10          7.0  TST03X: MVI     A,RDCLK         ;LOAD THE NORMAL READ PATH CLOCK
1705  44CF  D3  F0         10.0           OUT     CLKCTL          ;SET NORMAL CLOCKS
1706  44D1  CD  CD  48     18.0           CALL    CLEAR           ;CLEAR ALL TU PORTS
1707
1708  44D4  3E  80          7.0           MVI     A,R.TBJN        ;LOAD THE READ PATH ENABLE BIT
1709  44D6  D3  09         10.0           OUT     RPCTL           ;ENABLE THE READ PATH
1710  44D8  21  00  00     10.0           LXI     H,@0            ;INITIALIZE THE AMTIE TO ZERO
1711  44DB  22  FF  4A     16.0           SHLD    AMTMSK
1712  44DE  3A  FE  4A     13.0  AMDLPA: LDA     SUNIT
1713  44E1  D3  E0         10.0           OUT     MBSEL
1714  44E3  3E  04          7.0           MVI     A,P.RPEN
1715  44E5  D3  4C         10.0           OUT     PENAB
1716  44E7  3E  21          7.0           MVI     A,P.LWR!@1      ;SET THE PARITY AMTIE
1717  44E9  D3  48         10.0           OUT     PDIAG           ;SET THE PARITY AMTIE
1718  44EB  3E  BF          7.0           MVI     A,@277          ;SET AMTIES
1719  44FD  D3  44         10.0           OUT     TAMT
1720  44EF  06  0A          7.0           MVI     B,10            ;PERFORM A 50 MICROSECOND
1721  44F1  05              4.0  2$:      DCR     B               ;TIMEOUT
1722  44F2  C2  F1  44     10.0           JNZ     2$
1723  44F5  3E  03          7.0           MVI     A,RGCRI         ;LOAD THE READ GCR ID BURST CMD.
1724  44F7  D3  0B         10.0           OUT     RCMD            ;ISSUE THE COMMAND
1725
1726  44F9  06  0A          7.0           MVI     B,10            ;PERFORM A 50 MICROSECOND
1727  44FB  05              4.0  1$:      DCR     B               ;TIMEOUT
1728  44FC  C2  FB  44     10.0           JNZ     1$
1729
1730  44FF  DB  02         10.0           IN      RSTAT           ;GET THE READ PATH STATUS
1731  4501  A7              4.0           ANA     A               ;SET CONDITION BITS
1732  4502  CA  11  45     10.0           JZ      AMDCNA          ;CONTINUE IF ZERO
1733  4505                                ROUT    ADATA           ;SAVE THE ACTUAL STAIUS
 (1)  4505  D3  94         10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
```

```
 (1) 4507 7F                  4.0                 MOV     A,A             ;RETRY LINK
1734 4508 AF                  4.0         XRA     A               ;CLEAR THE ACCUMULATOR
1735 4509                             ROUT    EDATA           ;STORE AS EXPECTED STATUS
 (1) 4509 D3 95               10.0                OUT     EDATA           ;WRITE AC INTO EDATA
 (1) 450B 7F                  4.0                 MOV     A,A             ;RETRY LINK
1736 450C                           ERRB    AMDLPA,AMDCNA
 (1)                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1) 450C CD 12 28            18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0006                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 450F 06                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 4510 00                                     .BYTE                   ;PRINT ROUTINE NUMBER
 (1) 4511 CD 15 28            18.0                AMDCNA::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1) 4514 DA DE 44            10.0                JC      AMDLPA          ;LOOP ADDRESS IF LOOP SPECIFIED
1737                        ;>READ PATH STATUS INCORRECT WHEN MONITORING A GOOD GCR ID
1738
1739 4517 3A 00 4B           13.0   AMDLPB: LDA      AMTMSK+1        ;SET THE AMTIE PARITY BIT
1740 451A F6 20               7.0           ORI     P.LWR           ;SET LOOP W/R
1741 451C D3 48              10.0           OUT     PDIAG           ;AS PER THE MASK
1742 451E 3A FF 4A           13.0           LDA     AMTMSK          ;SET THE AMTIE BITS 7:0
1743 4521 D3 44              10.0           OUT     TAMT            ;AS PER THE MASK
1744 4523 06 0A               7.0           MVI     B,10            ;PERFORM A 50 MICROSECOND
1745 4525 05                  4.0   2$:     DCR     B               ;TIMEOUT
1746 4526 C2 25 45           10.0           JNZ     2$
1747 4529 DB 02              10.0           IN      RSTAT           ;GET THE READ PATH STATUS
1748 452B FE 90               7.0           CPI     BADST           ;BAD STATUS?
1749 452D CA 52 45           10.0           JZ      AMDCNB          ;YES-CONTINUE
1750 4530                             ROUT    ADATA           ;NO-STORE ACTUAL STATUS
 (1) 4530 D3 94               10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1) 4532 7F                  4.0                 MOV     A,A             ;RETRY LINK
1751 4533 3E 90               7.0         MVI     A,BADST         ;GET EXPECTED STATUS
1752 4535                           ROUT    EDATA           ;STORE EXPECTED STATUS
 (1) 4535 D3 95               10.0                OUT     EDATA           ;WRITE AC INTO EDATA
 (1) 4537 7F                  4.0                 MOV     A,A             ;RETRY LINK
1753
1754 4538 3A FF 4A           13.0         LDA     AMTMSK          ;GET AMTIE BITS 7:0
1755 453B                           ROUT    R16L            ;STORE IN CAS
 (1) 453B D3 9C               10.0                OUT     R16L            ;WRITE AC INTO R16L
 (1) 453D 7F                  4.0                 MOV     A,A             ;RETRY LINK
1756 453E 3A 00 4B           13.0         LDA     AMTMSK+1        ;GET AMTIE PARITY BIT
1757 4541                           ROUT    R16H            ;STORE IN CAS
 (1) 4541 D3 9D               10.0                OUT     R16H            ;WRITE AC INTO R16H
 (1) 4543 7F                  4.0                 MOV     A,A             ;RETRY LINK
1758 4544 3E 80               7.0         MVI     A,@200          ;SET THE ACTUAL DATA PRINT FLAG
1759 4546                           ROUT    R17L            ;STORE IN CAS
 (1) 4546 D3 9E               10.0                OUT     R17L            ;WRITE AC INTO R17L
 (1) 4548 7F                  4.0                 MOV     A,A             ;RETRY LINK
1760 4549 AF                  4.0         XRA     A               ;CLEAR THE EXPECTED DATA
1761 454A                           ROUT    R15I            ;PRINT FLAG
 (1) 454A D3 9A               10.0                OUT     R15L            ;WRITE AC INTO R15L
 (1) 454C 7F                  4.0                 MOV     A,A             ;RETRY LINK
```

```
1763  454D                                          ERRB    AMDLPB,AMDCNB,@5
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  454D  CD  12  28      18.0                            CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0007                                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4550  07                                              .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4551  05                                              .BYTE   @5              ;PRINT ROUTINE NUMBER
 (1)  4552  CD  15  28      18.0                    AMDCNB::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4555  DA  17  45      10.0                            JC      AMDLPB          ;LOOP ADDRESS IF LOOP SPECIFIED
1764                                          ;>READ PATH STATUS INCORRECT WHEN MONITORING A BAD PE ID
1765                                          ;<EXTENDED ACTUAL PRINTOUT = AMTIE LINES
1766  4558  2A  FF  4A        16.0  3$:      LHLD    AMTMSK          ;GET THE AMTIE MASK
1767  455B  23                   6.0          INX     H               ;INCREMENT
1768  455C  22  FF  4A        16.0          SHLD    AMTMSK          ;SAVE THE AMTIE MASK
1769  455F  3A  00  4B        13.0          LDA     AMTMSK+1        ;GET THE PARITY AMTIE MASK
1770  4562  FE  02                7.0          CPI     @2              ;DONE?
1771  4564  CA  77  45        10.0          JZ      AMDLPC          ;YES-EXIT THE TEST
1772  4567  FE  01                7.0          CPI     @1              ;NO-MASK=677?
1773  4569  C2  DE  44        10.0          JNZ     AMDLPA          ;NO-CONTINUE
1774  456C  3A  FF  4A        13.0          LDA     AMTMSK          ;MAYBE
1775  456F  FE  BF                7.0          CPI     @277            ;LO MASK=277?
1776  4571  CA  58  45        10.0          JZ      3$              ;YES-INCREMENT MASK AGAIN
1777  4574  C3  DE  44        10.0          JMP     AMDLPA          ;NO-CONTINUE
1778
1779  4577                                  AMDLPC: ENDTST  TST03X
 (1)                                          ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4577  CD  06  28      18.0                            REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  457A  00                                              CALL    REQST
 (2)  457B  00  00                                          .BYTE                   ;DATA PATTERN NUMBER
 (2)  457D  00  00                                          .WORD                   ;SYSTEM "" COUNT
 (2)  457F  00                                              .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4580  07                                              .BYTE                   ;DATA COMPARE FLAG IF =1
 (1)  4581  3A  9A  4F      13.0                            .BYTE   7               ;REQUEST CODE
 (1)  4584  3D                4.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4585  32  9A  4F      13.0          DCR     A               ;DOWNCOUNT
 (1)  4588  F2  CD  44      10.0          STA     ITERA           ;SAVE COUNT
1780  458B  CD  CD  48      18.0          JP      TST03X          ;DO TEST UNTIL TILL = 0
1781                                      CALL    CLEAR
```

```
1783                                .SBTTL  TEST 04 - GCR ARA BURST TEST
1784
1785  458E                          ST
 (1)                                ; ************************************************************************
 (1)                                ;*TEST TITLE
 (1)                                ;*---- -----
1786                                ;*GCR ARA BURST TEST
1787  458E                          SD
 (1)                                ; ************************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
1788  *                             ;*THIS TEST CHECKS THE READ PATHS ABILITY TO DETECT A CORRECT GCR ARA
1789                                ;*ID BURST, AND REJECT INCORRECT ONES.
1790  458E                          SP
 (1)                                ; ************************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
1791                                ;*BGNTST
1792                                ;*    SET NORMAL READ PATH CLOCK
1793                                ;*    CALL SUBROUTINE CLEAR
1794                                ;*    ENABLE THE READ MICRO CONTROLELR
1795                                ;*    INIT THE AMTIE MASK TO 001(8) (9 BITS WIDE)
1796                                ;*      BGNDO
1797                                ;*       :   CLEAR ALL AMTIES FOR A GOOD ARA BURST
1798                                ;*       :   DELAY 50 MICROSECONDS
1799                                ;*       :   ISSUE THE ARA BURST COMMAND
1800                                ;*       :   DELAY 50 MICRO SECONDS
1801                                ;*       :   READ THE READ MICRO STATUS
1802                                ;*       :       IF STATUS = 0
1803                                ;*       :       :   THEN-CONTINUE
1804                                ;*       :       :   ELSE-ERROR
1805                                ;*       :       ENDIF
1806                                ;*       :   ISSUE THE AMTIE LINES AS PER THE AMTIF MASK
1807                                ;*       :   DELAY 50 MICRO SECONDS
1808                                ;*       :   READ THE READ MICRO STATUS
1809                                ;*       :       IF STATUS = "BADST"
1810                                ;*       :       :   THEN-CONTINUE
1811                                ;*       :       :   ELSE-ERROR
1812                                ;*       :       ENDIF
1813                                ;*       :   INCREMENT THE AMTIE MASK BY ONE
1814                                ;*       .   DO UNTIL THE AMTIE MASK = 1000(8)
1815                                ;*    ENDDO
1816                                ;*ENDTST
1817  458E                          SE
 (1)                                ; ************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1818                                ;*RPM3 MICRO TEST 04
1819                                ;*RPM3 MICRO ERROR 10
1820                                ;*RPM3-READ PATH-GCR ARA BURST DETECT TEST
1821                                ;*M8953, M8950'S
1822                                ;*READ PATH STATUS INCORRECT WHEN MONITORING A GOOD ARA BURST
1823                                ;*ACTUAL = NNNN
1824                                ;*EXPECTED = NNNN
```

```
                                        ;*
                                        ;*RPM3 MICRO TEST 04
                                        ;*RPM3 MICRO ERROR 11
                                        ;*RPM3-READ PATH-GCR ARA BURST DETECT TEST
                                        ;*M8953, M8950'S
                                        ;*READ PATH STATUS INCORRECT WHEN MONITORING A BAD ARA BURST
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*EXPECTED 18 BITS =    E EEEEEE
                                        ;*ACTUAL 18 BITS =      A AAAAAA
                                        ;*EXTENDED ACTUAL PRINTOUT=AMTIE LINES
458E                                    $
 (1)                                    ; ************************************************************************************
                                        TEST4:  TESTX   @4              ;INITIALIZE THE TEST
 (1) 458E  3E  04          7.0                         MVI     A,@4            ;DEFINE THE TEST NUMBER
 (1) 4590  CD  03  28     18.0                         CALL    TSET            ;SETUP THE TEST
                                        ;%RPM3-READ PATH-GCR ARA BURST DETECT TEST
                                        ;&M8953, M8950'S

4593  3E  10                7.0  TST04X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCK
4595  D3  F0               10.0          OUT     CLKCTL
4597  CD  CD  48           18.0          CALL    CLEAR           ;CLEAR ALL TU PORTS

459A  3E  80                7.0          MVI     A,R.TBJN                ;ENABLE THE READ MICRO CONTROLLER
459C  D3  09               10.0          OUT     RPCTL

459E  21  01  00           10.0          LXI     H,@1            ;INIT THE AMTIE MASK TO 1
45A1  22  FF  4A           16.0          SHLD    AMTMSK

45A4  3A  FE  4A           13.0  AMDLPD: LDA     SUNIT
45A7  D3  E0               10.0          OUT     MBSEL
45A9  3E  04                7.0          MVI     A,P.RPEN
45AB  D3  4C               10.0          OUT     PENAB
45AD  3E  20                7.0          MVI     A,P.LWR         ;CLEAR THE PARITY AMTIE
45AF  D3  48               10.0          OUT     PDIAG
45B1  AF                    4.0          XRA     A               ;CLEAR ALL AMTIES
45B2  D3  44               10.0          OUT     TAMT
45B4  06  0A                7.0          MVI     B,10            ;PERFORM A 50 MICROSECOND
45B6  05                    4.0  2$:     DCR     B               ;TIMEOUT
45B7  C2  B6  45           10.0          JNZ     2$

45BA  3E  06                7.0          MVI     A,RARA          ;LOAD THE ARA BURST COMMAND
45BC  D3  0B               10.0          OUT     RCMD            ;ISSUE THE COMMAND
45BE  06  0A                7.0          MVI     B,10            ;PERFORM A 50 MICROSECOND
45C0  05                    4.0  1$:     DCR     B               ;TIMEOUT
45C1  C2  C0  45           10.0          JNZ     1$

45C4  DB  02               10.0          IN      RSTAT           ;GET THE READ PATH STATUS
45C6  A7                    4.0          ANA     A               ;SET CONDITION BITS
45C7  CA  D6  45           10.0          JZ      AMDCND          ;CONTINUE IF ZERO
45CA                                    ROUT    ADATA           ;SAVE THE ACTUAL STATUS
 (1) 45CA  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1) 45CC  7F               4.0                  MOV     A,A             ;RETRY LINK
```

```
1874  45CD  AF                    4.0          XRA     A                ;CLEAR THE ACCUMULATOR
1875  45CE                                     ROUT    EDATA            ;STORE AS EXPECTED STATUS
 (1)  45CE  D3  95                10.0                 OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  45D0  7F                    4.0                  MOV     A,A              ;RETRY LINK
1876  45D1                                     ERRB    AMDLPD,AMDCND
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  45D1  CD  12  28            18.0                 CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)  0008                                             MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45D4  08                                         .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  45D5  00                                         .BYTE                    ;PRINT ROUTINE NUMBER
 (1)  45D6  CD  15  28            18.0         AMDCND::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45D9  DA  A4  45            10.0                 JC      AMDLPD           ;LOOP ADDRESS IF LOOP SPECIFIED
1877                                 ;>READ PATH STATUS INCORRECT WHEN MONITORING A GOOD ARA BURST
1878
1879  45DC  3A  00  4B            13.0  AMDLPE: LDA     AMTMSK+1         ;SET THE AMTIE PARITY BIT
1880  45DF  F6  20                 7.0          ORI     P.LWR            ;SET LOOP W/R
1881  45E1  D3  48                10.0          OUT     PDIAG            ;AS PER THE MASK
1882  45E3  3A  FF  4A            13.0          LDA     AMTMSK           ;SET AMTIE BITS 7:0
1883  45E6  D3  44                10.0          OUT     TAMT             ;AS PER THE MASK
1884
1885  45E8  06  0A                 7.0          MVI     B,10             ;PERFORM A 50 MICRO SECOND
1886  45EA  05                     4.0  2$:     DCR     B                ;TIMEOUT
1887  45EB  C2  EA  45            10.0          JNZ     2$
1888  45EE  DB  02                10.0          IN      RSTAT            ;GET THE READ PATH STATUS
1889  45F0  FE  90                 7.0          CPI     BADST            ;BAD STATUS
1890  45F2  CA  17  46            10.0          JZ      AMDCNE           ;YES-CONTINUE
1891  45F5                                     ROUT    ADATA            ;NO-STORE ACTUAL STATUS
 (1)  45F5  D3  94                10.0                 OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  45F7  7F                    4.0                  MOV     A,A              ;RETRY LINK
1892  45F8  3E  90                 7.0          MVI     A,BADST          ;GET EXPECTED STATUS
1893  45FA                                     ROUT    EDATA            ;STORE EXPECTED STATUS
 (1)  45FA  D3  95                10.0                 OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  45FC  7F                    4.0                  MOV     A,A              ;RETRY LINK
1894  45FD  3A  FF  4A            13.0          LDA     AMTMSK           ;GET THE AMTIE BITS 7:0
1895  4600                                     ROUT    R16L             ;STORE IN CAS
 (1)  4600  D3  9C                10.0                 OUT     R16L             ;WRITE AC INTO R16L
 (1)  4602  7F                    4.0                  MOV     A,A              ;RETRY LINK
1896  4603  3A  00  4B            13.0          LDA     AMTMSK+1         ;GET AMTIE PARITY
1897  4606                                     ROUT    R16H             ;STORE IN CAS
 (1)  4606  D3  9D                10.0                 OUT     R16H             ;WRITE AC INTO R16H
 (1)  4608  7F                    4.0                  MOV     A,A              ;RETRY LINK
1898  4609  3E  80                 7.0          MVI     A,@200           ;SET ACTUAL DATA PRINT FALG
1899  460B                                     ROUT    R17L             ;CLEAR THE EXPECTED DATA PRINT
 (1)  460B  D3  9E                10.0                 OUT     R17L             ;WRITE AC INTO R17L
 (1)  460D  7F                    4.0                  MOV     A,A              ;RETRY LINK
1900  460E  AF                     4.0          XRA     A                ;FLAG
1901  460F                                     ROUT    R15L
 (1)  460F  D3  9A                10.0                 OUT     R15L             ;WRITE AC INTO R15L
 (1)  4611  7F                    4.0                  MOV     A,A              ;RETRY LINK
```

```
1903  4612                                          ERRB    AMDLPE,AMDCNE,@5
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4612  CD  12  28        18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0009                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4615  09                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4616  05                                            .BYTE   @5              ;PRINT ROUTINE NUMBER
 (1)  4617  CD  15  28        18.0                AMDCNE::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  461A  DA  DC  45        10.0                        JC      AMDLPE          ;LOOP ADDRESS IF LOOP SPECIFIED
1904                                        ;>READ PATH STATUS INCORRECT WHEN MONITORING A BAD ARA BURST
1905                                        ;<EXTENDED ACTUAL PRINTOUT=AMTIE LINES
1906  461D  2A  FF  4A        16.0                LHLD    AMTMSK          ;GET THE AMTIE MASK
1907  4620  23                  6.0                INX     H               ;INCREMENT THE MASK
1908  4621  22  FF  4A        16.0                SHLD    AMTMSK          ;SAVE THE AMTIE MASK
1909  4624  3A  00  4B        13.0                LDA     AMTMSK+1        ;GET AMTIE PARITY MASK
1910  4627  FE  02              7.0                CPI     @2              ;DONE?
1911  4629  C2  A4  45        10.0                JNZ     AMDLPD          ;NO-CONTINUE
1912  462C                                        ENDTST  TST04X
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  462C                                          REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  462C  CD  06  28        18.0                        CALL    REQST
 (2)  462F  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2)  4630  00  00                                        .WORD                   ;SYSTEM "" COUNT
 (2)  4632  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4634  00                                            .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4635  07                                            .BYTE   7               ;REQUEST CODE
 (1)  4636  3A  9A  4F        13.0                LDA     ITERA           ;GET ITERATION COUNT
 (1)  4639  3D                 4.0                DCR     A               ;DOWNCOUNT
 (1)  463A  32  9A  4F        13.0                STA     ITERA           ;SAVE COUNT
 (1)  463D  F2  93  45        10.0                JP      TST04X          ;DO TEST UNTIL TILL = 0
1913  4640  CD  CD  48        18.0                CALL    CLEAR
```

```
                                .SBTTL  TEST 05 - GCR ARA ID CHARACTER TEST
4643                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*GCR ARA ID CHARACTER TEST
4643                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE READ PATHS ABILITY TO DETECT A CORRECT GCR ARA
                                ;*ID CHARACTERS, AND REJECT INCORRECT ONES.
4643                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    ENABLE THE READ MICRO CONTROLLER
                                ;*    INIT THE AMTIE MASK TO ZERO (9 BITS)
                                ;*    BGNDO
                                ;*      :   SET UP THE AMTIES FOR A GOOD ARA BURST
                                ;*      :   DELAY 50 MICROSECONDS
                                ;*      :   ISSUE THE ARA ID CHARACTER COMMAND
                                ;*      :   DELAY 50 MICRO SECONDS
                                ;*      :   READ THE READ MICRO STATUS
                                ;*      :       IF THE STATUS = 0
                                ;*      :       :   THEN-CONTINUE
                                ;*      :       :   ELSE-ERROR
                                ;*      :       ENDIF
                                ;*      :   ISSUE THE AMTIE LINES AS PER THE AMTIE MASK
                                ;*      :   DELAY 50 MICRO SECONDS
                                ;*      :   READ THE READ MICRO STATUS
                                ;*      :       IF STATUS = ''BADST''
                                ;*      :       :   THEN-CONTINUE
                                ;*      :       :   ELSE-ERROR
                                ;*      :       ENDIF
                                ;*      :   INCREMENT THE AMTIE MASK BY 1
                                ;*      :       IF AMTIE MASK = 604(8)
                                ;*      :       :   THEN-INCREMENT THE AMTIE MASK BY 1
                                ;*      :       :   ELSE-CONTINUE
                                ;*      :       ENDIF
                                ;*      :   DO UNTIL THE AMTIE MASK = 1000(8)
                                ;*    ENDDO
                                ;*ENDTST
4643                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*RPM3 MICRO TEST 05
                                ;*RPM3 MICRO ERROR 12
                                ;*RPM3-READ PATH-GCR ARA ID CHARACTER DETECT
                                ;*M8953, M8950'S
```

```
                                        ;*READ PATH STATUS INCORRECT WHEN MONITORING A GOOD APA ID CHARACTER
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM3 MICRO TEST 05
                                        ;*RPM3 MICRO ERROR 13
                                        ;*RPM3-READ PATH-GCR ARA ID CHARACTER DETECT
                                        ;*M8953, M8950'S
                                        ;*READ PATH STATUS INCORRECT WHEN MONITORING A BAD ARA ID CHARACTER
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*EXPECTED 18 BITS =    E EEEEEE
                                        ;*ACTUAL 18 BITS =      A AAAAAA
                                        ;*EXTENDED ACTUAL PRINTOUT=AMTIE LINES
4643                                    $
 (1)                                    ; *****************************************************************************
4643                                    TEST5:  TESTX   @5              ;INITIALIZE THE TEST
 (1)  4643  3E  05          7.0                  MVI     A,@5            ;DEFINE THE TEST NUMBER
 (1)  4645  CD  03  28     18.0                  CALL    TSET            ;SETUP THE TEST
                                        ;%RPM3-READ PATH-GCR ARA ID CHARACTER DETECT
                                        ;&M8953, M8950'S

4648  3E  10               7.0  TST05X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCK
464A  D3  F0              10.0          OUT     CLKCTL
464C  CD  CD  48          18.0          CALL    CLEAR           ;CLEAR ALL TU PORTS

464F  3E  8U               7.0          MVI     A,R.TBJN                ;ENABLE THE READ MICRO CONTROLER
4651  D3  09              10.0          OUT     RPCTL

4653  21  00  00          10.0          LXI     H,@0            ;CLEAR THE AMTIE MASK
4656  22  FF  4A          16.0          SHLD    AMTMSK

4659  3A  FE  4A          13.0  AMDLPG: LDA     SUNIT
465C  D3  E0              10.0          OUT     MBSEL
465E  3E  04               7.0          MVI     A,P.RPEN
4660  D3  4C              10.0          OUT     PENAB
4662  3E  21               7.0          MVI     A,P.LWR!P.AMTP  ;CLEAR THE PARITY AMTIE
4664  D3  48              10.0          OUT     PDIAG
4666  3E  84               7.0          MVI     A,@204          ;SET UP AMTIES 7:0
4668  D3  44              10.0          OUT     TAMT
466A  06  0A               7.0          MVI     B,10            ;PERFORM A 50 MICROSECOND
466C  05                   4.0  2$:     DCR     B               ;TIMEOUT
466D  C2  6C  46          10.0          JNZ     2$

4670  3E  04               7.0          MVI     A,RARAI         ;LOAD THE ARA ID CHARACTER COMMAND
4672  D3  0B              10.0          OUT     RCMD            ;ISSUE THE COMMAND

4674  06  0A               7.0          MVI     B,10            ;PERFORM A 50 MICROSECOND
4676  05                   4.0  1$:     DCR     B               ;TIMEOUT
4677  C2  76  46          10.0          JNZ     1$

467A  DB  02              10.0          IN      RSTAT           ;GET THE READ PATH STATUS
467C  A7                   4.0          ANA     A               ;SET CONDITION BITS
467D  CA  8C  46          10.0          JZ      AMDONG          ;CONTINUE IF ZERO
```

```
2008  4680                                    ROUT    ADATA           ;SAVE THE ACTUAL STATUS
 (1)  4680  D3  94          10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4682  7F               4.0                     MOV     A,A             ;RETRY LINK
2009  4683  AF               4.0             XRA     A               ;CLEAR THE ACCUMULATOR
2010  4684                                    ROUT    EDATA           ;STORE AS EXPECTED STATUS
 (1)  4684  D3  95          10.0                     OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4686  7F               4.0                     MOV     A,A             ;RETRY LINK
2011  4687                                    ERRB    AMDLPG,AMDCNG
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4687  CD  12  28      18.0                     CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      000A                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  468A  0A                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  468B  00                                     .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  468C  CD  15  28      18.0                 AMDCNG::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  468F  DA  59  46      10.0                             JC      AMDLPG          ;LOOP ADDRESS IF LOOP SPECIFIED
2012                                ;>READ PATH STATUS INCORRECT WHEN MONITORING A GOOD ARA ID CHARACTER
2013
2014  4692  3A  00  4B      13.0  AMDLPH: LDA     AMTMSK+1        ;SET THE AMTIE PARITY BIT
2015  4695  F6  20               7.0             ORI     P.LWR           ;SET LOOP W/R
2016  4697  D3  48              10.0             OUT     PDIAG           ;AS PER THE MASK
2017  4699  3A  FF  4A          13.0             LDA     AMTMSK          ;SET AMTIE BITS 7:0
2018  469C  D3  44              10.0             OUT     TAMT            ;AS PER THE MASK
2019
2020  469E  06  0A               7.0             MVI     B,10            ;PERFORM A 50 MICROSECOND
2021  46A0  05                   4.0  2$:        DCR     B               ;TIMEOUT
2022  46A1  C2  A0  46          10.0             JNZ     2$
2023
2024  46A4  DB  02              10.0             IN      RSTAT           ;GET THE READ PATH STATUS
2025  46A6  FE  90               7.0             CPI     BADST           ;BAD STATUS
2026  46A8  CA  CD  46          10.0             JZ      AMD£NH          ;YES-CONTINUE
2027  46AB                                    ROUT    ADATA           ;NO-STORE ACTUAL STATUS
 (1)  46AB  D3  94              10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46AD  7F                   4.0                     MOV     A,A             ;RETRY LINK
2028  46AE  3E  90               7.0             MVI     A,BADST         ;GET EXPECTED STATUS
2029  46B0                                    ROUT    EDATA           ;STORE EXPECTED STATUS
 (1)  46B0  D3  95              10.0                     OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  46B2  7F                   4.0                     MOV     A,A             ;RETRY LINK
2030  46B3  3A  FF  4A          13.0             LDA     AMTMSK          ;GET THE AMTIE BITS 7:0
2031  46B6                                    ROUT    R16L            ;STORE IN CAS
 (1)  46B6  D3  9C              10.0                     OUT     R16L            ;WRITE AC INTO R16L
 (1)  46B8  7F                   4.0                     MOV     A,A             ;RETRY LINK
2032  46B9  3A  00  4B          13.0             LDA     AMTMSK+1        ;GET AMTIE PARITY
2033  46BC                                    ROUT    R16H            ;STORE IN CAS
 (1)  46BC  D3  9D              10.0                     OUT     R16H            ;WRITE AC INTO R16H
 (1)  46BE  7F                   4.0                     MOV     A,A             ;RETRY LINK
2034  46BF  3E  80               7.0             MVI     A @200          ;SET ACTUAL DATA PRINT FLAG
2035  46C1                                    ROUT    R17L            ;CLEAR THE EXPECTED DATA PRINT
 (1)  46C1  D3  9E              10.0                     OUT     R17L            ;WRITE AC INTO R17L
 (1)  46C3  7F                   4.0                     MOV     A,A             ;RETRY LINK
2036  46C4  AF                   4.0             XRA     A               ;FLAG
2037  46C5                                    ROUT    R15L
 (1)  46C5  D3  9A              10.0                     OUT     R15L            ;WRITE AC INTO R15L
 (1)  46C7  7F                   4.0                     MOV     A,A             ;RETRY LINK
```

```
2039  46C8                                        ERRB    AMDLPH,AMDCNH,@5
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID .
 (1)  46C8  CD  12  28      18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000B                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46CB  0B                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46CC  05                                            .BYTE   @5              ;PRINT ROUTINE NUMBER
 (1)  46CD  CD  15  28      18.0                  AMDCNH::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46D0  DA  92  46      10.0                          JC      AMDLPH          ;LOOP ADDRESS IF LOOP SPECIFIED
2040                                    ;>READ PATH STATUS INCORRECT WHEN MONITORING A BAD ARA ID CHARACTER
2041                                    ;<EXTENDED ACTUAL PRINTOUT=AMTIE LINES
2042
2043  46D3  2A  FF  4A      16.0  1$:     LHLD    AMTMSK          ;GET THE AMTIE MASK
2044  46D6  23                     6.0          INX     H               ;UPDATE THE COUNTER
2045  46D7  22  FF  4A      16.0                 SHLD    AMTMSK          ;SAVE IT
2046  46DA  3A  00  4B      13.0                 LDA     AMTMSK+1        ;GET AMTIE MASK PARITY BIT
2047  46DD  FE  01                 7.0          CPI     @01             ;IS IT 1?
2048  46DF  C2  EA  46      10.0                 JNZ     2$              ;NO-CHECK FOR DONE
2049  46E2  3A  FF  4A      13.0                 LDA     AMTMSK          ;YES - GET THE LOW ORDER 8 BITS
2050  46E5  FE  84                 7.0          CPI     @204            ;DOES THE MASK = 604(8)
2051  46E7  CA  D3  46      10.0                 JZ      1$              ;YES-INCREMENT AGAIN
2052  46EA  3A  00  4B      13.0  2$:     LDA     AMTMSK+1        ;GET THE AMTIE PARITY MASK
2053  46ED  FE  02                 7.0          CPI     @02             ;DONE?
2054  46EF  C2  59  46      10.0                 JNZ     AMDLPG          ;NO-CONTINUE
2055  46F2                                       ENDTST  TST05X
 (1)                                    ·TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  46F2                                         REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  46F2  CD  06  28      18.0                          CALL    REQST
 (2)  46F5  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2)  46F6  00  00                                        .WORD                   ;SYSTEM "" COUNT
 (2)  46F8  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  46FA  00                                            .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  46FB  07                                            .BYTE   7               ;REQUEST CODE
 (1)  46FC  3A  9A  4F      13.0                 LDA     ITERA           ;GET ITERATION COUNT
 (1)  46FF  3D                     4.0          DCR     A               ;DOWNCOUNT
 (1)  4700  32  9A  4F      13.0                 STA     ITERA           ;SAVE COUNT
 (1)  4703  F2  48  46      10.0                 JP      TST05X          ;DO TEST UNTIL TILL = 0
2056  4706  CD  CD  48      18.0                 CALL    CLEAR           ;CLEAN UP PORTS
2057
2058
```

```
                                .SBTTL  TEST 06 - TAPE MARK TEST

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*TAPE MARK TEST
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE READ PATHS ABILITY TO DETECT A TAPE MARK, AND
                                ;*PROPERLY REJECT INCORRECT ONES.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    ENABLE THE READ MICRO CONTROLLER
                                ;*    INITIALIZE THE AMTIE MASK TO ZERO (9 BITS WIDE)
                                ;*    BGNDO
                                ;*       :    SET THE AMTIE LINES TO THE VALUE OF THE AMTIE MASK
                                ;*       :    DELAY 50 MICROSECONDS
                                ;*       :    ISSUE THE TAPE MARK COMMAND
                                ;*       :    DELAY 300 MICRO SECONDS
                                ;*       :    READ THE MICRO STATUS
                                ;*       :       IF STATUS = 0
                                ;*       :       :   THEN-CONTINUE
                                ;*       :       :   ELSE-ERROR
                                ;*       :       ENDIF
                                ;*       :    SET THE AMTIE LINES TO 130(8) FOR A GOOD TAPE MARK
                                ;*       :    DELAY 300 MICRO SECONDS
                                ;*       :    READ THE READ MICRO STATUS
                                ;*       :       IF STATUS=''GOODTM''
                                ;*       :       :   THEN-CONTINUE
                                ;*       :       :   ELSE-ERROR
                                ;*       :       ENDIF
                                ;*       :    INCREMENT THE AMTIE MASK BY 1
                                ;*       :       IF AMTIE MASK=130(8)
                                ;*       :       :   THEN-INCREMENT THE AMTIE MASK BY 1
                                ;*       :       :   ELSE-CONTINUE
                                ;*       :       ENDIF
                                ;*       :    DO UNTIL THE AMTIE MASK = 1000(8)
                                ;*    ENDDO
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM3 MICRO TEST 06
                                ;*RPM3 MICRO ERROR 14
                                ;*RPM3-READ PATH-TAPE MARK DETECT TEST
```

```
                                        ;*M8953, M8950'S
                                        ;*READ PATH STATUS INCORRECT WHEN MONITORING A BAD TAPE MARK
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*EXPECTED 18 BITS =    E EEEEEE
                                        ;*ACTUAL 18 BITS =      A AAAAAA
                                        ;*EXTENDED ACTUAL PRINTOUT = AMTIE LINES
                                        ;*
                                        ;*RPM3 MICRO TEST 06
                                        ;*RPM3 MICRO ERROR 15
                                        ;*RPM3-READ PATH-TAPE MARK DETECT TEST
                                        ;*M8953, M8950'S
                                        ;*READ PATH STATUS INCORRECT WHEN MONITORING A GOOD TAPE MARK
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4709                                    $
 (1)                                    ; ************************************************************************************
4709                                    TEST6:  TESTX   @6              ;INITIALIZE THE TEST
 (1)  4709  3E  06            7.0                MVI     A,@6            ;DEFINE THE TEST NUMBER
 (1)  470B  CD  03  28       18.0                CALL    ISET            ;SETUP THE TEST
                                        ;%RPM3-READ PATH-TAPE MARK DETECT TEST
                                        ;&M8953, M8950'S

470E  3E  10                 7.0  TST06X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCK
4710  D3  F0                10.0         OUT     CLKCTL
4712  CD  CD  48            18.0         CALL    CLEAR           ;CLEAR ALL TU PORTS
4715  D3  09                10.0         OUT     RPCTL
4717  21  00  00            10.0         LXI     H,@0            ;CLEAR THE AMTIE MASK
471A  22  FF  4A            16.0         SHLD    AMTMSK
471D  3A  FE  4A            13.0  AMDLPI: LDA     SUNIT
4720  D3  E0                10.0         OUT     MBSEL
4722  3E  04                 7.0         MVI     A,P.RPEN
4724  D3  4C                10.0         OUT     PENAB
4726  3A  00  4B            13.0         LDA     AMTMSK+1        ;SET THE AMTIE PARITY BIT
4729  F6  20                 7.0         ORI     P.LWR           ;SET LOOP W/R
472B  D3  48                10.0         OUT     PDIAG           ;AS PER THE AMTIE MASK
472D  3A  FF  4A            13.0         LDA     AMTMSK          ;SET AMTIE BITS 7:0
4730  D3  44                10.0         OUT     TAMT            ;AS PER THE AMTIE MASK
4732  06  0A                 7.0         MVI     B,10            ;PERFORM A 50 MICROSECOND
4734  05                    4.0  2$:     DCR     B               ;TIMEOUT
4735  C2  34  47            10.0         JNZ     2$
4738  3E  05                 7.0         MVI     A,RTM           ;LOAD THE TEST TAPE MARK COMMAND
473A  D3  0B                10.0         OUT     RCMD            ;ISSUE THE COMMAND

473C  06  3C                 7.0         MVI     B,60            ;PERFORM A 300 MICRO SECOND
473E  05                    4.0  1$:     DCR     B               ;TIMEOUT
473F  C2  3E  47            10.0         JNZ     1$

4742  DB  02                10.0         IN      RSTAT           ;GET THE READ PATH STATUS
4744  A7                    4.0         ANA     A               ;SET CONDITION BITS
4745  CA  69  47            10.0         JZ      AMDCNI          ;CONTINUE IF ZERO
4748                                      ROUT    ADATA
 (1)  4748  D3  94            10.0                 OUT     ADATA          ;WRITE AC INTO ADATA
 (1)  474A  7F                 4.0                 MOV     A,A            ;RETRY LINK
```

```
2151  474B  AF                    4.0   XRA     A
2152  474C                              ROUT    EDATA
 (1)  474C  D3   95                10.0          OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  474E  7F                     4.0          MOV     A,A             ;RETRY LINK
2153  474F  3A   FF   4A           13.0   LDA     AMTMSK
2154  4752                              ROUT    R16L
 (1)  4752  D3   9C                10.0          OUT     R16L            ;WRITE AC INTO R16L
 (1)  4754  7F                     4.0          MOV     A,A             ;RETRY LINK
2155  4755  3A   00   4B           13.0   LDA     AMTMSK+1
2156  4758                              ROUT    R16H
 (1)  4758  D3   9D                10.0          OUT     R16H            ;WRITE AC INTO R16H
 (1)  475A  7F                     4.0          MOV     A,A             ;RETRY LINK
2157  475B  3E   80                 7.0   MVI     A,@200
2158  475D                              ROUT    R17L
 (1)  475D  D3   9E                10.0          OUT     R17L            ;WRITE AC INTO R17L
 (1)  475F  7F                     4.0          MOV     A,A             ;RETRY LINK
2159  4760  AF                    4.0   XRA     A
2160  4761                              ROUT    R15L
 (1)  4761  D3   9A                10.0          OUT     R15L            ;WRITE AC INTO R15L
 (1)  4763  7F                     4.0          MOV     A,A             ;RETRY LINK
2161  4764                              ERRB    AMDLPI,AMDCNI,@5
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4764  CD   12   28           18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000C                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4767  0C                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4768  05                            .BYTE   @5              ;PRINT ROUTINE NUMBER
 (1)  4769  CD   15   28           18.0  AMDCNI::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  476C  DA   1D   47           10.0                  JC      AMDLPI          ;LOOP ADDRESS IF LOOP SPECIFIED
2162                                  ;>READ PATH STATUS INCORRECT WHEN MONITORING A BAD TAPE MARK
2163                                  ;<EXTENDED ACTUAL PRINTOUT = AMTIE LINES
2164  476F  3E   20                 7.0  AMDLPJ: MVI     A,P.LWR         ;CLEAR THE AMTIE PARITY
2165  4771  D3   48                10.0          OUT     PDIAG
2166  4773  3E   58                 7.0          MVI     A, @130         ;SET AMTIES FOR TRACKS 3, 6 AND 9
2167  4775  D3   44                10.0          OUT     TAMT
2168  4777  06   3C                 7.0          MVI     B,60            ;EPRFORM A 300 MICRO SECOND
2169  4779  05                     4.0  1$:     DCR     B               ;TIMEOUT
2170  477A  C2   79   47           10.0          JNZ     1$
2171  477D  DB   02                10.0          IN      RSTAT           ;GET THE READ PATH STATUS
2172  477F  FE   92                 7.0          CPI     GOODTM          ;GOOD STATUS?
2173  4781  CA   91   47           10.0          JZ      AMDCNJ          ;YES-CONTINUE
2174  4784                                       ROUT    ADATA           ;NO-ERROR-STORE ACTUAL STATUS
 (1)  4784  D3   94                10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4786  7F                     4.0                  MOV     A,A             ;RETRY LINK
2175  4787  3E   92                 7.0          MVI     A,GOODTM        ;GET EXPECTED STATUS
2176  4789                                       ROUT    EDATA           ;STORE EXPECTED STATUS
 (1)  4789  D3   95                10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  478B  7F                     4.0                  MOV     A,A             ;RETRY LINK
```

```
2178  478C                                          ERRB    AMDLPJ,AMDCNJ,
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  478C  CD  12  28        18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000D                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  478F  0D                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4790  00                                          .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4791  CD  15  28        18.0              AMDCNJ::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4794  DA  6F  47        10.0                      JC      AMDLPJ          ;LOOP ADDRESS IF LOOP SPECIFIED
2179                                        ;>READ PATH STATUS INCORRECT WHEN MONITORING A GOOD TAPE MARK
2180  4797  2A  FF  4A        16.0  2$:     LHLD    AMTMSK          ;INCREMENT THE AMTIE
2181  479A  23                   6.0          INX     H               ;MASK BY 1
2182  479B  22  FF  4A        16.0          SHLD    AMTMSK          ;SAVE THE MASK
2183  479E  3A  00  4B        13.0          LDA     AMTMSK+1
2184  47A1  FE  01                7.0          CPI     @1
2185  47A3  CA  1D  47        10.0          JZ      AMDLPI
2186  47A6  3A  FF  4A        13.0          LDA     AMTMSK          ;GET LOW ORDER 8 BITS OF AMTIE MASK
2187  47A9  FE  58                7.0          CPI     @130            ;EQUAL A GOOD TAPE MARK
2188  47AB  CA  97  47        10.0          JZ      2$              ;YES-INCREMENT AGAIN
2189  47AE  3A  00  4B        13.0          LDA     AMTMSK+1        ;NO-END OF TEST
2190  47B1  FE  02                7.0          CPI     @2              ;
2191  47B3  C2  1D  47        10.0          JNZ     AMDLPI          ;NO-KEEP ON GOING
2192  47B6                                    ENDTST  TST06X          ;YES-
 (1)                                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  47B6                                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  47B6  CD  06  28        18.0                      CALL    REQST
 (2)  47B9  00                                          .BYTE                   ;DATA PATTERN NUMBER
 (2)  47BA  00  00                                      .WORD                   ;SYSTEM "" COUNT
 (2)  47BC  00  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  47BE  00                                          .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  47BF  07                                          .BYTE   7               ;REQUEST CODE
 (1)  47C0  3A  9A  4F        13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  47C3  3D                   4.0          DCR     A               ;DOWNCOUNT
 (1)  47C4  32  9A  4F        13.0          STA     ITERA           ;SAVE COUNT
 (1)  47C7  F2  0F  47        10.0          JP      TST06X          ;DO TEST UNTIL TILL = 0
2193  47CA  CD  CD  48        18.0          CALL    CLEAR
```

```
2195                            .SBTTL  TEST 07 - 5 X 4 TRANSLATION TEST (FORWARD)
2196  47CD                      ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
2197                            ;*5 X 4 TRANSLATION TEST (FORWARD)
2198  47CD                      SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
2199                            ;*THIS TEST CHECKS THE DATA HANDSHAKE BETWEEN THE ECC AND READ CHANNELS
2200                            ;*AS WELL AS THE READ CHANNELS ABILITY TO 5 TO 4 TRANSLATE ALL POSSIBLE
2201                            ;*COMBINATIONS OF INPUT DATA IN THE FORWARD MODE.
2202  47CD                      SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
2203                            ;*BGNTST
2204                            ;*    SET NORMAL READ PATH CLOCK
2205                            ;*    CALL SUBROUTINE CLEAR
2206                            ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
2207                            ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
2208                            ;*    DELAY 50 MICROSECONDS
2209                            ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
2210                            ;*    CLEAR THE TIE BUS
2211                            ;*    SET UP INPUT TABLE POINTER
2212                            ;*    SET UP OUTPUT TABLE POINTER
2213                            ;*  BGNDO
2214                            ;*  : ENABLE THE READ PATH CLOCK
2215                            ;*  : SET PLO BYPASS MODE 2,TIE BUS JAM AND DISABLE PLO
2216                            ;*  : CLOCK THE FIFO
2217                            ;*  : ISSUE CLEAR ALL COMMAND
2218                            ;*  : WAIT
2219                            ;*  : STOP THE READ PATH
2220                            ;*  : PUT THE CONTROLLER IN GCR MODE
2221                            ;*  : INIT THE SUB-GROUP COUNT TO 2
2222                            ;*  : BGNDO
2223                            ;*  : : INIT THE LOOP COUNT TO 5
2224                            ;*  : : BGNDO
2225                            ;*  : : :    MOVE 8 BITS OF DATA FROM THE INPUT TABLE TO THE TU PORT COMMAND REGISTER
2226                            ;*  : : :    INCREMENT THE INPUT TABLE POINTER
2227                            ;*  : : :    MOVE THE PARITY BIT FROM THE INPUT TABLE TO THE TU PORT CONTROL REGISTER
2228                            ;*  : : :    CLOCK THE DATA INTO THE FIFO
2229                            ;*  : : :    DECREMENT THE LOOP COUNT
2230                            ;*  : : :    INCREMENT THE INPUT TABLE POINTER
2231                            ;*  : : :    DO UNTIL LOOP COUNT = 0
2232                            ;*  : : ENDDO
2233                            ;*  : : DECREMENT THE SUB-GROUP COUNTER
2234                            ;*  : : DO UNTIL THE SUB-GROUP COUNTER -0
2235                            ;*  : ENDDO
2236                            ;*  : ISSUE DIAGNOSTIC READ COMMAND
2237                            ;*  : SET UP WATCHDOG TIMER COUNT
2238                            ;*  : BGNDO
2239                            ;*  : :    SINGLE STEP THE READ PATH
```

```
                                ;*  : :    DECREMENT THE WATCHDOG TIMER
                                ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
                                ;*  : ENDDO
                                ;*  : IF WATCHDOG TIMER=0
                                ;*  : :    THEN-ERROR
                                ;*  : :    ELSE-CONTINUE
                                ;*  : ENDIF
                                ;*  : SINGLE STEP THE READ PATH
                                ;*  : INIT THE LOOP COUNT TO 4
                                ;*  : BGNDO
                                ;*  : :    SINGLE STEP THE READ PATH
                                ;*  : :    INPUT THE ACTUAL DATA
                                ;*  : :    INPUT THE ACTUAL PARITY
                                ;*  : :    GET THE EXPECTED DATA FROM THE EXPECTED DATA TABLE
                                ;*  : :    INCREMENT THE OUTPUT TABLE POINTER
                                ;*  : :    GET THE EXPECTED PARITY FROM THE EXPECTED DATA TABLE
                                ;*  : :    INCREMENT THE OUTPUT TABLE POINTER
                                ;*  : :    CALL SUBROUTINE "CK5X4"
                                ;*  : :    DECREMENT THE LOOP COUNT
                                ;*  : :    DO UNTIL LOOP COUNT = 0
                                ;*  : ENDDO
                                ;*  : DO UNTIL END OF INPUT DATA TABLE
                                ;*  ENDDO
                                ;*ENDTST
2264  47CD                      SE
 (1)                            ; *****************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
2265                            ;*RPM3 MICRO TEST 07
2266                            ;*RPM3 MICRO ERROR 16
2267                            ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
2268                            ;*M8950'S, M8953
2269                            ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
2270                            ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
2271                            ;*FATAL ERROR - MICRO TEST ABORTED
2272                            ;*
2273                            ;*RPM3 MICRO TEST 07
2274                            ;*RPM3 MICRO ERROR 17
2275                            ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
2276                            ;*M8950'S, M8953
2277                            ;*CHANNEL 0 FAILED
2278                            ;*ACTUAL = NNNN
2279                            ;*EXPECTED = NNNN
2280                            ;*SUBGROUP NUMBER = LLL
2281                            ;*
2282                            ;*RPM3 MICRO TEST 07
2283                            ;*RPM3 MICRO ERROR 20
2284                            ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
2285                            ;*M8950'S, M8953
2286                            ;*CHANNEL 1 FAILED
2287                            ;*ACTUAL = NNNN
2288                            ;*EXPECTED = NNNN
2289                            ;*SUBGROUP NUMBER - LLL
2290                            ;*
```

```
                                ;*RPM3 MICRO TEST 07
                                ;*RPM3 MICRO ERROR 21
                                ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
                                ;*M8950'S, M8953
                                ;*CHANNEL 2 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*SUBGROUP NUMBER = LLL
                                ;*
                                ;*RPM3 MICRO TEST 07
                                ;*RPM3 MICRO ERROR 22
                                ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
                                ;*M8950'S, M8953
                                ;*CHANNEL 3 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*SUBGROUP NUMBER = LLL
                                ;*
                                ;*RPM3 MICRO TEST 07
                                ;*RPM3 MICRO ERROR 23
                                ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
                                ;*M8950'S, M8953
                                ;*CHANNEL 4 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*SUBGROUP NUMBER = LLL
                                ;*
                                ;*RPM3 MICRO TEST 07
                                ;*RPM3 MICRO ERROR 24
                                ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
                                ;*M8950'S, M8953
                                ;*CHANNEL 5 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*SUBGROUP NUMBER = LLL
                                ;*
                                ;*RPM3 MICRO TEST 07
                                ;*RPM3 MICRO ERROR 25
                                ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
                                ;*M8950'S, M8953
                                ;*CHANNEL 6 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*SUBGROUP NUMBER = LLL
                                ;*
                                ;*RPM3 MICRO TEST 07
                                ;*RPM3 MICRO ERROR 26
                                ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
                                ;*M8950'S, M8953
                                ;*CHANNEL 7 FAILED
                                ;*ACTUAL = NNNN
                                ;*EXPECTED - NNNN
                                ;*SUBGROUP NUMBER - LLL
                                ;*
```

```
                                        ;*RPM3 MICRO TEST 07
                                        ;*RPM3 MICRO ERROR 27
                                        ;*RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
                                        ;*M8950'S, M8953
                                        ;*CHANNEL P FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*SUBGROUP NUMBER = LLL
47CD                                    $
 (1)                                    ; ************************************************************************************

47CD                                    TEST7:  TESTX   @7              ;INITIALIZE THE TEST
 (1) 47CD  3E  07            7.0                        MVI     A,@7        ;DEFINE THE TEST NUMBER
 (1) 47CF  CD  03  28       18.0                        CALL    TSET        ;SETUP THE TEST
                                        ;%RPM3-READ PATH-5X4 TRANSLATION TEST (FORWARD)
                                        ;&M8950'S, M8953

47D2  3E  10                7.0 TST07X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
47D4  D3  F0               10.0         OUT     CLKCTL
47D6  CD  CD  48           18.0         CALL    CLEAR           ;CLEAR ALL TU PORTS
47D9  3A  FE  4A           13.0         LDA     SUNIT
47DC  D3  E0               10.0         OUT     MBSEL
47DE  3E  04                7.0         MVI     A,P.RPEN
47E0  D3  4C               10.0         OUT     PENAB
47E2  3E  60                7.0         MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
47E4  D3  48               10.0         OUT     PDIAG           ;LCS, LWR AND DPEN
47E6  AF                    4.0         XRA     A
47E7  D3  44               10.0         OUT     TAMT
47E9  3E  0A                7.0         MVI     A,10            ;DELAY
47EB  3D                    4.0 1$:     DCR     A               ;50
47EC  C2  EB  47           10.0         JNZ     1$              ;MICROSECONDS
47EF  D3  D2               10.0         OUT     TRKENA          ;CLEAR ALL TRACKS FROM
47F1  D3  D2               10.0         OUT     TRKENA          ;THE TRANSLATOR

47F3  AF                    4.0         XRA     A               ;CLEAR THE TIE BUS
47F4  D3  0A               10.0         OUT     RTIEB
47F6  3E  01                7.0         MVI     A,@1            ;INITIALIZE THE SUBGROUP COUNT
47F8  32  0B  4B           13.0         STA     SUBCNT          ;

47FB  21  DD  49           10.0         LXI     H,M8950I        ;SET UP POINTER TO INPUT DATA TABLE
47FE  22  03  4B           16.0         SHLD    INTBL           ;SAVE IT
4801  21  7D  4A           10.0         LXI     H,M8950O        ;SET UP POINTER TO OUTPUT DATA TABLE
4804  22  05  4B           16.0         SHLD    OUTTBL          ;SAVE IT
4807  3E  A8                7.0 TST07C: MVI     A,R PLO1'R.PLOD.R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
                                                                ;DISABLE THE PLO AND SET JAM TIE BUS
4809  D3  09               10.0         OUT     RPCTL
480B  3E  10                7.0         MVI     A,W.GCR         ;SET GCR MODE
480D  D3  D3               10.0         OUT     WMCCTL
480F  D3  08               10.0         OUT     RFIFOL          ;CLOCK THE FIFO
4811  3E  0D                7.0         MVI     A,RCLRT         ;ISSUE CLEAR ALL COMMAND
4813  D3  0B               10.0         OUT     RCMD
4815  00                    4.0         NOP                     ;WAIT
4816  00                    4.0         NOP
4817  00                    4.0         NOP
```

```
2396  4818  00              4.0          NOP
2397  4819  3E  A9          7.0          MVI     A,R.PLO1!R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
2398  481B  D3  09         10.0          OUT     RPCTL
2399
2400  481D  0E  02          7.0          MVI     C,2             ;INITIALIZE THE SUB-GROUP COUNTER
2401  481F  2A  03  4B     16.0  3$:     LHLD    INTBL           ;GET POINTER TO INPUT DATA TABLE
2402  4822  06  05          7.0          MVI     B,@5            ;SET UP LOOP COUNT
2403  4824  7E              7.0  1$:     MOV     A,M             ;GET A DATA BYTE
2404  4825  D3  40         10.0          OUT     TCMD            ;STORE DATA IN COMMAND ADDRESS
2405  4827  23              6.0          INX     H               ;POINT TO DATA PARITY
2406  4828  7E              7.0          MOV     A,M             ;GET THE DATA PARITY
2407  4829  07              4.0          RLC                     ;POSITION FOR OUTPUT
2408  482A  F6  60          7.0          ORI     P.LWR!P.LCS     ;OR IN CONTROL BITS
2409  482C  D3  48         10.0          OUT     PDIAG           ;OUTPUT THE DATA PARITY
2410  482E  D3  08         10.0          OUT     RFIFOL          ;CLOCK DATA INTO THE FIFO'S
2411
2412  4830  23              6.0          INX     H               ;UPDATE THE TABLE POINTER
2413  4831  05              4.0          DCR     B               ;DECREMENT LOOP COUNT
2414  4832  C2  24  48      10.0          JNZ     1$              ;DO UNTIL LOOP COUNT = 0
2415  4835  0D              4.0          DCR     C               ;DECREMENT THE SUBGROUP COUNTER
2416  4836  C2  1F  48      10.0          JNZ     3$              ;LOAD THE SUB-GROUP TWICE
2417                                ;THE FIFO IS NOW FILLED WITH THE INPUT DATA TABLE
2418
2419  4839  22  01  4B     16.0          SHLD    INTBLTMP        ;SAVE THE INPUT DATA TABLE POINTER
2420  483C  3E  0B          7.0          MVI     A,DIARD         ;LOAD THE DIAGNOSTIC READ
2421  483E  D3  0B         10.0          OUT     RCMD            ;COMMAND
2422  4840  11  01  00     10.0          LXI     D,1             ;SET WATCH DOG INCREMENT
2423  4843  21  A8  FD     10.0          LXI     H,-600          ;SET WATCH DOG COUNT TO 600
2424  4846  D3  0C         10.0  2$:     OUT     RINST           ;STEP THE READ PATH
2425  4848  DB  01         10.0          IN      RPCHI           ;DATA READY SET?
2426  484A  E6  10          7.0          ANI     R.DRDY
2427  484C  C2  61  48      10.0          JNZ     TST07A          ;YES-GO PROCESS
2428  484F  19              10.0          DAD     D               ;WATCH DOG TIMEOUT?
2429  4850  D2  46  48      10.0          JNC     2$              ;NO-CONTINUE
2430  4853                               ERR     EXIT,DUMMX
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4853  CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000E                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4856  0E                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4857  00                                      .BYTE
 (1)  4858  CD  15  28        18.0          DUMMX:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  485B  DA  CA  48        10.0                  JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
2431                          ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
2432                          ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
2433                          ;<FATAL ERROR - MICRO TEST ABORTED
2434  485E  C3  CA  48        10.0          JMP     EXIT
2435
2436  4861  2A  05  4B     16.0  TST07A:  LHLD    OUTTBL          ;GET ADDRESS OF EXPECTED OUTPUT DATA TABLE
2437  4864  00              4.0          NOP
2438  4865  00              4.0          NOP
2439  4866  00              4.0          NOP
2440  4867  0E  04          7.0          MVI     C,@4            ;INIT THE LOOP COUNT
2441  4869  00              4.0  TST07B:  NOP
2442  486A  D3  0C         10.0          OUT     RINST           ;SINGLE STEP THE READ PATH
```

```
2443  486C  DB  17        10.0        IN      RDATA           ;INPUT THE ACTUAL DATA BYTE
2444  486E  32  07  4B    13.0        STA     DATAA           ;STORE IT
2445  4871  DB  15        10.0        IN      RPSTA           ;INPUT THE ACTUAL PARITY BIT
2446  4873  E6  40         7.0        ANI     R.DATA          ;REMOVE JUNK BITS
2447  4875  07             4.0        RLC                     ;POSITION THE BIT FOR COMPARE
2448  4876  07             4.0        RLC
2449  4877  32  08  4B    13.0        STA     DATAAP          ;STORE IT
2450  487A  7E              7.0        MOV     A,M             ;GET THE EXPECTED DATA BYTE
2451  487B  32  09  4B    13.0        STA     DATAE           ;STORE IT
2452  487E  23              6.0        INX     H               ;UPDATE THE POINTER
2453  487F  7E              7.0        MOV     A,M             ;GET THE EXPECTED PARITY BYTE
2454  4880  32  0A  4B    13.0        STA     DATAEP          ;STORE IT
2455  4883  EB              4.0        XCHG                    ;SAVE H AND L IN D AND E
2456  4884  3A  0B  4B    13.0        LDA     SUBCNT          ;GET THE SUBCROUP COUNT
2457  4887                             ROUT    R05L            ;STORE IN CAS
 (1)  4887  D3  8A         10.0                OUT     R05L            ;WRITE AC INTO R05L
 (1)  4889  7F              4.0                MOV     A,A             ;RETRY LINK
2458  488A  AF              4.0        XRA     A               ;CLEAR THE HIGH HALF OF THE SUBGROUP COUNTER
2459  488B                             ROUT    R05H            ;
 (1)  488B  D3  8B         10.0                OUT     R05H            ;WRITE AC INTO R05H
 (1)  488D  7F              4.0                MOV     A,A             ;RETRY LINK
2460  488E  CD  F8  48    18.0        CALL    CK5X4           ;GO CHECK FOR FAILING READ CHANNELS
2461  4891  DA  07  48    10.0        JC      TST07C          ;IF LOOP ON ERROR REPEAT THE TEST
2462  4894  EB              4.0        XCHG                    ;RESTORE H AND L FROM D AND E
2463  4895  23              6.0        INX     H               ;UPDATE THE EXPECTED DATA TABLE POINTER
2464  4896  0D              4.0        DCR     C               ;DECREMENT THE LOOP COUNT
2465  4897  C2  69  48    10.0        JNZ     TST07B          ;DO UNTIL LOOP COUNT = ZERO
2466  489A  22  05  4B    16.0        SHLD    OUTTBL          ;SAVE EXPECTED DATA TABLE POINTER
2467  489D  3A  0B  4B    13.0        LDA     SUBCNT          ;GET THE SUBGROUP COUNTER
2468  48A0  3C              4.0        INR     A               ;UPDATE FOR THE NEXT SUBGROUP
2469  48A1  32  0B  4B    13.0        STA     SUBCNT          ;SAVE IT
2470  48A4  2A  01  4B    16.0        LHLD    INTBLTMP        ;GET THE INPUT TABLE POINTER
2471  48A7  22  03  4B    16.0        SHLD    INTBL           ;UPDATE THE INPUT TABLE POINTER
2472  48AA  2B              6.0        DCX     H               ;UPDATE TO LAST PARITY BYTE
2473  48AB  7E              7.0        MOV     A,M             ;GET LAST PARITY BYTE
2474  48AC  A7              4.0        ANA     A               ;SET CONDITION BITS
2475  48AD  F2  07  48    10.0        JP      TST07C          ;CONTINUE UNTIL END OF INPUT TABLE
2476  48B0  CD  CD  48    18.0        CALL    CLEAR           ;CLEAR ALL TU PORTS
2477  48B3                             ENDTST  TST07X
 (1)                       ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  48B3                           REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  48B3  CD  06  28    18.0                        CALL    REQST
 (2)  48B6  00                                        .BYTE                   ;DATA PATTERN NUMBER
 (2)  48B7  00  00                                    .WORD                   ;SYSTEM "" COUNT
 (2)  48B9  00  00                                    .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  48BB  00                                        .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  48BC  07                                        .BYTE   7               ;REQUEST CODE
 (1)  48BD  3A  9A  4F    13.0        LDA     ITERA           ;GET ITERATION COUNT
 (1)  48C0  3D              4.0        DCR     A               ;DOWNCOUNT
 (1)  48C1  32  9A  4F    13.0        STA     ITERA           ;SAVE COUNT
 (1)  48C4  F2  D2  47    10.0        JP      TST07X          ;DO TEST UNTIL TILL = 0
2478  48C7  CD  CD  48    18.0        CALL    CLEAR
2479
2480  48CA  C3  18  28    10.0  EXIT: JMP     TSTEND
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
48CD                                    $SUB
 (1)                                    ; ***************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
48CD                                    $D
 (1)                                    ; ***************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
48CD                                    $P
 (1)                                    ; ***************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNSUB
                                        ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*    CLEAR PORT SELECT FOR TRANSPORT
                                        ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*    CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
48CD                                    $
 (1)                                    ; ***************************************************************************
48CD F5              12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
48CE C5              12.0          PUSH    B
48CF 06 00            7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
48D1 DB E0           10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
48D3 E6 80            7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
48D5 B0               4.0          ORA     B               ;ADD IN THE SELECTED PORT #
48D6 D3 E0           10.0          OUT     MBSEL           ;RESET TO THIS PORT
48D8 3E 80            7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
48DA D3 40           10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
48DC AF               4.0          XRA     A               ;CLEAR TU COMMAND A
48DD D3 40           10.0          OUT     TCMD            ;
48DF 3E 81            7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
48E1 D3 40           10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
48F3 3E 00            7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
48E5 D3 40           10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
48E7 AF               4.0          XRA     A
48E8 D3 44           10.0          OUT     TAMT            ;CLEAR AMTIE WORD
48EA D3 48           10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
48EC D3 4C           10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
48EE 04               4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
48EF 78               4.0          MOV     A,B
48F0 FE 04            7.0          CPI     4               ;DONE?
48F2 C2 D1 48        10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
48F5 C1              10.0          POP     B               ;RESET B & C
48F6 F1              10.0          POP     PSW             ;ALL DONE
48F7 C9              10.0          RET                     ;EXIT
```

```
2526                                .SBTTL  SUBROUTINE CHECK 5X4
2527  48F8                          $SUB
 (1)                                ;*************************************************************************
 (1)                                ;*SUBROUTINE TITLE
 (1)                                ;*---------- -----
2528                                ;*CHECK 5X4 SUBROUTINE
2529  48F8                          $D
 (1)                                ;*************************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
2530                                ;*THIS SUBROUTINE IS USED TO ISOLATE A BIT FAILURE IN THE READ CHANNEL DATA
2531                                ;*OR AMTIE BYTES TO A SPECIFIC READ CHANNEL BY COMPARING THE ACTUAL AND
2532                                ;*EXPECTED INPUTS TO THE SUBROUTINE ON A BIT BY BIT BASIS.
2533  48F8                          $P
 (1)                                ;*************************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
2534                                ;*BGNSUB
2535                                ;*     IF CHANNEL 0 ACTUAL INFORMATION NOT = CHANNEL 0 EXPECTED
2536                                ;*       THEN-ERROR CHANNEL 0
2537                                ;*       ELSE-CONTINUE
2538                                ;*     ENDIF
2539                                ;*     IF CHANNEL 1 ACTUAL INFORMATION NOT = CHANNEL 1 EXPECTED
2540                                ;*       THEN-ERROR CHANNEL 1
2541                                ;*       ELSE-CONTINUE
2542                                ;*     ENDIF
2543                                ;*     IF CHANNEL 2 ACTUAL INFORMATION NOT = CHANNEL 2 EXPECTED
2544                                ;*       THEN-ERROR CHANNEL 2
2545                                ;*       ELSE-CONTINUE
2546                                ;*     ENDIF
2547                                ;*     IF CHANNEL 3 ACTUAL INFORMATION NOT = CHANNEL 3 EXPECTED
2548                                ;*       THEN-ERROR CHANNEL 3
2549                                ;*       ELSE-CONTINUE
2550                                ;*     ENDIF
2551                                ;*     IF CHANNEL 4 ACTUAL INFORMATION NOT = CHANNEL 4 EXPECTED
2552                                ;*       THEN-ERROR CHANNEL 4
2553                                ;*       ELSE-CONTINUE
2554                                ;*     ENDIF
2555                                ;*     IF CHANNEL 5 ACTUAL INFORMATION NOT = CHANNEL 5 EXPECTED
2556                                ;*       THEN-ERROR CHANNEL 5
2557                                ;*       ELSE-CONTINUE
2558                                ;*     ENDIF
2559                                ;*     IF CHANNEL 6 ACTUAL INFORMATION NOT = CHANNEL 6 EXPECTED
2560                                ;*       THEN-ERROR CHANNEL 6
2561                                ;*       ELSE-CONTINUE
2562                                ;*     ENDIF
2563                                ;*     IF CHANNEL 7 ACTUAL INFORMATION NOT = CHANNEL 7 EXPECTED
2564                                ;*       THEN-ERROR CHANNEL 7
2565                                ;*       ELSE-CONTINUE
2566                                ;*     ENDIF
2567                                ;*     IF CHANNEL P ACTUAL INFORMATION NOT = CHANNEL P EXPECTED
2568                                ;*       THEN-ERROR CHANNEL P
2569                                ;*       ELSE-CONTINUE
2570                                ;*     ENDIF
```

```
2571                                        ;*ENDSUB
2572  48F8                                  $
 (1)                                        ; ************************************************************************
2573  48F8  21  09  4B      10.0  CK5X4:  LXI   H,DATAE          ;LOAD EXPECTED DATA ADDRESS
2574  48FB  3A  07  4B      13.0          LDA   DATAA            ;GET THE ACTUAL DATA
2575  48FE                       ROUT    ADATA                  ;WRITE THE ACTUAL DATA TO THE CAS
 (1)  48FE  D3  94            10.0                OUT   ADATA            ;WRITE AC INTO ADATA
 (1)  4900  7F                 4.0                 MOV   A,A              ;RETRY LINK
2576  4901  E6  01            7.0           ANI   $01              ;REMOVE UNWANTED BITS
2577  4903  47                 4.0           MOV   B,A              ;SAVE IN B
2578  4904  7E                 7.0           MOV   A,M              ;GET EXPECTED DATA
2579  4905                       ROUT    EDATA                  ;WRITE THE EXPECTED DATA TO THE CAS
 (1)  4905  D3  95            10.0                OUT   EDATA            ;WRITE AC INTO EDATA
 (1)  4907  7F                 4.0                 MOV   A,A              ;RETRY LINK
2580  4908  E6  01            7.0           ANI   $01              ;REMOVE UNWANTED BITS
2581  490A  B8                 4.0           CMP   B                ;COMPARE
2582  490B  CA  13  49        10.0           JZ    CKAM0            ;CONTINUE IF EQUAL
2583  490E                       ERRB    OUT,CKAM0,@7
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  490E  CD  12  28        18.0                CALL  ERLPB            ;PROCESS ERROR - DO 2.3
 (1)      000F                                    MSGN  =     MSGN+1             ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4911  0F                                    .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  4912  07                                    .BYTE @7               ;PRINT ROUTINE NUMBER
 (1)  4913  CD  15  28        18.0          CKAM0:: CALL  CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4916  DA  DC  49        10.0                JC    OUT              ;LOOP ADDRESS IF LOOP SPECIFIED
2584                                        ;>CHANNEL 0 FAILED
2585
2586  4919  3A  07  4B      13.0          LDA   DATAA            ;GET THE ACTUAL DATA
2587  491C  E6  02            7.0           ANI   $02              ;REMOVE UNWANTED BITS
2588  491E  47                 4.0           MOV   B,A              ;SAVE IN B
2589  491F  7E                 7.0           MOV   A,M              ;GET FXPECTED DATA
2590  4920  E6  02            7.0           ANI   $02              ;REMOVE UNWANTED BITS
2591  4922  B8                 4.0           CMP   B                ;COMPARE
2592  4923  CA  2B  49        10.0           JZ    CKAM1            ;CONTINUE IF EQUAL
2593  4926                       ERRB    OUT,CKAM1,@7
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4926  CD  12  28        18.0                CALL  ERLPB            ;PROCESS ERROR - DO 2.3
 (1)      0010                                    MSGN  =     MSGN+1             ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4929  10                                    .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  492A  07                                    .BYTE @7               ;PRINT ROUTINE NUMBER
 (1)  492B  CD  15  28        18.0          CKAM1:: CALL  CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  492E  DA  DC  49        10.0                JC    OUT              ;LOOP ADDRESS IF LOOP SPECIFIED
2594                                        ;>CHANNEL 1 FAILED
```

```
2596  4931  3A  07  4B  13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2597  4934  E6  04      7.0          ANI     $04             ;REMOVE UNWANTED BITS
2598  4936  47          4.0          MOV     B,A             ;SAVE IN B
2599  4937  7E          7.0          MOV     A,M             ;GET EXPECTED DATA
2600  4938  E6  04      7.0          ANI     $04             ;REMOVE UNWANTED BITS
2601  493A  B8          4.0          CMP     B               ;COMPARE
2602  493B  CA  43  49  10.0          JZ      CKAM2           ;CONTINUE IF EQUAL
2603  493E                           ERRB    OUT,CKAM2,@7
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  493E  CD  12  28  18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0011                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4941  11                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4942  07                                 .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  4943  CD  15  28  18.0            CKAM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4946  DA  DC  49  10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2604                           ;>CHANNEL 2 FAILED
2605
2606  4949  3A  07  4B  13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2607  494C  E6  08      7.0          ANI     $08             ;REMOVE UNWANTED BITS
2608  494E  47          4.0          MOV     B,A             ;SAVE IN B
2609  494F  7E          7.0          MOV     A,M             ;GET EXPECTED DATA
2610  4950  E6  08      7.0          ANI     $08             ;REMOVE UNWANTED BITS
2611  4952  B8          4.0          CMP     B               ;COMPARE
2612  4953  CA  5B  49  10.0          JZ      CKAM3           ;CONTINUE IF EQUAL
2613  4956                           ERRB    OUT,CKAM3,@7
 (1)                                 _AG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4956  CD  12  28  18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0012                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4959  12                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  495A  07                                 .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  495B  CD  15  28  18.0            CKAM3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  495E  DA  DC  49  10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2614                           ;>CHANNEL 3 FAILED
2615
2616  4961  3A  07  4B  13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2617  4964  E6  10      7.0          ANI     $10             ;REMOVE UNWANTED BITS
2618  4966  47          4.0          MOV     B,A             ;SAVE IN B
2619  4967  7E          7.0          MOV     A,M             ;GET EXPECTED DATA
2620  4968  E6  10      7.0          ANI     $10             ;REMOVE UNWANTED BITS
2621  496A  B8          4.0          CMP     B               ;COMPARE
2622  496B  CA  73  49  10.0          JZ      CKAM4           ;CONTINUE IF EQUAL
2623  496E                           ERRB    OUT,CKAM4,@7
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  496E  CD  12  28  18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0013                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4971  13                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4972  07                                 .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  4973  CD  15  28  18.0            CKAM4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4976  DA  DC  49  10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2624                           ;>CHANNEL 4 FAILED
```

```
2626  4979  3A 07 4B      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2627  497C  E6 20           7.0          ANI     $20             ;REMOVE UNWANTED BITS
2628  497E  47              4.0          MOV     B,A             ;SAVE IN B
2629  497F  7E              7.0          MOV     A,M             ;GET EXPECTED DATA
2630  4980  E6 20           7.0          ANI     $20             ;REMOVE UNWANTED BITS
2631  4982  B8              4.0          CMP     B               ;COMPARE
2632  4983  CA 8B 49       10.0          JZ      CKAM5           ;CONTINUE IF EQUAL
2633  4986                                ERRB    OUT,CKAM5,@7
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4986  CD 12 28       18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0014                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4989  14                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  498A  07                                     .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  498B  CD 15 28       18.0            CKAM5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  498E  DA DC 49       10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2634                                  ;>CHANNEL 5 FAILED
2635
2636  4991  3A 07 4B      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2637  4994  E6 40           7.0          ANI     $40             ;REMOVE UNWANTED BITS
2638  4996  47              4.0          MOV     B,A             ;SAVE IN B
2639  4997  7E              7.0          MOV     A,M             ;GET EXPECTED DATA
2640  4998  E6 40           7.0          ANI     $40             ;REMOVE UNWANTED BITS
2641  499A  B8              4.0          CMP     B               ;COMPARE
2642  499B  CA A3 49       10.0          JZ      CKAM6           ;CONTINUE IF EQUAL
2643  499E                                ERRB    OUT,CKAM6,@7
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  499E  CD 12 28       18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0015                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  49A1  15                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  49A2  07                                     .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  49A3  CD 15 28       18.0            CKAM6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  49A6  DA DC 49       10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2644                                  ;>CHANNEL 6 FAILED
2645
2646  49A9  3A 07 4B      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2647  49AC  E6 80           7.0          ANI     $80             ;REMOVE UNWANTED BITS
2648  49AE  47              4.0          MOV     B,A             ;SAVE IN B
2649  49AF  7E              7.0          MOV     A,M             ;GET EXPECTED DATA
2650  49B0  E6 80           7.0          ANI     $80             ;REMOVE UNWANTED BITS
2651  49B2  B8              4.0          CMP     B               ;COMPARE
2652  49B3  CA BB 49       10.0          JZ      CKAM7           ;CONTINUE IF EQUAL
2653  49B6                                ERRB    OUT,CKAM7,@7
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  49B6  CD 12 28       18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0016                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  49B9  16                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  49BA  07                                     .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  49BB  CD 15 28       18.0            CKAM7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  49BE  DA DC 49       10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2654                                  ;>CHANNEL 7 FAILED
```

```
2656  49C1  21  0A  4B  10.0          LXI   H,DATAEP          ;GET THE EXPECTED DATA PARITY ADDRESS
2657  49C4  3A  08  4B  13.0          LDA   DATAAP            ;GET THE ACTUAL DATA
2658  49C7  E6  01       7.0          ANI   $01               ;REMOVE UNWANTED BITS
2659  49C9  47           4.0          MOV   B,A               ;SAVE IN B
2660  49CA  7E           7.0          MOV   A,M               ;GET EXPECTED DATA
2661  49CB  E6  01       7.0          ANI   $01               ;REMOVE UNWANTED BITS
2662  49CD  B8           4.0          CMP   B                 ;COMPARE
2663  49CE  CA  D6  49   10.0          JZ    CKAMP             ;CONTINUE IF EQUAL
2664  49D1                           ERRB  OUT,CKAMP,@7
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  49D1  CD  12  28  18.0                    CALL  ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       00.7                                MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  49D4  17                                  .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  49D5  07                                  .BYTE @7              ;PRINT ROUTINE NUMBER
 (1)  49D6  CD  15  28  18.0              CKAMP:: CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  49D9  DA  DC  49  10.0                    JC    OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2665                          ;>CHANNEL P FAILED
2666
2667  49DC  C9           10.0  OUT:    RET
```

```
                                .SBTTL  TABLE UNTRANSLATED INPUT DATA

                                ;INPUT DATA TABLE - THIS DATA IS LOADED INTO THE READ CHANNEL FIFO'S
                                ;ONE GROUP AT A TIME AND THEN TRANSLATED.

49DD  FF        M8950I: .BYTE   @377            ;DATA GROUP 1 - CHARACTER 1
49DE  01                .BYTE   @001            ;DATA GROUP 1 - CHARACTER 1  PARITY
49DF  FF                .BYTE   @377            ;DATA GROUP 1 - CHARACTER 2
49E0  01                .BYTE   @001            ;DATA GROUP 1 - CHARACTER 2  PARITY
49E1  00                .BYTE   @000            ;DATA GROUP 1 - CHARACTER 3
49E2  00                .BYTE   @000            ;DATA GROUP 1 - CHARACTER 3  PARITY
49E3  00                .BYTE   @000            ;DATA GROUP 1 - CHARACTER 4
49E4  00                .BYTE   @000            ;DATA GROUP 1 - CHARACTER 4  PARITY
49E5  FF                .BYTE   @377            ;DATA GROUP 1 - CHARACTER 5
49E6  01                .BYTE   @001            ;DATA GROUP 1 - CHARACTER 5  PARITY

49E7  FF                .BYTE   @377            ;DATA GROUP 2 - CHARACTER 1
49E8  01                .BYTE   @001            ;DATA GROUP 2 - CHARACTER 1  PARITY
49E9  FF                .BYTE   @377            ;DATA GROUP 2 - CHARACTER 2
49EA  01                .BYTE   @001            ;DATA GROUP 2 - CHARACTER 2  PARITY
49EB  00                .BYTE   @000            ;DATA GROUP 2 - CHARACTER 3
49EC  00                .BYTE   @000            ;DATA GROUP 2 - CHARACTER 3  PARITY
49ED  FF                .BYTE   @377            ;DATA GROUP 2 - CHARACTER 4
49EE  01                .BYTE   @001            ;DATA GROUP 2 - CHARACTER 4  PARITY
49EF  FF                .BYTE   @377            ;DATA GROUP 2 - CHARACTER 5
49F0  01                .BYTE   @001            ;DATA GROUP 2 - CHARACTER 5  PARITY

49F1  FF                .BYTE   @377            ;DATA GROUP 3 - CHARACTER 1
49F2  01                .BYTE   @001            ;DATA GROUP 3 - CHARACTER 1  PARITY
49F3  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 2
49F4  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 2  PARITY
49F5  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 3
49F6  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 3  PARITY
49F7  FF                .BYTE   @377            ;DATA GROUP 3 - CHARACTER 4
49F8  01                .BYTE   @001            ;DATA GROUP 3 - CHARACTER 4  PARITY
49F9  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 5
49FA  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 5  PARITY

49FB  FF                .BYTE   @377            ;DATA GROUP 4 - CHARACTER 1
49FC  01                .BYTE   @001            ;DATA GROUP 4 - CHARACTER 1  PARITY
49FD  00                .BYTE   @000            ;DATA GROUP 4 - CHARACTER 2
49FE  00                .BYTE   @000            ;DATA GROUP 4 - CHARACTER 2  PARITY
49FF  00                .BYTE   @000            ;DATA GROUP 4 - CHARACTER 3
4A00  00                .BYTE   @000            ;DATA GROUP 4 - CHARACTER 3  PARITY
4A01  FF                .BYTE   @377            ;DATA GROUP 4 - CHARACTER 4
4A02  01                .BYTE   @001            ;DATA GROUP 4 - CHARACTER 4  PARITY
4A03  FF                .BYTE   @377            ;DATA GROUP 4 - CHARACTER 5
4A04  01                .BYTE   @001            ;DATA GROUP 4 - CHARACTER 5  PARITY

4A05  FF                .BYTE   @377            ;DATA GROUP 5 - CHARACTER 1
4A06  01                .BYTE   @001            ;DATA GROUP 5 - CHARACTER 1  PARITY
4A07  FF                .BYTE   @377            ;DATA GROUP 5 - CHARACTER 2
4A08  01                .BYTE   @001            ;DATA GROUP 5 - CHARACTER 2  PARITY
4A09  FF                .BYTE   @377            ;DATA GROUP 5 - CHARACTER 3
```

```
2723  4A0A  01          .BYTE   @001          ;DATA GROUP 5 - CHARACTER 3  PARITY
2724  4A0B  00          .BYTE   @000          ;DATA GROUP 5 - CHARACTER 4
2725  4A0C  00          .BYTE   @000          ;DATA GROUP 5 - CHARACTER 4  PARITY
2726  4A0D  FF          .BYTE   @377          ;DATA GROUP 5 - CHARACTER 5
2727  4A0E  01          .BYTE   @001          ;DATA GROUP 5 - CHARACTER 5  PARITY
2728
2729  4A0F  FF          .BYTE   @377          ;DATA GROUP 6 - CHARACTER 1
2730  4A10  01          .BYTE   @001          ;DATA GROUP 6 - CHARACTER 1  PARITY
2731  4A11  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 2
2732  4A12  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 2  PARITY
2733  4A13  FF          .BYTE   @377          ;DATA GROUP 6 - CHARACTER 3
2734  4A14  01          .BYTE   @001          ;DATA GROUP 6 - CHARACTER 3  PARITY
2735  4A15  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 4
2736  4A16  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 4  PARITY
2737  4A17  FF          .BYTE   @377          ;DATA GROUP 6 - CHARACTER 5
2738  4A18  01          .BYTE   @001          ;DATA GROUP 6 - CHARACTER 5  PARITY
2739
2740  4A19  FF          .BYTE   @377          ;DATA GROUP 7 - CHARACTER 1
2741  4A1A  01          .BYTE   @001          ;DATA GROUP 7 - CHARACTER 1  PARITY
2742  4A1B  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 2
2743  4A1C  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 2  PARITY
2744  4A1D  FF          .BYTE   @377          ;DATA GROUP 7 - CHARACTER 3
2745  4A1E  01          .BYTE   @001          ;DATA GROUP 7 - CHARACTER 3  PARITY
2746  4A1F  FF          .BYTE   @377          ;DATA GROUP 7 - CHARACTER 4
2747  4A20  01          .BYTE   @001          ;DATA GROUP 7 - CHARACTER 4  PARITY
2748  4A21  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 5
2749  4A22  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 5  PARITY
2750
2751  4A23  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 1
2752  4A24  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 1  PARITY
2753  4A25  00          .BYTE   @000          ;DATA GROUP 8 - CHARACTER 2
2754  4A26  00          .BYTE   @000          ;DATA GROUP 8 - CHARACTER 2  PARITY
2755  4A27  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 3
2756  4A28  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 3  PARITY
2757  4A29  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 4
2758  4A2A  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 4  PARITY
2759  4A2B  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 5
2760  4A2C  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 5  PARITY
2761
2762  4A2D  FF          .BYTE   @377          ;DATA GROUP 9 - CHARACTER 1
2763  4A2E  01          .BYTE   @001          ;DATA GROUP 9 - CHARACTER 1  PARITY
2764  4A2F  FF          .BYTE   @377          ;DATA GROUP 9 - CHARACTER 2
2765  4A30  01          .BYTE   @001          ;DATA GROUP 9 - CHARACTER 2  PARITY
2766  4A31  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 3
2767  4A32  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 3  PARITY
2768  4A33  FF          .BYTE   @377          ;DATA GROUP 9 - CHARACTER 4
2769  4A34  01          .BYTE   @001          ;DATA GROUP 9 - CHARACTER 4  PARITY
2770  4A35  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 5
2771  4A36  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 5  PARITY
2772
2773  4A37  00          .BYTE   @000          ;DATA GROUP 10 - CHARACTER 1
2774  4A38  00          .BYTE   @000          ;DATA GROUP 10 - CHARACTER 1  PARITY
2775  4A39  FF          .BYTE   @377          ;DATA GROUP 10 - CHARACTER 2
2776  4A3A  01          .BYTE   @001          ;DATA GROUP 10 - CHARACTER 2  PARITY
```

```
2777  4A3B  00          .BYTE  @000    ;DATA GROUP 10 - CHARACTER 3
2778  4A3C  00          .BYTE  @000    ;DATA GROUP 10 - CHARACTER 3  PARITY
2779  4A3D  00          .BYTE  @000    ;DATA GROUP 10 - CHARACTER 4
2780  4A3E  00          .BYTE  @000    ;DATA GROUP 10 - CHARACTER 4  PARITY
2781  4A3F  FF          .BYTE  @377    ;DATA GROUP 10 - CHARACTER 5
2782  4A40  01          .BYTE  @001    ;DATA GROUP 10 - CHARACTER 5  PARITY
2783
2784  4A41  00          .BYTE  @000    ;DATA GROUP 11 - CHARACTER 1
2785  4A42  00          .BYTE  @000    ;DATA GROUP 11 - CHARACTER 1  PARITY
2786  4A43  FF          .BYTE  @377    ;DATA GROUP 11 - CHARACTER 2
2787  4A44  01          .BYTE  @001    ;DATA GROUP 11 - CHARACTER 2  PARITY
2788  4A45  00          .BYTE  @000    ;DATA GROUP 11 - CHARACTER 3
2789  4A46  00          .BYTE  @000    ;DATA GROUP 11 - CHARACTER 3  PARITY
2790  4A47  FF          .BYTE  @377    ;DATA GROUP 11 - CHARACTER 4
2791  4A48  01          .BYTE  @001    ;DATA GROUP 11 - CHARACTER 4  PARITY
2792  4A49  00          .BYTE  @000    ;DATA GROUP 11 - CHARACTER 5
2793  4A4A  00          .BYTE  @000    ;DATA GROUP 11 - CHARACTER 5  PARITY
2794
2795  4A4B  00          .BYTE  @000    ;DATA GROUP 12 - CHARACTER 1
2796  4A4C  00          .BYTE  @000    ;DATA GROUP 12 - CHARACTER 1  PARITY
2797  4A4D  FF          .BYTE  @377    ;DATA GROUP 12 - CHARACTER 2
2798  4A4E  01          .BYTE  @001    ;DATA GROUP 12 - CHARACTER 2  PARITY
2799  4A4F  00          .BYTE  @000    ;DATA GROUP 12 - CHARACTER 3
2800  4A50  00          .BYTE  @000    ;DATA GROUP 12 - CHARACTER 3  PARITY
2801  4A51  FF          .BYTE  @377    ;DATA GROUP 12 - CHARACTER 4
2802  4A52  01          .BYTE  @001    ;DATA GROUP 12 - CHARACTER 4  PARITY
2803  4A53  FF          .BYTE  @377    ;DATA GROUP 12 - CHARACTER 5
2804  4A54  01          .BYTE  @001    ;DATA GROUP 12 - CHARACTER 5  PARITY
2805
2806  4A55  FF          .BYTE  @377    ;DATA GROUP 13 - CHARACTER 1
2807  4A56  01          .BYTE  @001    ;DATA GROUP 13 - CHARACTER 1  PARITY
2808  4A57  FF          .BYTE  @377    ;DATA GROUP 13 - CHARACTER 2
2809  4A58  01          .BYTE  @001    ;DATA GROUP 13 - CHARACTER 2  PARITY
2810  4A59  FF          .BYTE  @377    ;DATA GROUP 13 - CHARACTER 3
2811  4A5A  01          .BYTE  @001    ;DATA GROUP 13 - CHARACTER 3  PARITY
2812  4A5B  FF          .BYTE  @377    ;DATA GROUP 13 - CHARACTER 4
2813  4A5C  01          .BYTE  @001    ;DATA GROUP 13 - CHARACTER 4  PARITY
2814  4A5D  00          .BYTE  @000    ;DATA GROUP 13 - CHARACTER 5
2815  4A5E  00          .BYTE  @000    ;DATA GROUP 13 - CHARACTER 5  PARITY
2816
2817  4A5F  00          .BYTE  @000    ;DATA GROUP 14 - CHARACTER 1
2818  4A60  00          .BYTE  @000    ;DATA GROUP 14 - CHARACTER 1  PARITY
2819  4A61  FF          .BYTE  @377    ;DATA GROUP 14 - CHARACTER 2
2820  4A62  01          .BYTE  @001    ;DATA GROUP 14 - CHARACTER 2  PARITY
2821  4A63  FF          .BYTE  @377    ;DATA GROUP 14 - CHARACTER 3
2822  4A64  01          .BYTE  @001    ;DATA GROUP 14 - CHARACTER 3  PARITY
2823  4A65  00          .BYTE  @000    ;DATA GROUP 14 - CHARACTER 4
2824  4A66  00          .BYTE  @000    ;DATA GROUP 14 - CHARACTER 4  PARITY
2825  4A67  FF          .BYTE  @377    ;DATA GROUP 14 - CHARACTER 5
2826  4A68  01          .BYTE  @001    ;DATA GROUP 14 - CHARACTER 5  PARITY
2827
2828  4A69  00          .BYTE  @000    ;DATA GROUP 15 - CHARACTER 1
2829  4A6A  00          .BYTE  @000    ;DATA GROUP 15 - CHARACTER 1  PARITY
2830  4A6B  FF          .BYTE  @377    ;DATA GROUP 15 - CHARACTER 2
```

```
2831  4A6C  01          .BYTE   @001          ;DATA GROUP 15 - CHARACTER 2  PARITY
2832  4A6D  FF          .BYTE   @377          ;DATA GROUP 15 - CHARACTER 3
2833  4A6E  01          .BYTE   @001          ;DATA GROUP 15 - CHARACTER 3  PARITY
2834  4A6F  FF          .BYTE   @377          ;DATA GROUP 15 - CHARACTER 4
2835  4A70  01          .BYTE   @001          ;DATA GROUP 15 - CHARACTER 4  PARITY
2836  4A71  00          .BYTE   @000          ;DATA GROUP 15 - CHARACTER 5
2837  4A72  00          .BYTE   @000          ;DATA GROUP 15 - CHARACTER 5  PARITY
2838
2839  4A73  00          .BYTE   @000          ;DATA GROUP 16 - CHARACTER 1
2840  4A74  00          .BYTE   @000          ;DATA GROUP 16 - CHARACTER 1  PARITY
2841  4A75  FF          .BYTE   @377          ;DATA GROUP 16 - CHARACTER 2
2842  4A76  01          .BYTE   @001          ;DATA GROUP 16 - CHARACTER 2  PARITY
2843  4A77  FF          .BYTE   @377          ;DATA GROUP 16 - CHARACTER 3
2844  4A78  01          .BYTE   @001          ;DATA GROUP 16 - CHARACTER 3  PARITY
2845  4A79  FF          .BYTE   @377          ;DATA GROUP 16 - CHARACTER 4
2846  4A7A  01          .BYTE   @001          ;DATA GROUP 16 - CHARACTER 4  PARITY
2847  4A7B  FF          .BYTE   @377          ;DATA GROUP 16 - CHARACTER 5
2848  4A7C  81          .BYTE   @201          ;DATA GROUP 16 - CHARACTER 5  PARITY
2849
```

```
2851                                    .SBTTL  TABLE TRANSLATED OUTPUT DATA
2852                                    ;OUTPUT DATA TABLE - THIS DATA IS COMPARED WITH THE OUTPUT OF THE READ
2853                                    ;CHANNEL BOARDS WHEN DATA READY SETS.  IT REFLECTS THE TRANSLATED INPUT
2854                                    ;DATA ON SUBGROUP BASIS.
2855
2856  4A7D  00                          M89500: .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 1
2857  4A7E  00                                  .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 1 PARITY
2858  4A7F  00                                  .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 2
2859  4A80  00                                  .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 2 PARITY
2860  4A81  00                                  .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 3
2861  4A82  00                                  .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 3 PARITY
2862  4A83  00                                  .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 4
2863  4A84  00                                  .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 4 PARITY
2864
2865  4A85  00                                  .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 1
2866  4A86  00                                  .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 1 PARITY
2867  4A87  00                                  .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 2
2868  4A88  00                                  .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 2 PARITY
2869  4A89  00                                  .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 3
2870  4A8A  00                                  .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 3 PARITY
2871  4A8B  FF                                  .BYTE   @377            ;DATA GROUP 2 - DATA VALUE 4
2872  4A8C  01                                  .BYTE   @001            ;DATA GROUP 2 - DATA VALUE 4 PARITY
2873
2874  4A8D  00                                  .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 1
2875  4A8E  00                                  .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 1 PARITY
2876  4A8F  00                                  .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 2
2877  4A90  00                                  .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 2 PARITY
2878  4A91  FF                                  .BYTE   @377            ;DATA GROUP 3 - DATA VALUE 3
2879  4A92  01                                  .BYTE   @001            ;DATA GROUP 3 - DATA VALUE 3 PARITY
2880  4A93  00                                  .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 4
2881  4A94  00                                  .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 4 PARITY
2882
2883  4A95  00                                  .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 1
2884  4A96  00                                  .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 1 PARITY
2885  4A97  00                                  .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 2
2886  4A98  00                                  .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 2 PARITY
2887  4A99  FF                                  .BYTE   @377            ;DATA GROUP 4 - DATA VALUE 3
2888  4A9A  01                                  .BYTE   @001            ;DATA GROUP 4 - DATA VALUE 3 PARITY
2889  4A9B  FF                                  .BYTE   @377            ;DATA GROUP 4 - DATA VALUE 4
2890  4A9C  01                                  .BYTE   @001            ;DATA GROUP 4 - DATA VALUE 4 PARITY
2891
2892  4A9D  00                                  .BYTE   @000            ;DATA GROUP 5 - DATA VALUE 1
2893  4A9E  00                                  .BYTE   @000            ;DATA GROUP 5 - DATA VALUE 1 PARITY
2894  4A9F  FF                                  .BYTE   @377            ;DATA GROUP 5 - DATA VALUE 2
2895  4AA0  01                                  .BYTE   @001            ;DATA GROUP 5 - DATA VALUE 2 PARITY
2896  4AA1  00                                  .BYTE   @000            ;DATA GROUP 5 - DATA VALUE 3
2897  4AA2  00                                  .BYTE   @000            ;DATA GROUP 5 - DATA VALUE 3 PARITY
2898  4AA3  00                                  .BYTE   @000            ;DATA GROUP 5 - DATA VALUE 4
2899  4AA4  00                                  .BYTE   @000            ;DATA GROUP 5 - DATA VALUE 4 PARITY
2900
2901  4AA5  00                                  .BYTE   @000            ;DATA GROUP 6 - DATA VALUE 1
2902  4AA6  00                                  .BYTE   @000            ;DATA GROUP 6 - DATA VALUE 1 PARITY
2903  4AA7  FF                                  .BYTE   @377            ;DATA GROUP 6 - DATA VALUE 2
2904  4AA8  01                                  .BYTE   @001            ;DATA GROUP 6 - DATA VALUE 2 PARITY
```

```
2905  4AA9  00          .BYTE  @000      ;DATA GROUP 6 - DATA VALUE 3
2906  4AAA  00          .BYTE  @000      ;DATA GROUP 6 - DATA VALUE 3 PARITY
2907  4AAB  FF          .BYTE  @377      ;DATA GROUP 6 - DATA VALUE 4
2908  4AAC  01          .BYTE  @001      ;DATA GROUP 6 - DATA VALUE 4 PARITY
2909
2910  4AAD  00          .BYTE  @000      ;DATA GROUP 7 - DATA VALUE 1
2911  4AAE  00          .BYTE  @000      ;DATA GROUP 7 - DATA VALUE 1 PARITY
2912  4AAF  FF          .BYTE  @377      ;DATA GROUP 7 - DATA VALUE 2
2913  4AB0  01          .BYTE  @001      ;DATA GROUP 7 - DATA VALUE 2 PARITY
2914  4AB1  FF          .BYTE  @377      ;DATA GROUP 7 - DATA VALUE 3
2915  4AB2  01          .BYTE  @001      ;DATA GROUP 7 - DATA VALUE 3 PARITY
2916  4AB3  00          .BYTE  @000      ;DATA GROUP 7 - DATA VALUE 4
2917  4AB4  00          .BYTE  @000      ;DATA GROUP 7 - DATA VALUE 4 PARITY
2918
2919  4AB5  00          .BYTE  @000      ;DATA GROUP 8 - DATA VALUE 1
2920  4AB6  00          .BYTE  @000      ;DATA GROUP 8 - DATA VALUE 1 PARITY
2921  4AB7  FF          .BYTE  @377      ;DATA GROUP 8 - DATA VALUE 2
2922  4AB8  01          .BYTE  @001      ;DATA GROUP 8 - DATA VALUE 2 PARITY
2923  4AB9  FF          .BYTE  @377      ;DATA GROUP 8 - DATA VALUE 3
2924  4ABA  01          .BYTE  @001      ;DATA GROUP 8 - DATA VALUE 3 PARITY
2925  4ABB  FF          .BYTE  @377      ;DATA GROUP 8 - DATA VALUE 4
2926  4ABC  01          .BYTE  @001      ;DATA GROUP 8 - DATA VALUE 4 PARITY
2927
2928  4ABD  FF          .BYTE  @377      ;DATA GROUP 9 - DATA VALUE 1
2929  4ABE  01          .BYTE  @001      ;DATA GROUP 9 - DATA VALUE 1 PARITY
2930  4ABF  00          .BYTE  @000      ;DATA GROUP 9 - DATA VALUE 2
2931  4AC0  00          .BYTE  @000      ;DATA GROUP 9 - DATA VALUE 2 PARITY
2932  4AC1  00          .BYTE  @000      ;DATA GROUP 9 - DATA VALUE 3
2933  4AC2  00          .BYTE  @000      ;DATA GROUP 9 - DATA VALUE 3 PARITY
2934  4AC3  00          .BYTE  @000      ;DATA GROUP 9 - DATA VALUE 4
2935  4AC4  00          .BYTE  @000      ;DATA GROUP 9 - DATA VALUE 4 PARITY
2936
2937  4AC5  FF          .BYTE  @377      ;DATA GROUP 10 - DATA VALUE 1
2938  4AC6  01          .BYTE  @001      ;DATA GROUP 10 - DATA VALUE 1 PARITY
2939  4AC7  00          .BYTE  @000      ;DATA GROUP 10 - DATA VALUE 2
2940  4AC8  00          .BYTE  @000      ;DATA GROUP 10 - DATA VALUE 2 PARITY
2941  4AC9  00          .BYTE  @000      ;DATA GROUP 10 - DATA VALUE 3
2942  4ACA  00          .BYTE  @000      ;DATA GROUP 10 - DATA VALUE 3 PARITY
2943  4ACB  FF          .BYTE  @377      ;DATA GROUP 10 - DATA VALUE 4
2944  4ACC  01          .BYTE  @001      ;DATA GROUP 10 - DATA VALUE 4 PARITY
2945
2946  4ACD  FF          .BYTE  @377      ;DATA GROUP 11 - DATA VALUE 1
2947  4ACE  01          .BYTE  @001      ;DATA GROUP 11 - DATA VALUE 1 PARITY
2948  4ACF  00          .BYTE  @000      ;DATA GROUP 11 - DATA VALUE 2
2949  4AD0  00          .BYTE  @000      ;DATA GROUP 11 - DATA VALUE 2 PARITY
2950  4AD1  FF          .BYTE  @377      ;DATA GROUP 11 - DATA VALUE 3
2951  4AD2  01          .BYTE  @001      ;DATA GROUP 11 - DATA VALUE 3 PARITY
2952  4AD3  00          .BYTE  @000      ;DATA GROUP 11 - DATA VALUE 4
2953  4AD4  00          .BYTE  @000      ;DATA GROUP 11 - DATA VALUE 4 PARITY
2954
2955  4AD5  FF          .BYTE  @377      ;DATA GROUP 12 - DATA VALUE 1
2956  4AD6  01          .BYTE  @001      ;DATA GROUP 12 - DATA VALUE 1 PARITY
2957  4AD7  00          .BYTE  @000      ;DATA GROUP 12 - DATA VALUE 2
2958  4AD8  00          .BYTE  @000      ;DATA GROUP 12 - DATA VALUE 2 PARITY
```

```
2959  4AD9  FF          .BYTE  @377          ;DATA GROUP 12 - DATA VALUE 3
2960  4ADA  01          .BYTE  @001          ;DATA GROUP 12 - DATA VALUE 3 PARITY
2961  4ADB  FF          .BYTE  @377          ;DATA GROUP 12 - DATA VALUE 4
2962  4ADC  01          .BYTE  @001          ;DATA GROUP 12 - DATA VALUE 4 PARITY
2963
2964  4ADD  FF          .BYTE  @377          ;DATA GROUP 13 - DATA VALUE 1
2965  4ADE  01          .BYTE  @001          ;DATA GROUP 13 - DATA VALUE 1 PARITY
2966  4ADF  FF          .BYTE  @377          ;DATA GROUP 13 - DATA VALUE 2
2967  4AE0  01          .BYTE  @001          ;DATA GROUP 13 - DATA VALUE 2 PARITY
2968  4AE1  00           BYTE  @000          ;DATA GROUP 13 - DATA VALUE 3
2969  4AE2  00          .BYTE  @000          ;DATA GROUP 13 - DATA VALUE 3 PARITY
2970  4AE3  00          .BYTE  @000          ;DATA GROUP 13 - DATA VALUE 4
2971  4AE4  00          .BYTE  @000          ;DATA GROUP 13 - DATA VALUE 4 PARITY
2972
2973  4AE5  FF          .BYTE  @377          ;DATA GROUP 14 - DATA VALUE 1
2974  4AE6  01          .BYTE  @001          ;DATA GROUP 14 - DATA VALUE 1 PARITY
2975  4AE7  FF          .BYTE  @377          ;DATA GROUP 14 - DATA VALUE 2
2976  4AE8  01          .BYTE  @001          ;DATA GROUP 14 - DATA VALUE 2 PARITY
2977  4AE9  00          .BYTE  @000          ;DATA GROUP 14 - DATA VALUE 3
2978  4AEA  00          .BYTE  @000          ;DATA GROUP 14 - DATA VALUE 3 PARITY
2979  4AEB  FF          .BYTE  @377          ;DATA GROUP 14 - DATA VALUE 4
2980  4AEC  01          .BYTE  @001          ;DATA GROUP 14 - DATA VALUE 4 PARITY
2981
2982  4AED  FF          .BYTE  @377          ;DATA GROUP 15 - DATA VALUE 1
2983  4AEE  01          .BYTE  @001          ;DATA GROUP 15 - DATA VALUE 1 PARITY
2984  4AEF  FF          .BYTE  @377          ;DATA GROUP 15 - DATA VALUE 2
2985  4AF0  01          .BYTE  @001          ;DATA GROUP 15 - DATA VALUE 2 PARITY
2986  4AF1  FF          .BYTE  @377          ;DATA GROUP 15 - DATA VALUE 3
2987  4AF2  01          .BYTE  @001          ;DATA GROUP 15 - DATA VALUE 3 PARITY
2988  4AF3  00          .BYTE  @000          ;DATA GROUP 15 - DATA VALUE 4
2989  4AF4  00          .BYTE  @000          ;DATA GROUP 15 - DATA VALUE 4 PARITY
2990
2991  4AF5  FF          .BYTE  @377          ;DATA GROUP 16 - DATA VALUE 1
2992  4AF6  01          .BYTE  @001          ;DATA GROUP 16 - DATA VALUE 1 PARITY
2993  4AF7  FF          .BYTE  @377          ;DATA GROUP 16 - DATA VALUE 2
2994  4AF8  01          .BYTE  @001          ;DATA GROUP 16 - DATA VALUE 2 PARITY
2995  4AF9  FF          .BYTF  @377          ;DATA GROUP 16 - DATA VALUE 3
2996  4AFA  01          .BYTE  @001          ;DATA GROUP 16 - DATA VALUE 3 PARITY
2997  4AFB  FF          .BYTE  @377          ;DATA GROUP 16 - DATA VALUE 4
2998  4AFC  01          .BYTE  @001          ;DATA GROUP 16 - DATA VALUE 4 PARITY
2999
```

```
3001                                    .SBTTL  PROGRAM VARIABLES
3002
3003  4AFD   00                 UNITMP: .BYTE   0               ;UNIT MAP
3004  4AFE   00                 SUNIT:  .BYTE   0               ;PORT/TU SELECT BYTE
3005  4AFF   00   00             AMTMSK: .WORD   0               ;AMTIE MASK WORD
3006  4B01   00   00             INTBLTMP: .WORD 0               ;TEMP. STORAGE FOR THE INPUT TABLE POINTER
3007  4B03   00   00             INTBL:  .WORD   0               ;INPUT TABLE ADDRESS STORAGE
3008  4B05   00   00             OUTTBL: .WORD   0               ;OUTPUT TABLE ADDRESS STORAGE
3009  4B07   00                  DATAA:  .BYTE   0               ;ACTUAL DATA AFTER TRANSLATION
3010  4B08   00                  DATAAP: .BYTE   0               ;ACTUAL DATA PARITY AFTER TRANSLATION
3011  4B09   00                  DATAE:  .BYTE   0               ;EXPECTED DATA AFTER TRANSLATION
3012  4B0A   00                  DATAEP: .BYTE   0               ;EXPECTED DATA PARITY AFTER TRANSLATION
3013  4B0B   00                  SUBCNT: .BYTE   0               ;SUBGROUP COUNTER
3014        0000                          .END
```

```
A     =%0007          ADATA = 0094          AMDCNA  4511  G       AMDCNB  4552  G
AMDCND  45D6  G       AMDCNE  4617  G       AMDCNG  468C  G       AMDCNH  46CD  G
AMDCNI  4769  G       AMDCNJ  4791  G       AMDCN1  438D  G       AMDCN2  43CE  G
AMDCN7  444B  G       AMDCN8  448C  G       AMDLPA  44DE          AMDLPB  4517
AMDLPC  4577          AMDLPD  45A4          AMDLPE  45DC          AMDLPG  4659
AMDLPH  4692          AMDLPI  471D          AMDLPJ  476F          AMDLP1  435F
AMDLP2  4393          AMDLP7  4418          AMDLP8  4451          AMDLP9  44B1
AMTIEP= 0001          AMTIE7= 0002          AMTMSK  4AFF          ARAIDF= 0098
ASAVE   4F9B          ATTCD   4F97          AXNUM   4F91          B     =%0000
BADST = 0090          BIT0  = 0001          BIT1  = 0002          BIT15 = 8000
BIT2  = 0004          BIT3  = 0008          BIT4  = 0010          BIT5  = 0020
BIT6  = 0040          BIT7  = 0080          BIT8  = 0100          BIT9  = 0200
BRKPBC= 4F0A          BRKRAM= 4F10          BRKSTR= 4E60          BRKXCT= 4F00
BSAVE   4F9C          BYTCNT= 00D4          BYTEH   4F24          BYTEL   4F23
C     =%0001          CASCT   4F21          CASCTL= 00A0          CASSTA= 00A0
CATTH = 0089          CATTL = 0088          CBUSST= 00A1          CBYTH = 008B
CBYTL = 008A          CDG1H = 0087          CDG1L = 0086          CDG2H = 0093
CDG2L = 0092          CDG3H = 0095          CDG3L = 0094          CDVTH = 008D
CDVTL = 008C          CHPTIE= 0028          CHOTIE= 0020          CH1TIE= 0021
CH2TIE= 0022          CH3TIE= 0023          CH4TIE= 0024          CH5TIE= 0025
CH6TIE= 0026          CH7TIE= 0027          CKAMP   49D6  G       CKAM0   4913  G
CKAM1   492B  G       CKAM2   4943  G       CKAM3   495B  G       CKAM4   4973  G
CKAM5   498B  G       CKAM6   49A3  G       CKAM7   49BB  G       CKLOP = 2815
CK5X4   48F8          CLEAR   48CD          CLKCTL= 00F0          CLOCK   4F26
CLRLP   48D1          CMC0H = 0099          CMC0L = 0098          CMC1H = 009B
CMC1L = 009A          CMC2H = 009D          CMC2L = 009C          CMC3H = 009F
CMC3L = 009E          CMINH = 0097          CMINL = 0096          CNTCTL  00D7
CRCWRD= 0018          CSAVE   4F9D          CSRLH = 0091          CSRLL = 0090
CTCH  = 0085          CTCL  = 0084          CTSTH = 008F          CTSTL = 008E
CXCTH = 0081          CXCTL = 0080          CXINH = 0083          CXINL = 0082
C.    = 0001          C.AVAI= 0080          C.DP  = 0008          C.DSE = 0070
C.DTU = 0003          C.DVA = 0008          C.FAIL= 00FC          C.FMT = 0070
C.FNCT= 003E          C.GO  = 0001          C.INTC= 00FE          C.MAIN= 0020
C.NSA = 0080          C.RCT = 00FC          C.SER = 0080          C.SHR = 0040
C.SKPC= 000F          C.TAPE= 0040          C.WCS = 0002          D     =%0002
DATAA   4B07          DATAAP  4B08          DATACT= 00D0          DATAE   4B09
DATAEP  4B0A          DBUS    4F28          DBUSCT= 00C0          DBUSST= 00C0
DDRA  = 00D8          DDRAIN= 0010          DDRB  = 00D9          DDRBIN= 0002
DDRC  = 00DA          DDRCIN= 0001          DDRCO = 0088          DDRCTL= 00DB
DIAFLG  4F22          DIAGPG= 4300          DIAGRM= 4F90          DIARD = 000B
DONE1 = 0045          DONINT= 0010          DSAVE   4F9E          DSE   = 0006
DUMMX   4858  G       DUMMY   431E  G       D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010          D.LAGC= 0020          D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008          D.WR4 = 0080          E     =%0003          ECCBAD= 0042
ECCCOR= 0019          ECCOK = 0041          ECCSTA= 001A          ECCTST= 000E
EDATA = 0095          EOTCLR= 0003          ERFLG   4F93          ERLP  = 2809
ERLPA = 280F          ERLPB = 2812          ERLPE = 280C          ERNUM   4F90
ERRCNT= 00D6          ESAVE   4F9F          EXIT    48CA          E.ACRC= 0010
E.AMT = 0020          E.CDP = 0080          E.CRC = 0080          E.PNTR= 0008
E.RPE = 0040          E.STEC= 0001          E.TTEC= 0002          E.UNC = 0004
FIFORD= 006A          FORMAT  4F25          FOUND   4344          FWDTST= 0061
GCRID = 0089          GCRSET= 0002          GOODTM= 0092          H     =%0004
HLSAVE  4FA0          IE    = 0008          INTBL   4B03          INTBLT  4B01
INTSTA= 00E0          ITERA   4F9A          I.PWR = 0020          I.RMPE= 0040
```

```
15.5   = 0010          16.5   = 0020          17.5   = 0040          KCALL  = 005F
KCLR   = 007B          KDEP   = 003F          KENAB  = 0078          KEXAM  = 003E
KEYBRD= 00C8           KEY1   = 0078          KEY10  = 006D          KEY11  = 006E
KEY12  = 006F          KEY13  = 005C          KEY14  = 005D          KEY15  = 005E
KEY16  = 005F          KEY17  = 003C          KEY18  = 003D          KEY19  = 003E
KEY2   = 0079          KEY20  = 003F          KEY3   = 007A          KEY4   = 007B
KEY5   = 0074          KEY6   = 0075          KEY7   = 0076          KEY8   = 0077
KEY9   = 006C          KINTA  = 006F          KLDAD  = 003D          KN0    = 003C
KN1    = 005C          KN2    = 005D          KN3    = 005E          KN4    = 006C
KN5    = 006D          KN6    = 006E          KN7    = 0074          KN8    = 0075
KN9    = 0076          KU2    = 0079          KU3    = 007A          KU8    = 0077
L      =%0005          LBLANK= 000F           LCE    = 000B          LCH    = 000C
LCL    = 000D          LCP    = 000E          LC0    = 0000          LC1    = 0001
LC2    = 0002          LC3    = 0003          LC4    = 0004          LC5    = 0005
LC6    = 0006          LC7    = 0007          LC8    = 0008          LC9    = 0009
LDLEDA= 00CA           LDLEDB= 00CB           LDLEDC= 00CC           LDLEDD= 00CD
LDLEDE= 00CE           LDLEDF= 00CF           LKDIAG= 2800           LKKBD  = 004C
LKKEY  = 0049          LKLWMG= 0058           LKLWMP= 0055           LKLWPG= 0052
LKLWPP= 004F           LKMOD7= 0046           LKOPR  = 0046          LPFLG    4F94
LPNUM    4F92          M      =%0006          MBSEL  = 00E0          MB.A   = 0008
MB.B   = 0004          MEMTOP= 4FFF           MINUS  = 000A          MM     = 8000
MSE    = 0008          MSGN   = 0017          MTACLR= 0000           MT.ARA= 0020
MT.CPE= 0080           MT.DSE= 0001           MT.FWD= 0040           MT.INH= 0008
MT.LWR= 0004           MT.MOT= 0002           MT.NWT= 0080           MT.PEC= 0040
MT.PSB= 0004           MT.PSO= 0001           MT.PS1= 0002           MT.REV= 0020
MT.WRT= 0010           MT.Z   = 0008          M.ATA  = 0080          M.CAPE= 0020
M.CONT= 0080           M.DEM  = 0020          M.EBL  = 0010          M.EXC  = 0008
M.FAIL= 0008           M.ILR  = 0010          M.INIT= 0010           M.OCC  = 0020
M.ONLI= 0001           M.PE   = 0040          M.PORT= 0080           M.RDFN= 0002
M.RDPE= 0008           M.RUN  = 0004          M.SCLK= 0001           M.TRA  = 0040
M.UNIT= 0007           M.WCLK= 0040           M.WCLN= 0080           M.WREN= 0080
M5.5   = 0001          M6.5   = 0002          M7.5   = 0004          M8950I   49DD
M8950O   4A7D          NOTCAP= 0088           OKAY   = 00FF          OPRRAM= 4300
OPSTRT= 0058           OPVER  = 0040          OUT      49DC          OUTTBL   4B05
PADCNT= 00D5           PADCRC= 0080           PDIAG  = 0048          PEID   = 008A
PENAB  = 004C          PESET  = 0001          PL     = 00B1          PRDD   = 004C
PRENF  = 009C          PS     = 00B2          PSTAT  = 0048          PSW    =%0009
P.AMTP= 0001           P.BCTC= 0040           P.CMDP= 0020           P.INTE= 0080
P.LCS  = 0040          P.LWR  = 0020          P.RDP  = 0002          P.RPEN= 0004
P.RPST= 0002           P.RP0E= 0020           P.RP1E= 0010           P.RP2E= 0020
P.RP3E= 0010           P.SING= 0080           P.STAT= 0002           P.STPE= 0080
P.TACH= 0008           P.TUPR= 0010           P.WCSP= 0004           P.WDS  = 0040
P.WFLP= 0001           P.WPEN= 0010           P.WP0E= 0008           P.WP1E= 0004
P.WP2E= 0008           P.WP3E= 0004           P.5VOK= 0002           QUE    = 281B
QUEM   = 281E          RAMT   = 0010          RARA   = 0006          RARAI  = 0004
RCHBD0= 0048           RCHBD1= 0047           RCHOK  = 0046          RCHTST= 000C
RCLRT  = 000D          RCMD   = 000B          RCMLP  = 0003          RCONT  = 0080
RDATA  = 0017          RDCLK  = 0010          RDON   = 0011          READG  = 0007
REND   = 0014          REQST  = 2806          RESCHR= 00D1           REVTST= 0064
REWIND= 0004           RFIFOL= 0008           RGCLK  = 0002          RGCRI  = 0003
RIBG   = 0001          RILL   = 0012          RINST  = 000C          RMCTST= 0008
RMK2   = 0013          RNOP   = 0000          RPATH  = 0001          RPBAD  = 0044
RPCHI  = 0001          RPCLK  = 0003          RPCTL  = 0009          RPEI   = 0002
RPFAIL= 0000           RPF1   = 009D          RPF2   = 009E          RPOK   = 0043
```

```
RPOSTN= 0016    RPSTA = 0015    RRCMT = 000A    RSTAT = 0002
RTIEB = 000A    RTIER = 0030    RTM   = 0005    RUNKI = 0009
RUPTST= 005E    RWDUNL= 0005    R.AMT = 0001    R.BOP = 0008
R.DATA= 0040    R.DON = 0002    R.DRDY= 0010    R.END = 0010
R.ILL = 0004    R.JVOK= 0004    R.MK2 = 0008    R.PLOD= 0008
R.PL00= 0010    R.PL01= 0020    R.POST= 0020    R.STNM= 0002
R.STOP= 0004    R.STPC= 0001    R.TBJN= 0080    R.TSTD= 0040
R.VOK = 0080    R00H  = 0081    R00L  = 0080    R01H  = 0083
R01L  = 0082    R02H  = 0085    R02L  = 0084    R03H  = 0087
R03L  = 0086    R04H  = 0089    R04L  = 0088    R05H  = 008B
R05L  = 008A    R06H  = 008D    R06L  = 008C    R07H  = 008F
R07L  = 008E    R10H  = 0091    R10L  = 0090    R11H  = 0093
R11L  = 0092    R12H  = 0095    R12L  = 0094    R13H  = 0097
R13L  = 0096    R14H  = 0099    R14L  = 0098    R15H  = 009B
R15L  = 009A    R16H  = 009D    R16L  = 009C    R17H  = 009F
R17L  = 009E    R7.5  = 0010    SELCLR= 0000    SETATA= 00A1
SID   = 0080    SOD   = 0080    SOE   = 0040    SP    =%0008
SSCLK = 0040    SSTEP = 0005    STACK = 4FFF    STATRM= 4F20
STPCT   4F20    STRSP = 5000    SUBCNT  4B0B    SUNIT   4AFE
TADR00= 0080    TADR01= 0081    TADR02= 0082    TADR03= 0083
TADR04= 0084    TADR05= 0085    TADR06= 0086    TADR07= 0087
TADR10= 0088    TADR11= 0089    TADR12= 008A    TADR13= 008B
TAMT  = 0044    TASEL = 0080    TCMD  = 0040    TC.FWD= 0040
TC.INH= 0008    TC.LWR= 0004    TC.REV= 0020    TC.WRT= 0010
TEMP    4F99    TEST1   4300    TEST2   4402    TEST3   44C8
TEST4   458E    TEST5   4643    TEST6   4709    TEST7   47CD
TMF   = 0099    TMRDY = 0040    TRKENA= 00D2    TSET  = 2803
TSTEND= 2818    TSTS  = 0040    TST01X  4351    TST02X  4407
TST03X  44CD    TST04X  4593    TST05X  4648    TST06X  470E
TST07A  4861    TST07B  4869    TST07C  4807    TST07X  47D2
TUSEL0= 00D1    TUSEL1= 00D2    TU78  = 0010    T.ATH0= 0001
T.ATH1= 0002    T.BOT = 0004    T.EOT = 0002    T.FPT = 0001
T.NTHR= 0004    T.ONL = 0020    T.PES = 0008    T.PSBJ= 0020
T.PSOJ= 0008    T.PS1J= 0010    T.RDY = 0080    T.RDYO= 0040
T.RWD = 0010    T.SCLK= 0002    UIBG  = 00A1    UNITMP  4AFD
UNITSL  4327    VALFC   4F98    VALTB   4F95    VELTST= 005B
WDR.P = 0010    WMCCTL= 00D3    WMCERR= 00DA    WMCSTA= 00D0
WRTCLK= 0000    WRTDAT= 00D3    W.ACRC= 0004    W.CRC = 0008
W.DIAG= 0002    W.DONN= 0040    W.ECC = 0010    W.ENAB= 0080
W.ERR = 0020    W.FMT = 0070    W.GCR = 0010    W.LEFT= 0004
W.ONES= 0020    W.RESI= 0002    W.REV = 0004    W.ROME= 0010
W.RST = 0001    W.SKIP= 000F    W.WRIT= 0008    W.XFER= 0020
X     =%000A    X.DONN= 0080    X.ENAB= 0040    X.PEPE= 0002
X.ROME= 0001    X.WCLK= 0001    Y     =%000B    .     = 4B0C

ERRORS DETECTED:  0

*RPM3.A78/PTP,RPM3=NLIST,PARAM,MACRO,LIST,RPM3
RUN-TIME: 5 8 0 SECONDS
CORE USED:  10K
```

     3   "MEMORY MAP" EXTENDED ADDRESS BIT
    19   READ MICRO CONTROLLER CONTROL & STATUS WORDS
   139   TRANSPORT/PORT COMMAND + STATUS WORDS
   380   COMMON ADDRESS SPACE REGISTERS
   444   MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
   485   MASSBUS CONTROL WORD DEFINITIONS
   557   WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
   709   KEYBOARD/DISPLAY CONTROL + STATUS WORD
   797   MISCELLANEOUS VARIABLE DEFINITIONS
   801   RAM ADDRESS DEFINITIONS
   819   LINKAGE TABLE ADDRESSES
   842   BIT DEFINITIONS
   856   RIM AND SIM INSTRUCTION BIT DEFINITIONS
   877   INTERRUPTABLE INSTRUCTION MACROS
   924   DIAGNOSTIC VARIABLES
   945   COMMON VARIABLE STORAGE
   959   MACRO CALL DEFINITIONS
  1047   TM78 MICRO-DIAGNOSTIC MACRO CALLS
  1101   8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
  1333   TEST 01 - INVALID 5X4 TRANSLATION TEST
  1559   TEST 02 - 5 X 4 TRANSLATION TEST (REVERSE)
  1851   SUBROUTINE CLEAR ALL TU PORTS
  1895   SUBROUTINE CHECK ILLEGAL 5X4
  1996   SUBROUTINE CHECK 5X4
  2139   TABLE INPUT DATA
  2296   TABLE UNTRANSLATED INPUT DATA
  2478   TABLE TRANSLATED OUTPUT DATA REVERSE
  2629   PROGRAM VARIABLES

```
                ;*************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;               "BYTE/SCLK COUNT NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               MM = DATA FORMAT FROM CAS REGISTER 2
                ;               NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;               BYTE-SCLK COUNT = LLL
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               LLL = AS ABOVE
                ;               MM  = AS ABOVE
                ;               NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;               TRANSITION COUNT = LLL
                ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;               EXPECTED 18 BITS =      E EEEEEE
                ;               ACTUAL 18 BITS =        A AAAAAA
                ;
                ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;               OF CAS REG 15 LOW BYTE.
                ;
                ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;               17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #' AND #5
                ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;               AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;               "SUBGROUP NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;*************************************************************************
```

```
1016                    ;*************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ;*************************************************************************
1035                    ;*************************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ;*************************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
1332                                .TITLE  RPM4 - READ PATH MICRO CONTROLLER PART #4
1333                                .SBTTL  TEST 01 - INVALID 5X4 TRANSLATION TEST
1334                                ;ID     RPM4-READ PATH MICRO CONTROLLER PART #4
1335  4300                          ST
 (1)                                ; *********************************************************************
 (1)                                ;*TEST TITLE
 (1)                                ;*---- -----
1336                                ;*INVALID 5 X 4 TRANSLATION TEST
1337  4300                          SD
 (1)                                ; *********************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
1338                                ;*THIS TEST CHECKS THE DATA HANDSHAKE BETWEEN THE ECC AND READ CHANNELS
1339                                ;*AS WELL AS THE READ CHANNELS ABILITY TO DETECT ALL INVALID 5 TO 4
1340                                ;*TRANSLATIONS POSSIBLE.
1341  4300                          SP
 (1)                                ; *********************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
1342                                ;*BGNTST
1343                                ;*    SET NORMAL READ PATH CLOCK
1344                                ;*    CALL SUBROUTINE CLEAR
1345                                ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
1346                                ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
1347                                ;*    DELAY 50 MICROSECONDS
1348                                ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
1349                                ;*    CLEAR THE TIE BUS
1350                                ;*    SET UP INPUT TABLE POINTER
1351                                ;*  BGNDO
1352                                ;*  : ENABLE THE READ PATH CLOCK
1353                                ;*  : SET PLO BYPASS MODE 2,TIE BUS JAM AND PLO DISABLE
1354                                ;*  : CLOCK THE FIFO
1355                                ;*  : ISSUE CLEAR ALL COMMAND
1356                                ;*  : WAIT
1357                                ;*  : STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
1358                                ;*  : INIT THE LOOP COUNT TO 5
1359                                ;*  : BGNDO
1360                                ;*  : :    MOVE 8 BITS OF DATA FROM THE INPUT TABLE TO THE TU PORT COMMAND REGISTER
1361                                ;*  : :    INCREMENT THE INPUT TABLE POINTER
1362                                ;*  : :    MOVE THE PARITY BIT FROM THE INPUT TABLE TO THE TU PORT CONTROL REGISTER
1363                                ;*  : :    CLOCK THE DATA INTO THE FIFO
1364                                ;*  : :    DECREMENT THE LOOP COUNT
1365                                ;*  : :    INCREMENT THE INPUT TABLE POINTER
1366                                ;*  : :    DO UNTIL LOOP COUNT = 0
1367                                ;*  : ENDDO
1368                                ;*  : ISSUE DIAGNOSTIC READ COMMAND
1369                                ;*  : SET UP WATCHDOG TIMER COUNT
1370                                ;*  : BGNDO
1371                                ;*  : :    SINGLE STEP THE READ PATH
1372                                ;*  : :    DECREMENT THE WATCHDOG TIMER
1373                                ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1374                                ;*  : ENDDO
1375                                ;*  : IF WATCHDOG TIMER-0
1376                                ;*  : :    THEN-ERROR
```

```
1377                                    ;*  : :    ELSE-CONTINUE
1378                                    ;*  : ENDIF
1379                                    ;*  : SINGLE STEP THE READ PATH
1380                                    ;*  : CALL SUBROUTINE CKILL
1381                                    ;*  : DO UNTIL END OF INPUT DATA TABLE
1382                                    ;*  ENDDO
1383                                    ;*ENDTST
1384  4300                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1385                                    ;*RPM4 MICRO TEST 01
1386                                    ;*RPM4 MICRO ERROR 01
1387                                    ;*RPM4-READ PATH-INVALID 5X4 TRANSLATION TEST
1388                                    ;*M8950'S, M8953
1389                                    ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1390                                    ;*FATAL ERROR - TEST ABORTED
1391                                    ;*
1392                                    ;*RPM4 MICRO TEST 01
1393                                    ;*RPM4 MICRO ERROR 03
1394                                    ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
1395                                    ;*M8950'S, M8953
1396                                    ;*CHANNEL 0 NOT ILLEGAL 5X4
1397                                    ;*
1398                                    ;*RPM4 MICRO TEST 01
1399                                    ;*RPM4 MICRO ERROR 04
1400                                    ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
1401                                    ;*M8950'S, M8953
1402                                    ;*CHANNEL 1 NOT ILLEGAL 5X4
1403                                    ;*
1404                                    ;*RPM4 MICRO TEST 01
1405                                    ;*RPM4 MICRO ERROR 05
1406                                    ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
1407                                    ;*M8950'S, M8953
1408                                    ;*CHANNEL 2 NOT ILLEGAL 5X4
1409                                    ;*
1410                                    ;*RPM4 MICRO TEST 01
1411                                    ;*RPM4 MICRO ERROR 06
1412                                    ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
1413                                    ;*M8950'S, M8953
1414                                    ;*CHANNEL 3 NOT ILLEGAL 5X4
1415                                    ;*
1416                                    ;*RPM4 MICRO TEST 01
1417                                    ;*RPM4 MICRO ERROR 07
1418                                    ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
1419                                    ;*M8950'S, M8953
1420                                    ;*CHANNEL 4 NOT ILLEGAL 5X4
1421                                    ;*
1422                                    ;*RPM4 MICRO TEST 01
1423                                    ;*RPM4 MICRO ERROR 10
1424                                    ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
1425                                    ;*M8950'S, M8953
1426                                    ;*CHANNEL 5 NOT ILLEGAL 5X4
1427                                    ;*
```

```
                                        ;*RPM4 MICRO TEST 01
                                        ;*RPM4 MICRO ERROR 11
                                        ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL 6 NOT ILLEGAL 5X4
                                        ;*
                                        ;*RPM4 MICRO TEST 01
                                        ;*RPM4 MICRO ERROR 12
                                        ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL 7 NOT ILLEGAL 5X4
                                        ;*
                                        ;*RPM4 MICRO TEST 01
                                        ;*RPM4 MICRO ERROR 13
                                        ;*RPM4-READ PATH-INVALID 4X4 TRANSLATION TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL P NOT ILLEGAL 5X4
1445  4300                              S
 (1)                                    ; ************************************************************************************
1446
1447  4300                              TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1)  4300  3E  01            7.0                     MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                     CALL    TSET            ;SETUP THE TEST
1448                                    ;%RPM4-READ PATH-INVALID 5X4 TRANSLATION TEST
1449                                    ·&M8950'S, M8953
1450  4305                                      REQ     @7,0,0,0,0
 (1)  4305  CD  06  28       18.0                     CALL    REQST
 (1)  4308  00                                        .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4309  00  00                                    .WORD   0               ,SYSTEM "0" COUNT
 (1)  430B  00  00                                    .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                        .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1)  430E  07                                        .BYTE   @7              ;REQUEST CODE
1451  430F                                    RIN     R12L
 (1)  430F  DB  94           10.0                     IN      R12L            ;READ R12L INTO AC
 (1)  4311  7F                4.0                     MOV     A,A             ;RETRY LINK
1452  4312  32  3F  48        13.0             STA     UNITMP
1453  4315  A7                 4.0             ANA     A
1454  4316  C2  27  43        10.0             JNZ     UNITSL
1455  4319                                    ERR     EXIT,DUMMY
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28       18.0                     CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                      MSGN    -       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                        .BYTE
 (1)  431E  CD  15  28       18.0             DUMMY:: CALL    CKLOP           ,CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  DD  44       10.0                     JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1456                                    ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1457                                    ;>FATAL ERROR - TEST ABORTED
1458  4324  C3  DD  44       10.0             JMP     EXIT
1459
1460  4327  06  00            7.0     UNITSL: MVI     B,@0            ;LOAD THE BASE UNIT NUMBER
1461  4329  3A  3F  48        13.0             LDA     UNITMP          ;DID THE USER SPECIFY TU
1462  432C  E6  01             7.0             ANI     @01             ;PORT 0?
1463  432E  C2  44  43        10.0             JNZ     FOUND           ;YES-GO USE PORT #0
```

```
1464  4331  04          4.0          INR   B              ;NO-UPDATE POINTER TO PORT #1
1465  4332  3A 3F 48   13.0          LDA   UNITMP         ;DID THE USER SPECIFY TU
1466  4335  E6 02      7.0          ANI   @02            ;PORT 1?
1467  4337  C2 44 43   10.0          JNZ   FOUND          ;YES-GO USE PORT #1
1468  433A  04          4.0          INR   B              ;NO-UPDATE POINTER TO PORT #2
1469  433B  3A 3F 48   13.0          LDA   UNITMP         ;DID THE USER SPECIFY TU
1470  433E  E6 04      7.0          ANI   @04            ;PORT 2?
1471  4340  C2 44 43   10.0          JNZ   FOUND          ;YES-GO USE PORT #2
1472  4343  04          4.0          INR   B              ;NO-ASSUME PORT #3
1473  4344  CD E0 44   18.0  FOUND:   CALL  CLEAR         ;CLEAR ALL THE TU PORTS
1474  4347  DB E0      10.0          IN    INTSTA         ;GET THE MASS BUS SELECT BIT
1475  4349  E6 80      7.0          ANI   BIT7           ;
1476  434B  B0          4.0          ORA   B              ;OR IN THE TU PORT SELECT BITS
1477  434C  D3 E0      10.0          OUT   MBSEL          ;SELECT THE MASS BUS AND TU PORTS
1478  434E  32 40 48   13.0          STA   SUNIT          ;STORE THE UNIT NUMBER FOR LATER USE
1479
1480  4351  3E 10      7.0  TST01X: MVI   A,RDCLK        ;SET NORMAL READ PATH CLOCKS
1481  4353  D3 F0      10.0          OUT   CLKCTL
1482
1483  4355  3E 04      7.0          MVI   A,P.RPEN              ;ENABLE THE READ PATH FROM THE TU PORT
1484  4357  D3 4C      10.0          OUT   PENAB          ;
1485  4359  3E 10      7.0          MVI   A,W.GCR        ;SET GCR MODE
1486  435B  D3 D3      10.0          OUT   WMCCTL
1487  435D  3E 60      7.0          MVI   A,P.LWR!P.LCS  ;SET THE PORT CONTROL TO
1488  435F  D3 48      10.0          OUT   PDIAG          ;LCS, LWR AND DPEN
1489  4361  AF          4.0          XRA   A
1490  4362  D3 44      10.0          OUT   TAMT           ;
1491  4364  3E 0A      7.0          MVI   A,10           ;DELAY
1492  4366  3D          4.0  1$:     DCR   A              ;50
1493  4367  C2 66 43   10.0          JNZ   1$             ;MICROSECONDS
1494
1495  436A  D3 D2      10.0          OUT   TRKENA         ;CLEAR ALL TRACKS FROM
1496  436C  D3 D2      10.0          OUT   TRKENA         ;THE TRANSLATOR
1497
1498  436E  AF          4.0          XRA   A              ;CLEAR THE TIE BUS
1499  436F  D3 0A      10.0          OUT   RTIEB
1500
1501  4371  21 93 46   10.0          LXI   H,M8950E        ;SET UP POINTER TO INPUT DATA TABLE
1502  4374  22 45 48   16.0          SHLD  INTBL          ;SAVE IT
1503  4377  3E A8      7.0  TST01C: MVI   A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1504                                                     ;TIE BUS JAM AND PLO DISABLE
1505  4379  D3 09      10.0          OUT   RPCTL
1506  437B  D3 08      10.0          OUT   RFIFOL         ;CLOCK FIFO TO CLEAR GCR DATA FF'S ON M8950'S
1507  437D  3E 0D      7.0          MVI   A,RCLRT        ;ISSUE CLEAR ALL COMMAND
1508  437F  D3 0B      10.0          OUT   RCMD
1509
1510  4381  00          4.0          NOP                  ;WAIT
1511  4382  00          4.0          NOP
1512  4383  00          4.0          NOP
1513  4384  00          4.0          NOP
1514
1515  4385  3E A9      7.0          MVI   A,R.PLO1!R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
1516                                                                      ;SET TIE BUS JAM AND PLO DISABLE
1517  4387  D3 09      10.0          OUT   RPCTL
```

```
1518
1519  4389  0E  02          7.0          MVI   C,2              ;SET UP THE FIFO LOAD COUNTER
1520  438B  2A  45  48       16.0  3$:    LHLD  INTBL            ;GET POINTER TO INPUT DATA TABLE
1521  438E  06  05           7.0          MVI   B,@5             ;SET UP LOOP COUNT
1522  4390  7E               7.0  1$:     MOV   A,M              ;GET A DATA BYTE
1523  4391  D3  40          10.0          OUT   TCMD             ;STORE DATA IN COMMAND ADDRESS
1524  4393  23               6.0          INX   H                ;POINT TO DATA PARITY
1525  4394  7E               7.0          MOV   A,M              ;GET THE DATA PARITY
1526  4395  07               4.0          RLC                    ;POSITION FOR OUTPUT
1527  4396  F6  60           7.0          ORI   P.LWR!P.LCS  ;OR IN CONTROL BITS
1528  4398  D3  48          10.0          OUT   PDIAG            ;OUTPUT THE DATA PARITY
1529  439A  D3  08          10.0          OUT   RFIFOL           ;CLOCK DATA INTO THE FIFO'S
1530
1531  439C  23               6.0          INX   H                ;UPDATE THE TABLE POINTER
1532  439D  05               4.0          DCR   B                ;DECREMENT LOOP COUNT
1533  439E  C2  90  43       10.0          JNZ   1$               ;DO UNTIL LOOP COUNT = 0
1534
1535  43A1  0D               4.0          DCR   C                ;DECREMENT THE FIFO LOAD LOOP COUNTER
1536  43A2  C2  8B  43       10.0          JNZ   3$               ;CONTINUE UNTIL ZERO
1537                                ;THE FIFO IS NOW FILLED WITH THE INPUT DATA TABLE
1538
1539  43A5  22  45  48       16.0          SHLD  INTBL            ;SAVE THE INPUT DATA TABLE POINTER
1540  43A8  3E  0B           7.0          MVI   A,DIARD          ;LOAD THE DIAGNOSTIC READ
1541  43AA  D3  0B          10.0          OUT   RCMD             ;COMMAND
1542  43AC  11  01  00       10.0          LXI   D,1              ;SET WATCH DOG INCREMENT
1543  43AF  21  A8  FD       10.0          LXI   H,-600           ;SET WATCH DOG COUNT TO 600
1544  43B2  D3  0C          10.0  2$:     OUT   RINST            ;STEP THE READ PATH
1545  43B4  19               10.0          DAD   D                ;WATCH DOG TIMEOUT?
1546  43B5  D2  B2  43       10.0          JNC   2$               ;NO-CONTINUE
1547  43B8  D3  0C          10.0  TST01A: OUT   RINST            ;SINGLE STEP THE READ PATH
1548  43BA  CD  0B  45       18.0          CALL  CKILL            ;CHECK FOR FAILING READ CHANNELS
1549  43BD  DA  51  43       10.0          JC    TST01X           ;LOOP ON ERROR
1550  43C0  2A  45  48       16.0          LHLD  INTBL            ;GET THE INPUT TABLE POINTER
1551  43C3  2B               6.0          DCX   H                ;UPDATE TO LAST PARITY BYTE
1552  43C4  7E               7.0          MOV   A,M              ;GET LAST PARITY BYTE
1553  43C5  A7               4.0          ANA   A                ;SET CONDITION BITS
1554  43C6  F2  77  43       10.0          JP    TST01C           ;CONTINUE TO END INPUT TABLE
1555  43C9  CD  E0  44       18.0          CALL  CLEAR            ;CLEAR ALL TU PORTS
1556  43CC                          ENDTST  TST01X
 (1)                        ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43CC                          REQ   7                      ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43CC  CD  06  28       18.0                         CALL    REQST
 (2)  43CF  00                                            .BYTE                ;DATA PATTERN NUMBER
 (2)  43D0  00  00                                        .WORD                ;SYSTEM "" COUNT
 (2)  43D2  00  00                                        .WORD                ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43D4  00                                            .BYTE                ;DATA COMPARE FLAG IF =1
 (2)  43D5  07                                            .BYTE   7            ;REQUEST CODE
 (1)  43D6  3A  9A  4F       13.0          LDA   ITERA            ;GET ITERATION COUNT
 (1)  43D9  3D               4.0          DCR   A                ;DOWNCOUNT
 (1)  43DA  32  9A  4F       13.0          STA   ITERA            ;SAVE COUNT
 (1)  43DD  F2  51  43       10.0          JP    TST01X           ;DO TEST UNTIL TILL = 0
1557
```

```
1559                                .SBTTL  TEST 02 - 5 X 4 TRANSLATION TEST (REVERSE)
1560  43E0                          ST
 (1)                                ;*********************************************************************
 (1)                                ;*TEST TITLE
 (1)                                ;*---- -----
1561                                ;*5 X 4 TRANSLATION TEST (REVERSE)
1562  43E0                          SD
 (1)                                ;*********************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
1563                                ;*THIS TEST CHECKS THE DATA HANDSHAKE BETWEEN THE ECC AND READ CHANNELS
1564                                ;*AS WELL AS THE READ CHANNELS ABILITY TO 5 TO 4 TRANSLATE ALL POSSIBLE
1565                                ;*COMBINATIONS OF INPUT DATA IN THE REVERSE MODE.
1566  43E0                          SP
 (1)                                ;*********************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
1567                                ;*BGNTST
1568                                ;*     SET NORMAL READ PATH CLOCK
1569                                ;*     CALL SUBROUTINE CLEAR
1570                                ;*     SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
1571                                ;*     CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
1572                                ;*     DELAY 50 MICROSECONDS
1573                                ;*     CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
1574                                ;*     CLEAR THE TIE BUS
1575                                ;*     SET UP INPUT TABLE POINTER
1576                                ;*     SET UP OUTPUT TABLE POINTER
1577                                ;*   BGNDO
1578                                ;*   : ENABLE THE READ PATH CLOCK
1579                                ;*   : SET PLO BYPASS MODE 2,TIE BUS JAM AND PLO DISABLE
1580                                ;*   : CLOCK THE FIFO
1581                                ;*   : ISSUE CLEAR ALL COMMAND
1582                                ;*   : WAIT
1583                                ;*   : STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
1584                                ;*   : PUT THE CONTROLLER IN GCR REVERSE MODE
1585                                ;*   : INIT THE SUB-GROUP COUNT TO 2
1586                                ;*   : BGNDO
1587                                ;*   : : INIT THE LOOP COUNT TO 5
1588                                ;*   : : BGNDO
1589                                ;*   : : :     MOVE 8 BITS OF DATA FROM THE INPUT TABLE TO THE TU PORT COMMAND REGISTER
1590                                ;*   : : :     INCREMENT THE INPUT TABLE POINTER
1591                                ;*   : : :     MOVE THE PARITY BIT FROM THE INPUT TABLE TO THE TU PORT CONTROL REGISTER
1592                                ;*   : : :     CLOCK THE DATA INTO THE FIFO
1593                                ;*   . : :     DECREMENT THE LOOP COUNT
1594                                ;*   : : :     INCREMENT THE INPUT TABLE POINTER
1595                                ;*   : : :     DO UNTIL LOOP COUNT = 0
1596                                ;*   : : ENDDO
1597                                ;*   : : DECREMENT THE SUB-GROUP COUNTER
1598                                ;*   : : DO UNTIL THE SUB-GROUP COUNTER =0
1599                                ;*   : ENDDO
1600                                ;*   : ISSUE DIAGNOSTIC READ COMMAND
1601                                ;*   : SET UP WATCHDOG TIMER COUNT
1602                                ;*   : BGNDO
1603                                ;*   : :     SINGLE STEP THE READ PATH
```

```
1604                                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1605                                    ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1606                                    ;*  : ENDDO
1607                                    ;*  : IF WATCHDOG TIMER=0
1608                                    ;*  : :    THEN-ERROR
1609                                    ;*  : :    ELSE-CONTINUE
1610                                    ;*  : ENDIF
1611                                    ;*  : SINGLE STEP THE READ PATH
1612                                    ;*  : INIT THE LOOP COUNT TO 4
1613                                    ;*  : BGNDO
1614                                    ;*  : :    SINGLE STEP THE READ PATH
1615                                    ;*  : :    INPUT THE ACTUAL DATA
1616                                    ;*  : :    INPUT THE ACTUAL PARITY
1617                                    ;*  : :    GET THE EXPECTED DATA FROM THE EXPECTED DATA TABLE
1618                                    ;*  : :    INCREMENT THE OUTPUT TABLE POINTER
1619                                    ;*  : :    GET THE EXPECTED PARITY FROM THE EXPECTED DATA TABLE
1620                                    ;*  : :    INCREMENT THE OUTPUT TABLE POINTER
1621                                    ;*  : :    CALL SUBROUTINE "CK5X4"
1622                                    ;*  : :    DECREMENT THE LOOP COUNT
1623                                    ;*  : :    DO UNTIL LOOP COUNT = 0
1624                                    ;*  : ENDDO
1625                                    ;*  :  DO UNTIL END OF INPUT DATA TABLE
1626                                    ;*  ENDDO
1627                                    ;*ENDTST
1628  43E0                              SE
 (1)                                    ; *****************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1629                                    ;*RPM4 MICRO TEST 02
1630                                    ;*RPM4 MICRO ERROR 02
1631                                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1632                                    ;*M8950'S, M8953
1633                                    ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
1634                                    ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1635                                    ;*FATAL ERROR - MICRO TEST ABORTED
1636                                    ;*
1637                                    ;*RPM4 MICRO TEST 02
1638                                    ;*RPM4 MICRO ERROR 14
1639                                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1640                                    ;*M8950'S, M8953
1641                                    ;*CHANNEL 0 FAILED
1642                                    ;*ACTUAL = NNNN
1643                                    ;*EXPECTED = NNNN
1644                                    ;*SUBGROUP NUMBER = LLL
1645                                    ;*
1646                                    ;*RPM4 MICRO TEST 02
1647                                    ;*RPM4 MICRO ERROR 15
1648                                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1649                                    ;*M8950'S, M8953
1650                                    ;*CHANNEL 1 FAILED
1651                                    ;*ACTUAL = NNNN
1652                                    ;*EXPECTED = NNNN
1653                                    ;*SUBGROUP NUMBER - LLL
1654                                    ;*
```

```
1655                    ;*RPM4 MICRO TEST 02
1656                    ;*RPM4 MICRO ERROR 16
1657                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1658                    ;*M8950'S, M8953
1659                    ;*CHANNEL 2 FAILED
1660                    ;*ACTUAL = NNNN
1661                    ;*EXPECTED = NNNN
1662                    ;*SUBGROUP NUMBER = LLL
1663                    ;*
1664                    ;*RPM4 MICRO TEST 02
1665                    ;*RPM4 MICRO ERROR 17
1666                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1667                    ;*M8950'S, M8953
1668                    ;*CHANNEL 3 FAILED
1669                    ;*ACTUAL = NNNN
1670                    ;*EXPECTED = NNNN
1671                    ;*SUBGROUP NUMBER = LLL
1672                    ;*
1673                    ;*RPM4 MICRO TEST 02
1674                    ;*RPM4 MICRO ERROR 20
1675                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1676                    ;*M8950'S, M8953
1677                    ;*CHANNEL 4 FAILED
1678                    ;*ACTUAL = NNNN
1679                    ;*EXPECTED = NNNN
1680                    ;*SUBGROUP NUMBER = LLL
1681                    ;*
1682                    ;*RPM4 MICRO TEST 02
1683                    ;*RPM4 MICRO ERROR 21
1684                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1685                    ;*M8950'S, M8953
1686                    ;*CHANNEL 5 FAILED
1687                    ;*ACTUAL = NNNN
1688                    ;*EXPECTED = NNNN
1689                    ;*SUBGROUP NUMBER = LLL
1690                    ;*
1691                    ;*RPM4 MICRO TEST 02
1692                    ;*RPM4 MICRO ERROR 22
1693                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1694                    ;*M8950'S, M8953
1695                    ;*CHANNEL 6 FAILED
1696                    ;*ACTUAL = NNNN
1697                    ;*EXPECTED = NNNN
1698                    ;*SUBGROUP NUMBER = LLL
1699                    ;*
1700                    ;*RPM4 MICRO TEST 02
1701                    ;*RPM4 MICRO ERROR 23
1702                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1703                    ;*M8950'S, M8953
1704                    ;*CHANNEL 7 FAILED
1705                    ;*ACTUAL = NNNN
1706                    ;*EXPECTED = NNNN
1707                    ;*SUBGROUP NUMBER = LLL
1708                    ;*
```

```
1709                                    ;*RPM4 MICRO TEST 02
1710                                    ;*RPM4 MICRO ERROR 24
1711                                    ;*RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1712                                    ;*M8950'S, M8953
1713                                    ;*CHANNEL P FAILED
1714                                    ;*ACTUAL = NNNN
1715                                    ;*EXPECTED = NNNN
1716                                    ;*SUBGROUP NUMBER = LLL
1717  43E0                              $
 (1)                                    ; ************************************************************************************
1718
1719  43E0                              TEST2:  TESTX   @2                  ;INITIALIZE THE TEST
 (1)  43E0  3E  02           7.0                         MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  43E2  CD  03  28       18.0                        CALL    TSET            ;SETUP THE TEST
1720                                    ;%RPM4-READ PATH-5X4 TRANSLATION TEST (REVERSE)
1721                                    ;&M8950'S, M8953
1722
1723  43E5  3E  10           7.0  TST02X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
1724  43E7  D3  F0          10.0         OUT     CLKCTL
1725
1726  43E9  CD  E0  44       18.0         CALL    CLEAR           ;CLEAR ALL TU PORTS
1727  43EC  3A  40  48       13.0         LDA     SUNIT
1728  43EF  D3  E0          10.0         OUT     MBSEL
1729  43F1  3E  04           7.0         MVI     A,P.RPEN
1730  43F3  D3  4C          10.0         OUT     PENAB
1731
1732  43F5  3E  60           7.0         MVI     A,P.LWR.P.LCS   ;SET THE PORT CONTROL TO
1733  43F7  D3  48           10.0         OUT     PDIAG           ;LCS, LWR AND DPEN
1734  43F9  AF               4.0         XRA     A
1735  43FA  D3  44           10.0         OUT     TAMT
1736  43FC  3E  0A           7.0         MVI     A,10            ;DELAY
1737  43FE  3D               4.0  1$:     DCR     A               ;50
1738  43FF  C2  FE  43       10.0         JNZ     1$              ;MICROSECONDS
1739
1740  4402  D3  D2          10.0         OUT     TRKENA          ;CLEAR ALL TRACKS FROM
1741  4404  D3  D2          10.0         OUT     TRKENA          ;THE TRANSLATOR
1742
1743  4406  AF               4.0         XRA     A               ;CLEAR THE TIE BUS
1744  4407  D3  0A          10.0         OUT     RTIEB
1745  4409  3E  01           7.0         MVI     A,@1            ;INITIALIZE THE SUBGROUP COUNT
1746  440B  32  4D  48       13.0         STA     SUBCNT          ;
1747
1748  440E  21  1F  47       10.0         LXI     H,M8950I        ;SET UP POINTER TO INPUT DATA TABLE
1749  4411  22  45  48       16.0         SHLD    INTBL           ;SAVE IT
1750  4414  21  BF  47       10.0         LXI     H,M8950O        ;SET UP POINTER TO OUTPUT DATA TABLE
1751  4417  22  47  48       16.0         SHLD    OUTTBL          ;SAVE IT
1752  441A  3E  A8           7.0  TST02C: MVI     A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1753                                                               ;TIE BUS JAM AND PLO DISABLE
1754  441C  D3  09          10.0         OUT     RPCTL
1755  441E  3E  14           7.0         MVI     A,W.GCR!W.REV           ;SET GCR MODE
1756  4420  D3  D3          10.0         OUT     WMCCTL
1757  4422  D3  08          10.0         OUT     RFIFOL          ;CLOCK FIFO TO CLEAR GCR DATA FF'S ON M8950'S
1758  4424  3E  0D           7.0         MVI     A,RCLRT         ;ISSUE CLEAR ALL COMMAND
1759  4426  D3  0B          10.0         OUT     RCMD
```

```
1760  4428  00              4.0          NOP                        ;WAIT
1761  4429  00              4.0          NOP
1762  442A  00              4.0          NOP
1763  442B  00              4.0          NOP
1764  442C  3E  A9           7.0          MVI     A,R.PLO!R.STPC.R.PLOD R.TBJN   ;STOP THE READ PATH
1765                                                                  ;SET TIE BUS JAM AND PLO DISABLE
1766  442E  D3  09          10.0          OUT     RPCTL
1767
1768  4430  0E  02           7.0          MVI     C,2,               ;INITIALIZE THE SUB-GROUP COUNTER
1769  4432  2A  45  48       16.0  3$:     LHLD    INTBL              ;GET POINTER TO INPUT DATA TABLE
1770  4435  06  05           7.0          MVI     B,@5               ;SET UP LOOP COUNT
1771  4437  7E               7.0  1$:     MOV     A,M                ;GET A DATA BYTE
1772  4438  D3  40          10.0          OUT     TCMD               ;STORE DATA IN COMMAND ADDRESS
1773  443A  23               6.0          INX     H                  ;POINT TO DATA PARITY
1774  443B  7E               7.0          MOV     A,M                ;GET THE DATA PARITY
1775
1776  443C  07               4.0          RLC                        ;POSITION FOR OUTPUT
1777  443D  F6  60           7.0          ORI     P.LWR.P.LCS        ;OR IN CONTROL BITS
1778
1779  443F  D3  48          10.0          OUT     PDIAG              ;OUTPUT THE DATA PARITY
1780  4441  D3  08          10.0          OUT     RFIFOL             ;CLOCK DATA INTO THE FIFO'S
1781  4443  23               6.0          INX     H                  ;UPDATE THE TABLE POINTER
1782  4444  05               4.0          DCR     B                  ;DECREMENT LOOP COUNT
1783  4445  C2  37  44       10.0          JNZ     1$                 ;DO UNTIL LOOP COUNT = 0
1784  4448  0D               4.0          DCR     C                  ;DECREMENT THE SUBGROUP COUNTER
1785  4449  C2  32  44       10.0          JNZ     3$                 ;LOAD THE SUB-GROUP TWICE
1786                                 ;THE FIFO IS NOW FILLED WITH THE INPUT DATA TABLE
1787  444C  22  43  48       16.0          SHLD    INTBLTMP           ;SAVE THE INPUT DATA TABLE POINTER
1788  444F  3E  0B           7.0          MVI     A,DIARD            ;LOAD THE DIAGNOSTIC READ
1789  4451  D3  0B          10.0          OUT     RCMD               ;COMMAND
1790  4453  11  01  00       10.0          LXI     D,1                ;SET WATCH DOG INCREMENT
1791  4456  21  A8  FD       10.0          LXI     H,-600             ;SET WATCH DOG COUNT TO 600
1792  4459  D3  0C          10.0  2$:     OUT     RINST              ;STEP THE READ PATH
1793  445B  DB  01          10.0          IN      RPCHI              ;DATA READY SET?
1794  445D  E6  10           7.0          ANI     R.DRDY
1795  445F  C2  74  44       10.0          JNZ     TST02A             ·YES-GO PROCESS
1796  4462  19               10.0          DAD     D                  ;WATCH DOG TIMEOUT?
1797  4463  D2  59  44       10.0          JNC     2$                 ;NO-CONTINUE
1798  4466                                 ERR     EXIT,DUMMX
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4466  CD  09  28       18.0                  CALL    ERLP               ;PROCESS ERROR - DO 2.3
 (1)       0002                                   MSGN    =       MSGN+1             ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4469  02                                    .BYTE   MSGN               ;MESSAGE NUMBER ID
 (1)  446A  00                                    .BYTE
 (1)  446B  CD  15  28       18.0          DUMMX:: CALL    CKLOP              ;CHECK LOOP FUNCTION - DO 2.3
 (1)  446E  DA  DD  44       10.0                  JC      EXIT               ;LOOP ADDRESS IF LOOP SPECIFIED
1799                                 ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
1800                                 ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1801                                 ;<FATAL ERROR - MICRO TEST ABORTED
1802  4471  C3  DD  44       10.0          JMP     EXIT
1803
1804  4474  2A  47  48       16.0  TST02A: LHLD    OUTTBL             ;GET ADDRESS OF EXPECTED OUTPUT DATA TABLE
1805  4477  00               4.0          NOP
1806  4478  00               4.0          NOP
```

```
1807  4479  00              4.0          NOP
1808  447A  0E  04          7.0          MVI     C,@4            ;INIT THE LOOP COUNT
1809  447C  00              4.0  TST02B: NOP
1810  447D  D3  0C          10.0         OUT     RINST           ;SINGLE STEP THE READ PATH
1811  447F  DB  17          10.0         IN      RDATA           ;INPUT THE ACTUAL DATA BYTE
1812  4481  32  49  48      13.0         STA     DATAA           ;STORE IT
1813  4484  DB  15          10.0         IN      RPSTA           ;INPUT THE ACTUAL PARITY BIT
1814  4486  E6  40          7.0          ANI     R.DATA          ;REMOVE JUNK BITS
1815  4488  07              4.0          RLC                     ;POSITION THE BIT FOR COMPARE
1816  4489  07              4.0          RLC
1817  448A  32  4A  48      13.0         STA     DATAAP          ;STORE IT
1818  448D  7E              7.0          MOV     A,M             ;GET THE EXPECTED DATA BYTE
1819  448E  32  4B  48      13.0         STA     DATAE           ;STORE IT
1820  4491  23              6.0          INX     H               ;UPDATE THE POINTER
1821  4492  7E              7.0          MOV     A,M             ;GET THE EXPECTED PARITY BYTE
1822  4493  32  4C  48      13.0         STA     DATAEP          ;STORE IT
1823  4496  EB              4.0          XCHG                    ;SAVE H AND L IN D AND E
1824  4497  3A  4D  48      13.0         LDA     SUBCNT          ;GET THE SUBGROUP COUNT
1825  449A                               ROUT    R05L            ;STORE IN CAS
 (1)  449A  D3  8A          10.0                 OUT     R05L            ;WRITE AC INTO R05L
 (1)  449C  7F              4.0                  MOV     A,A             ;RETRY LINK
1826  449D  AF              4.0          XRA     A               ;CLEAR THE HIGH HALF OF THE SUBGROUP COUNTER
1827  449E                               ROUT    R05H            ;
 (1)  449E  D3  8B          10.0                 OUT     R05H            ;WRITE AC INTO R05H
 (1)  44A0  7F              4.0                  MOV     A,A             ;RETRY LINK
1828  44A1  CD  AE  45      18.0         CALL    CK5X4           ;GO CHECK FOR FAILING READ CHANNELS
1829  44A4  DA  1A  44      10.0         JC      TST02C          ;IF LOOP ON ERROR REPEAT THE TEST
1830  44A7  EB              4.0          XCHG                    ;RESTORE H AND L FROM D AND E
1831  44A8  23              6.0          INX     H               ;UPDATE THE EXPECTED DATA TABLE POINTER
1832  44A9  0D              4.0          DCR     C               ;DECREMENT THE LOOP COUNT
1833  44AA  C2  7C  44      10.0         JNZ     TST02B          ;DO UNTIL LOOP COUNT = ZERO
1834  44AD  22  47  48      16.0         SHLD    OUTTBL          ;SAVE EXPECTED DATA TABLE POINTER
1835  44B0  3A  4D  48      13.0         LDA     SUBCNT          ;GET THE SUBGROUP COUNTER
1836  44B3  3C              4.0          INR     A               ;UPDATE FOR THE NEXT SUBGROUP
1837  44B4  32  4D  48      13.0         STA     SUBCNT          ;SAVE IT
1838  44B7  2A  43  48      16.0         LHLD    INTBLTMP        ;GET THE INPUT TABLE POINTER
1839  44BA  22  45  48      16.0         SHLD    INTBL           ;UPDATE THE INPUT TABLE POINTER
1840  44BD  2B              6.0          DCX     H               ;UPDATE TO LAST PARITY BYTE
1841  44BE  7E              7.0          MOV     A,M             ;GET LAST PARITY BYTE
1842  44BF  A7              4.0          ANA     A               ;SET CONDITION BITS
1843  44C0  F2  1A  44      10.0         JP      TST02C          ;CONTINUE UNTIL END OF INPUT TABLE
1844  44C3  CD  E0  44      18.0         CALL    CLEAR           ;CLEAR ALL TU PORTS
```

```
1846  44C6                                    ENDTST  TST02X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44C6                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44C6  CD  06  28      18.0                          CALL    REQST
 (2)  44C9  00                                            .BYTE                   ;DATA PATTERN NUMBER
 (2)  44CA  00  00                                        .WORD                   ;SYSTEM '"' COUNT
 (2)  44CC  00  00                                        .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  44CE  00                                            .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  44CF  07                                            .BYTE   7               ;REQUEST CODE
 (1)  44D0  3A  9A  4F      13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  44D3  3D               4.0          DCR     A               ;DOWNCOUNT
 (1)  44D4  32  9A  4F      13.0          STA     ITERA           ;SAVE COUNT
 (1)  44D7  F2  E5  43      10.0          JP      TST02X          ;DO TEST UNTIL TILL = 0
1847  44DA  CD  E0  44      18.0          CALL    CLEAR
1848
1849  44DD  C3  18  28      10.0  EXIT:   JMP     TSTEND
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ; ************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ; ************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ; ************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*    CLEAR PORT SELECT FOR TRANSPORT
                                        ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*    CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ; ************************************************************************
44E0  F5              12.0  CLEAR:  PUSH   PSW          ;SAVE THE SELECTED PORT #
44E1  C5              12.0          PUSH   B
44E2  06  00           7.0          MVI    B,0         ;START TO CLEAR AT PORT #0
44E4  DB  E0          10.0  CLRLP:  IN     INTSTA      ;GET MB SELECT INFO
44E6  E6  80           7.0          ANI    BIT7        ;SAVE ONLY THE MASSBUS SELECT BIT
44E8  B0               4.0          ORA    B           ;ADD IN THE SELECTED PORT #
44E9  D3  E0          10.0          OUT    MBSEL       ;RESET TO THIS PORT
44EB  3E  80           7.0          MVI    A,@200      ;LOAD MTA REGISTER #0 SELECT CODE
44ED  D3  40          10.0          OUT    TCMD        ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
44EF  AF               4.0          XRA    A           ;CLEAR TU COMMAND A
44F0  D3  40          10.0          OUT    TCMD        ;
44F2  3E  81           7.0          MVI    A,@201      ;LOAD MTA REGISTER #1 SELECT CODE
44F4  D3  40          10.0          OUT    TCMD        ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
44F6  3E  00           7.0          MVI    A,SELCLR    ;LOAD TU "CLEAR SELECT" COMMAND
44F8  D3  40          10.0          OUT    TCMD        ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
44FA  AF               4.0          XRA    A
44FB  D3  44          10.0          OUT    TAMT        ;CLEAR AMTIE WORD
44FD  D3  48          10.0          OUT    PDIAG       ;CLEAR DIAG CONTROL WORD
44FF  D3  4C          10.0          OUT    PENAB       ;CLEAR PORT ENABLE WORD
4501  04               4.0          INR    B           ;POINT TO THE NEXT PORT TO CLEAR
4502  78               4.0          MOV    A,B
4503  FE  04           7.0          CPI    4           ;DONE?
4505  C2  E4  44      10.0          JNZ    CLRLP       ;NO - CLEAR THIS PORT ALSO
4508  C1              10.0          POP    B           ;RESET B & C
4509  F1              10.0          POP    PSW         ;ALL DONE
450A  C9              10.0          RET                ;EXIT
```

```
                                .SBTTL  SUBROUTINE CHECK ILLEGAL 5X4
450B                            $SUB
                                ;*****************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*CHECK READ CHANNELS ILLEGAL 5X4
450B                            $D
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO CHECK THAT ALL READ CHANNEL "ILLEGAL 5X4"
                                ;*BITS EQUAL ONE AND TO FLAG ANY EXCEPTIONS WITH AN ERROR MESSAGE.
450B                            $P
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*  IF CHANNEL 0=ILLEGAL 5X4
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 0
                                ;*  ENDIF
                                ;*  IF CHANNEL 1=ILLEGAL 5X4
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 1
                                ;*  ENDIF
                                ;*  IF CHANNEL 2=ILLEGAL 5X4
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 2
                                ;*  ENDIF
                                ;*  IF CHANNEL 3=ILLEGAL 5X4
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 3
                                ;*  ENDIF
                                ;*  IF CHANNEL 4=ILLEGAL 5X4
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 4
                                ;*  ENDIF
                                ;*  IF CHANNEL 5=ILLEGAL 5X4
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 5
                                ;*  ENDIF
                                ;*  IF CHANNEL 6=ILLEGAL 5X4
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 6
                                ;*  ENDIF
                                ;*  IF CHANNEL 7=ILLEGAL 5X4
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 7
                                ;*  ENDIF
                                ;*  IF CHANNEL P=ILLEGAL 5X4
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL P
                                ;*  ENDIF
                                ;*ENDSUB
```

```
1940  450B                              S
 (1)                                    ; ***************************************************************************
1941
1942  450B   DB   12          10.0  CKILL:  IN      RILL            ;GET THE READ CHANNEL ILLEGAL BITS
1943  450D   E6   01           7.0          ANI     $01             ;CHANNEL 0 ILLEGAL?
1944  450F   C2   17   45      10.0          JNZ     CKIL0           ;YES-CONTINUE
1945  4512                                  ERR     OUTILL,CKIL0
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4512   CD   09   28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0003                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4515   03                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4516   00                               .BYTE
 (1)  4517   CD   15   28      18.0                  CKIL0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  451A   DA   AD   45      10.0                          JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1946                                    ;>CHANNEL 0 NOT ILLEGAL 5X4
1947
1948  451D   DB   12          10.0          IN      RILL            ;GET THE READ CHANNEL ILLEGAL BITS
1949  451F   E6   02           7.0          ANI     $02             ;CHANNEL 1 ILLEGAL?
1950  4521   C2   29   45      10.0          JNZ     CKIL1           ;YES-CONTINUE
1951  4524                                  ERR     OUTILL,CKIL1
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4524   CD   09   28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0004                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4527   04                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4528   00                               .BYTE
 (1)  4529   CD   15   28      18.0                  CKIL1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  452C   DA   AD   45      10.0                          JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1952                                    ;>CHANNEL 1 NOT ILLEGAL 5X4
1953
1954  452F   DB   12          10.0          IN      RILL            ;GET THE READ CHANNEL ILLEGAL BITS
1955  4531   E6   04           7.0          ANI     $04             ;CHANNEL 2 ILLEGAL?
1956  4533   C2   3B   45      10.0          JNZ     CKIL2           ;YES-CONTINUE
1957  4536                                  ERR     OUTILL,CKIL2
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4536   CD   09   28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0005                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4539   05                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  453A   00                               .BYTE
 (1)  453B   CD   15   28      18.0                  CKIL2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  453E   DA   AD   45      10.0                          JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1958                                    ;>CHANNEL 2 NOT ILLEGAL 5X4
1959
1960  4541   DB   12          10.0          IN      RILL            ;GET THE READ CHANNEL ILLEGAL BITS
1961  4543   E6   08           7.0          ANI     $08             ;CHANNEL 3 ILLEGAL?
1962  4545   C2   4D   45      10.0          JNZ     CKIL3           ;YES-CONTINUE
1963  4548                                  ERR     OUTILL,CKIL3
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4548   CD   09   28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0006                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  454B   06                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  454C   00                               .BYTE
 (1)  454D   CD   15   28      18.0                  CKIL3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4550   DA   AD   45      10.0                          JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1964                                    ;>CHANNEL 3 NOT ILLEGAL 5X4
```

```
1965  4553  DB  12        10.0        IN      RILL            ;GET THE READ CHANNEL ILLEGAL BITS
1966  4555  E6  10         7.0        ANI     $10             ;CHANNEL 4 ILLEGAL?
1967  4557  C2  5F  45    10.0        JNZ     CKIL4           ;YES-CONTINUE
1968  455A                             ERR     OUTILL,CKIL4
 (1)                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  455A  CD  09  28    18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0007                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  455D  07                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  455E  00                                    .BYTE
 (1)  455F  CD  15  28    18.0            CKIL4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4562  DA  AD  45    10.0                    JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1969                            ;>CHANNEL 4 NOT ILLEGAL 5X4
1970
1971  4565  DB  12        10.0        IN      RILL            ;GET THE READ CHANNEL ILLEGAL BITS
1972  4567  E6  20         7.0        ANI     $20             ;CHANNEL 5 ILLEGAL?
1973  4569  C2  71  45    10.0        JNZ     CKIL5           ;YES-CONTINUE
1974  456C                             ERR     OUTILL,CKIL5
 (1)                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  456C  CD  09  28    18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0008                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  456F  08                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4570  00                                    .BYTE
 (1)  4571  CD  15  28    18.0            CKIL5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4574  DA  AD  45    10.0                    JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1975                            ;>CHANNEL 5 NOT ILLEGAL 5X4
1976  4577  DB  12        10.0        IN      RILL            ;GET THE READ CHANNEL ILLEGAL BITS
1977  4579  E6  40         7.0        ANI     $40             ;CHANNEL 6 ILLEGAL?
1978  457B  C2  83  45    10.0        JNZ     CKIL6           ;YES-CONTINUE
1979  457E                             ERR     OUTILL,CKIL6
 (1)                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  457E  CD  09  28    18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0009                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4581  09                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4582  00                                    .BYTE
 (1)  4583  CD  15  28    18.0            CKIL6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4586  DA  AD  45    10.0                    JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1980                            ;>CHANNEL 6 NOT ILLEGAL 5X4
1981
1982  4589  DB  12        10.0        IN      RILL            ;GET THE READ CHANNEL ILLEGAL BITS
1983  458B  E6  80         7.0        ANI     $80             ;CHANNEL 7 ILLEGAL?
1984  458D  C2  95  45    10.0        JNZ     CKIL7           ;YES-CONTINUE
1985  4590                             ERR     OUTILL,CKIL7
 (1)                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4590  CD  09  28    18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000A                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4593  0A                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4594  00                                    .BYTE
 (1)  4595  CD  15  28    18.0            CKIL7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4598  DA  AD  45    10.0                    JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1986                            ;>CHANNEL 7 NOT ILLEGAL 5X4
```

```
1988  459B  DB  15        10.0          IN      RPSTA           ;GET THE CHANNEL P BIT
1989  459D  E6  04         7.0          ANI     R.ILL           ;CHANNEL P ILLEGAL?
1990  459F  C2  A7  45     10.0          JNZ     CKILP           ;YES-CONTINUE
1991  45A2                               ERR     OUTILL,CKILP
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45A2  CD  09  28     18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000B                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45A5  0B                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45A6  00                                          .BYTE
 (1)  45A7  CD  15  28     18.0                  CKILP:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45AA  DA  AD  45     10.0                          JC      OUTILL          ;LOOP ADDRESS IF LOOP SPECIFIED
1992                                    ;>CHANNEL P NOT ILLEGAL 5X4
1993
1994  45AD  C9                 10.0  OUTILL: RET                       ;RETURN
```

```
                                .SBTTL  SUBROUTINE CHECK 5X4
45AE                            SSUB
(1)                             ;*************************************************************************
(1)                             ;*SUBROUTINE TITLE
(1)                             ;*---------- -----
                                ;*CHECK 5X4 SUBROUTINE
45AE                            SD
(1)                             ;*************************************************************************
(1)                             ;*DESCRIPTION
(1)                             ;*-----------
                                ;*THIS SUBROUTINE IS USED TO ISOLATE A BIT FAILURE IN THE READ CHANNEL DATA
                                ;*OR AMTIE BYTES TO A SPECIFIC READ CHANNEL BY COMPARING THE ACTUAL AND
                                ;*EXPECTED INPUTS TO THE SUBROUTINE ON A BIT BY BIT BASIS.
45AE                            SP
(1)                             ;*************************************************************************
(1)                             ;*PROCEDURE
(1)                             ;*---------
                                ;*BGNSUB
                                ;*    IF CHANNEL 0 ACTUAL INFORMATION NOT = CHANNEL 0 EXPECTED
                                ;*      THEN-ERROR CHANNEL 0
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 1 ACTUAL INFORMATION NOT = CHANNEL 1 EXPECTED
                                ;*      THEN-ERROR CHANNEL 1
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 2 ACTUAL INFORMATION NOT = CHANNEL 2 EXPECTED
                                ;*      THEN-ERROR CHANNEL 2
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 3 ACTUAL INFORMATION NOT = CHANNEL 3 EXPECTED
                                ;*      THEN-ERROR CHANNEL 3
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 4 ACTUAL INFORMATION NOT = CHANNEL 4 EXPECTED
                                ;*      THEN-ERROR CHANNEL 4
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 5 ACTUAL INFORMATION NOT = CHANNEL 5 EXPECTED
                                ;*      THEN-ERROR CHANNEL 5
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 6 ACTUAL INFORMATION NOT = CHANNEL 6 EXPECTED
                                ;*      THEN-ERROR CHANNEL 6
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 7 ACTUAL INFORMATION NOT = CHANNEL 7 EXPECTED
                                ;*      THEN-ERROR CHANNEL 7
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL P ACTUAL INFORMATION NOT = CHANNEL P EXPECTED
                                ;*      THEN-ERROR CHANNEL P
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
```

```
2041                                        ;*ENDSUB
2042  45AE                                  S
 (1)                                        ; ************************************************************************
2043  45AE  21  4B  48      10.0  CK5X4:  LXI   H,DATAE          ;LOAD EXPECTED DATA ADDRESS
2044  45B1  3A  49  48      13.0          LDA   DATAA            ;GET THE ACTUAL DATA
2045  45B4                               ROUT  ADATA            ;WRITE THE ACTUAL DATA TO THE CAS
 (1)  45B4  D3  94            10.0                OUT   ADATA          ;WRITE AC INTO ADATA
 (1)  45B6  7F                 4.0                MOV   A,A            ;RETRY LINK
2046  45B7  E6  01             7.0          ANI   $01              ;REMOVE UNWANTED BITS
2047  45B9  47                 4.0          MOV   B,A              ;SAVE IN B
2048  45BA  7E                 7.0          MOV   A,M              ;GET EXPECTED DATA
2049  45BB                               ROUT  EDATA            ;WRITE THE EXPECTED DATA TO THE CAS
 (1)  45BB  D3  95            10.0                OUT   EDATA          ;WRITE AC INTO EDATA
 (1)  45BD  7F                 4.0                MOV   A,A            ;RETRY LINK
2050  45BE  E6  01             7.0          ANI   $01              ;REMOVE UNWANTED BITS
2051  45C0  B8                 4.0          CMP   B                ;COMPARE
2052  45C1  CA  C9  45       10.0          JZ    CKAM0            ;CONTINUE IF EQUAL
2053  45C4                               ERRB  OUT,CKAM0,@7
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  45C4  CD  12  28      18.0                CALL  ERLPB            ;PROCESS ERROR - DO 2.3
 (1)      000C                              MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45C7  0C                              .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  45C8  07                              .BYTE @7               ;PRINT ROUTINE NUMBER
 (1)  45C9  CD  15  28      18.0      CKAM0:: CALL  CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45CC  DA  92  46      10.0              JC    OUT              ;LOOP ADDRESS IF LOOP SPECIFIED
2054                                       ;>CHANNEL 0 FAILED
2055
2056  45CF  3A  49  48      13.0          LDA   DATAA            ;GET THE ACTUAL DATA
2057  45D2  E6  02             7.0          ANI   $02              ;REMOVE UNWANTED BITS
2058  45D4  47                 4.0          MOV   B,A              ;SAVE IN B
2059  45D5  7E                 7.0          MOV   A,M              ;GET EXPECTED DATA
2060  45D6  E6  02             7.0          ANI   $02              ;REMOVE UNWANTED BITS
2061  45D8  B8                 4.0          CMP   B                ;COMPARE
2062  45D9  CA  E1  45       10.0          JZ    CKAM1            ;CONTINUE IF EQUAL
2063  45DC                               ERRB  OUT,CKAM1,@7
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  45DC  CD  12  28      18.0                CALL  ERLPB            ;PROCESS ERROR - DO 2.3
 (1)      000D                              MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45DF  0D                              .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  45E0  07                              .BYTE @7               ;PRINT ROUTINE NUMBER
 (1)  45E1  CD  15  28      18.0      CKAM1:: CALL  CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45E4  DA  92  46      10.0              JC    OUT              ;LOOP ADDRESS IF LOOP SPECIFIED
2064                                       ;>CHANNEL 1 FAILED
```

```
2066  45E7  3A 49 48      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2067  45EA  E6 04          7.0          ANI     $04             ;REMOVE UNWANTED BITS
2068  45EC  47             4.0          MOV     B,A             ;SAVE IN B
2069  45ED  7E             7.0          MOV     A,M             ;GET EXPECTED DATA
2070  45EE  E6 04          7.0          ANI     $04             ;REMOVE UNWANTED BITS
2071  45F0  B8             4.0          CMP     B               ;COMPARE
2072  45F1  CA F9 45      10.0          JZ      CKAM2           ;CONTINUE IF EQUAL
2073  45F4                              ERRB    OUT,CKAM2,@7
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  45F4  CD 12 28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000E                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45F7  0E                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45F8  07                                  .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  45F9  CD 15 28      18.0          CKAM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45FC  DA 92 46      10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2074                                ;>CHANNEL 2 FAILED
2075
2076  45FF  3A 49 48      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2077  4602  E6 08          7.0          ANI     $08             ;REMOVE UNWANTED BITS
2078  4604  47             4.0          MOV     B,A             ;SAVE IN B
2079  4605  7E             7.0          MOV     A,M             ;GET EXPECTED DATA
2080  4606  E6 08          7.0          ANI     $08             ;REMOVE UNWANTED BITS
2081  4608  B8             4.0          CMP     B               ;COMPARE
2082  4609  CA 11 46      10.0          JZ      CKAM3           ;CONTINUE IF EQUAL
2083  460C                              ERRB    OUT,CKAM3,@7
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  460C  CD 12 28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000F                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  460F  0F                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4610  07                                  .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  4611  CD 15 28      18.0          CKAM3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4614  DA 92 46      10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2084                                ;>CHANNEL 3 FAILED
2085
2086  4617  3A 49 48      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2087  461A  E6 10          7.0          ANI     $10             ;REMOVE UNWANTED BITS
2088  461C  47             4.0          MOV     B,A             ;SAVE IN B
2089  461D  7E             7.0          MOV     A,M             ;GET EXPECTED DATA
2090  461E  E6 10          7.0          ANI     $10             ;REMOVE UNWANTED BITS
2091  4620  B8             4.0          CMP     B               ;COMPARE
2092  4621  CA 29 46      10.0          JZ      CKAM4           ;CONTINUE IF EQUAL
2093  4624                              ERRB    OUT,CKAM4,@7
 (1)                                    ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4624  CD 12 28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0010                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4627  10                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4628  07                                  .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  4629  CD 15 28      18.0          CKAM4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  462C  DA 92 46      10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2094                                ;>CHANNEL 4 FAILED
```

```
2096  462F  3A  49  48  13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2097  4632  E6  20       7.0          ANI     $20             ;REMOVE UNWANTED BITS
2098  4634  47           4.0          MOV     B,A             ;SAVE IN B
2099  4635  7E           7.0          MOV     A,M             ;GET EXPECTED DATA
2100  4636  E6  20       7.0          ANI     $20             ;REMOVE UNWANTED BITS
2101  4638  B8           4.0          CMP     B               ;COMPARE
2102  4639  CA  41  46  10.0          JZ      CKAM5           ;CONTINUE IF EQUAL
2103  463C                           ERRB    OUT,CKAM5,@7
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  463C  CD  12  28  18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0011                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  463F  11                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4640  07                               .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  4641  CD  15  28  18.0          CKAM5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4644  DA  92  46  10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2104                           ;>CHANNEL 5 FAILED
2105
2106  4647  3A  49  48  13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2107  464A  E6  40       7.0          ANI     $40             ;REMOVE UNWANTED BITS
2108  464C  47           4.0          MOV     B,A             ;SAVE IN B
2109  464D  7E           7.0          MOV     A,M             ;GET EXPECTED DATA
2110  464E  E6  40       7.0          ANI     $40             ;REMOVE UNWANTED BITS
2111  4650  B8           4.0          CMP     B               ;COMPARE
2112  4651  CA  59  46  10.0          JZ      CKAM6           ;CONTINUE IF EQUAL
2113  4654                           ERRB    OUT,CKAM6,@7
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4654  CD  12  28  18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0012                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4657  12                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4658  07                               .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  4659  CD  15  28  18.0          CKAM6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  465C  DA  92  46  10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2114                           ;>CHANNEL 6 FAILED
2115
2116  465F  3A  49  48  13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2117  4662  E6  80       7.0          ANI     $80             ;REMOVE UNWANTED BITS
2118  4664  47           4.0          MOV     B,A             ;SAVE IN B
2119  4665  7E           7.0          MOV     A,M             ;GET EXPECTED DATA
2120  4666  E6  80       7.0          ANI     $80             ;REMOVE UNWANTED BITS
2121  4668  B8           4.0          CMP     B               ;COMPARE
2122  4669  CA  71  46  10.0          JZ      CKAM7           ;CONTINUE IF EQUAL
2123  466C                           ERRB    OUT,CKAM7,@7
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  466C  CD  12  28  18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0013                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  466F  13                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4670  07                               .BYTE   @7              ;PRINT ROUTINE NUMBER
 (1)  4671  CD  15  28  18.0          CKAM7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4674  DA  92  46  10.0                  JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2124                           ;>CHANNEL 7 FAILED
```

```
 2126  4677  21  4C  48   10.0          LXI     H,DATAEP        ;GET THE EXPECTED DATA PARITY ADDRESS
 2127  467A  3A  4A  48   13.0          LDA     DATAAP          ;GET THE ACTUAL DATA
 2128  467D  E6  01        7.0          ANI     $01             ;REMOVE UNWANTED BITS
 2129  467F  47            4.0          MOV     B,A             ;SAVE IN B
 2130  4680  7E            7.0          MOV     A,M             ;GET EXPECTED DATA
 2131  4681  E6  01        7.0          ANI     $01             ;REMOVE UNWANTED BITS
 2132  4683  B8            4.0          CMP     B               ;COMPARE
 2133  4684  CA  8C  46   10.0          JZ      CKAMP           ;CONTINUE IF EQUAL
 2134  4687                             ERRB    OUT,CKAMP,@7
  (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  4687  CD  12  28   18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0014                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  468A  14                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  468B  07                                    .BYTE   @7              ;PRINT ROUTINE NUMBER
  (1)  468C  CD  15  28   18.0              CKAMP:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  468F  DA  92  46   10.               JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
 2135                          ,>CHANNEL P FAILED
 2136
 2137  4692  C9            10.0  OUT:     RET
```

```
2139                                    .SBTTL  TABLE INPUT DATA
2140
2141  4693  00                          M8950E: .BYTE   @000            ;DATA GROUP 1 - CHARACTER 1
2142  4694  00                                  .BYTE   @000            ;DATA GROUP 1 - CHARACTER 1  PARITY
2143  4695  00                                  .BYTE   @000            ;DATA GROUP 1 - CHARACTER 2
2144  4696  00                                  .BYTE   @000            ;DATA GROUP 1 - CHARACTER 2  PARITY
2145  4697  00                                  .BYTE   @000            ;DATA GROUP 1 - CHARACTER 3
2146  4698  00                                  .BYTE   @000            ;DATA GROUP 1 - CHARACTER 3  PARITY
2147  4699  00                                  .BYTE   @000            ;DATA GROUP 1 - CHARACTER 4
2148  469A  00                                  .BYTE   @000            ;DATA GROUP 1 - CHARACTER 4  PARITY
2149  469B  00                                  .BYTE   @000            ;DATA GROUP 1 - CHARACTER 5
2150  469C  00                                  .BYTE   @000            ;DATA GROUP 1 - CHARACTER 5  PARITY
2151
2152  469D  00                                  .BYTE   @000            ;DATA GROUP 2 - CHARACTER 1
2153  469E  00                                  .BYTE   @000            ;DATA GROUP 2 - CHARACTER 1  PARITY
2154  469F  00                                  .BYTE   @000            ;DATA GROUP 2 - CHARACTER 2
2155  46A0  00                                  .BYTE   @000            ;DATA GROUP 2 - CHARACTER 2  PARITY
2156  46A1  00                                  .BYTE   @000            ;DATA GROUP 2 - CHARACTER 3
2157  46A2  00                                  .BYTE   @000            ;DATA GROUP 2 - CHARACTER 3  PARITY
2158  46A3  00                                  .BYTE   @000            ;DATA GROUP 2 - CHARACTER 4
2159  46A4  00                                  .BYTE   @000            ;DATA GROUP 2 - CHARACTER 4  PARITY
2160  46A5  F                                   .BYTE   @377            ;DATA GROUP 2 - CHARACTER 5
2161  46A6  01                                  .BYTE   @001            ;DATA GROUP 2 - CHARACTER 5  PARITY
2162
2163  46A7  00                                  .BYTE   @000            ;DATA GROUP 3 - CHARACTER 1
2164  46A8  00                                  .BYTE   @000            ;DATA GROUP 3 - CHARACTER 1  PARITY
2165  46A9  00                                  .BYTE   @000            ;DATA GROUP 3 - CHARACTER 2
2166  46AA  00                                  .BYTE   @000            ;DATA GROUP 3 - CHAPACTER 2  PARITY
2167  46AB  00                                  .BYTE   @000            ;DATA GROUP 3 - CHARACTER 3
2168  46AC  00                                  .BYTE   @000            ;DATA GROUP 3 - CHARACTER 3  PARITY
2169  46AD  FF                                  .BYTE   @377            ;DATA GROUP 3 - CHARACTER 4
2170  46AE  01                                  .BYTE   @001            ;DATA GROUP 3 - CHARACTER 4  PARITY
2171  46AF  00                                  .BYTE   @000            ;DATA GROUP 3 - CHARACTER 5
2172  46B0  00                                  .BYTE   @000            ;DATA GROUP 3 - CHARACTER 5  PARITY
2173
2174  46B1  00                                  .BYTE   @000            ;DATA GROUP 4 - CHARACTER 1
2175  46B2  00                                  .BYTE   @000            ;DATA GROUP 4 - CHARACTER 1  PARITY
2176  46B3  00                                  .BYTE   @000            ;DATA GROUP 4 - CHARACTER 2
2177  46B4  00                                  .BYTE   @000            ;DATA GROUP 4 - CHARACTER 2  PARITY
2178  46B5  00                                  .BYTE   @000            ;DATA GROUP 4 - CHARACTER 3
2179  46B6  00                                  .BYTE   @000            ;DATA GROUP 4 - CHARACTER 3  PARITY
2180  46B7  FF                                  .BYTE   @377            ;DATA GROUP 4 - CHARACTER 4
2181  46B8  01                                  .BYTE   @001            ;DATA GROUP 4 - CHARACTER 4  PARITY
2182  46B9  FF                                  .BYTE   @377            ;DATA GROUP 4 - CHARACTER 5
2183  46BA  01                                  .BYTE   @001            ;DATA GROUP 4 - CHARACTER 5  PARITY
2184
2185  46BB  00                                  .BYTE   @000            ;DATA GROUP 5 - CHARACTER 1
2186  46BC  00                                  .BYTE   @000            ;DATA GROUP 5 - CHARACTER 1  PARITY
2187  46BD  00                                  .BYTE   @000            ;DATA GROUP 5 - CHARACTER 2
2188  46BE  00                                  .BYTE   @000            ;DATA GROUP 5 - CHARACTER 2  PARITY
2189  46BF  FF                                  .BYTE   @377            ;DATA GROUP 5 - CHARACTER 3
2190  46C0  01                                  .BYTE   @001            ;DATA GROUP 5 - CHARACTER 3  PARITY
2191  46C1  00                                  .BYTE   @000            ;DATA GROUP 5 - CHARACTER 4
2192  46C2  00                                  .BYTE   @000            ;DATA GROUP 5 - CHARACTER 4  PARITY
```

```
2193  46C3  00          .BYTE   @000           ;DATA GROUP 5 - CHARACTER 5
2194  46C4  00          .BYTE   @000           ;DATA GROUP 5 - CHARACTER 5  PARITY
2195
2196  46C5  00          .BYTE   @000           ;DATA GROUP 6 - CHARACTER 1
2197  46C6  00          .BYTE   @000           ;DATA GROUP 6 - CHARACTER 1  PARITY
2198  46C7  00          .BYTE   @000           ;DATA GROUP 6 - CHARACTER 2
2199  46C8  00          .BYTE   @000           ;DATA GROUP 6 - CHARACTER 2  PARITY
2200  46C9  FF          .BYTE   @377           ;DATA GROUP 6 - CHARACTER 3
2201  46CA  01          .BYTE   @001           ;DATA GROUP 6 - CHARACTER 3  PARITY
2202  46CB  00          .BYTE   @000           ;DATA GROUP 6 - CHARACTER 4
2203  46CC  00          .BYTE   @000           ;DATA GROUP 6 - CHARACTER 4  PARITY
2204  46CD  FF          .BYTE   @377           ;DATA GROUP 6 - CHARACTER 5
2205  46CE  01          .BYTE   @001           ;DATA GROUP 6 - CHARACTER 5  PARITY
2206
2207  46CF  00          .BYTE   @000           ;DATA GROUP 7 - CHARACTER 1
2208  46D0  00          .BYTE   @000           ;DATA GROUP 7 - CHARACTER 1  PARITY
2209  46D1  00          .BYTE   @000           ;DATA GROUP 7 - CHARACTER 2
2210  46D2  00          .BYTE   @000           ;DATA GROUP 7 - CHARACTER 2  PARITY
2211  46D3  FF          .BYTE   @377           ;DATA GROUP 7 - CHARACTER 3
2212  46D4  01          .BYTE   @001           ;DATA GROUP 7 - CHARACTER 3  PARITY
2213  46D5  FF          .BYTE   @377           ;DATA GROUP 7 - CHARACTER 4
2214  46D6  01          .BYTE   @001           ;DATA GROUP 7 - CHARACTER 4  PARITY
2215  46D7  00          .BYTE   @000           ;DATA GROUP 7 - CHARACTER 5
2216  46D8  00          .BYTE   @000           ;DATA GROUP 7 - CHARACTER 5  PARITY
2217
2218  46D9  00          .BYTE   @000           ;DATA GROUP 8 - CHARACTER 1
2219  46DA  00          .BYTE   @000           ;DATA GROUP 8 - CHARACTER 1  PARITY
2220  46DB  00          .BYTE   @000           ;DATA GROUP 8 - CHARACTER 2
2221  46DC  00          .BYTE   @000           ;DATA GROUP 8 - CHARACTER 2  PARITY
2222  46DD  FF          .BYTE   @377           ;DATA GROUP 8 - CHARACTER 3
2223  46DE  01          .BYTE   @001           ;DATA GROUP 8 - CHARACTER 3  PARITY
2224  46DF  FF          .BYTE   @377           ;DATA GROUP 8 - CHARACTER 4
2225  46E0  01          .BYTE   @001           ;DATA GROUP 8 - CHARACTER 4  PARITY
2226  46E1  FF          .BYTE   @377           ;DATA GROUP 8 - CHARACTER 5
2227  46E2  01          .BYTE   @001           ;DATA GROUP 8 - CHARACTER 5  PARITY
2228
2229  46E3  00          .BYTE   @000           ;DATA GROUP 9 - CHARACTER 1
2230  46E4  00          .BYTE   @000           ;DATA GROUP 9 - CHARACTER 1  PARITY
2231  46E5  FF          .BYTE   @377           ;DATA GROUP 9 - CHARACTER 2
2232  46E6  01          .BYTE   @001           ;DATA GROUP 9 - CHARACTER 2  PARITY
2233  46E7  00          .BYTE   @000           ;DATA GROUP 9 - CHARACTER 3
2234  46E8  00          .BYTE   @000           ;DATA GROUP 9 - CHARACTER 3  PARITY
2235  46E9  00          .BYTE   @000           ;DATA GROUP 9 - CHARACTER 4
2236  46EA  00          .BYTE   @000           ;DATA GROUP 9 - CHARACTER 4  PARITY
2237  46EB  00          .BYTE   @000           ;DATA GROUP 9 - CHARACTER 5
2238  46EC  00          .BYTE   @000           ;DATA GROUP 9 - CHARACTER 5  PARITY
2239
2240  46ED  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 1
2241  46EE  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 1  PARITY
2242  46EF  FF          .BYTE   @377           ;DATA GROUP 10 - CHARACTER 2
2243  46F0  01          .BYTE   @001           ;DATA GROUP 10 - CHARACTER 2  PARITY
2244  46F1  FF          .BYTE   @377           ;DATA GROUP 10 - CHARACTER 3
2245  46F2  01          .BYTE   @001           ;DATA GROUP 10 - CHARACTER 3  PARITY
2246  46F3  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 4
```

```
2247  46F4  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 4  PARITY
2248  46F5  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 5
2249  46F6  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 5  PARITY
2250
2251  46F7  FF          .BYTE   @377           ;DATA GROUP 11 - CHARACTER 1
2252  46F8  01          .BYTE   @001           ;DATA GROUP 11 - CHARACTER 1  PARITY
2253  46F9  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 2
2254  46FA  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 2  PARITY
2255  46FB  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 3
2256  46FC  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 3  PARITY
2257  46FD  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 4
2258  46FE  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 4  PARITY
2259  46FF  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 5
2260  4700  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 5  PARITY
2261
2262  4701  FF          .BYTE   @377           ;DATA GROUP 12 - CHARACTER 1
2263  4702  01          .BYTE   @001           ;DATA GROUP 12 - CHARACTER 1  PARITY
2264  4703  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 2
2265  4704  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 2  PARITY
2266  4705  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 3
2267  4706  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 3  PARITY
2268  4707  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 4
2269  4708  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 4  PARITY
2270  4709  FF          .BYTE   @377           ;DATA GROUP 12 - CHARACTER 5
2271  470A  01          .BYTE   @001           ;DATA GROUP 12 - CHARACTER 5  PARITY
2272
2273  470B  FF          .BYTE   @377           ;DATA GROUP 13 - CHARACTER 1
2274  470C  01          .BYTE   @001           ;DATA GROUP 13 - CHARACTER 1  PARITY
2275  470D  00          .BYTE   @000           ;DATA GROUP 13 - CHARACTER 2
2276  470E  00          .BYTE   @000           ;DATA GROUP 13 - CHARACTER 2  PARITY
2277  470F  FF          .BYTE   @377           ;DATA GROUP 13 - CHARACTER 3
2278  4710  01          .BYTE   @001           ;DATA GROUP 13 - CHARACTER 3  PAPITY
2279  4711  00          .BYTE   @000           ;DATA GROUP 13 - CHARACTER 4
2280  4712  00          .BYTE   @000           ;DATA GROUP 13 - CHARACTER 4  PARITY
2281  4713  00          .BYTE   @000           ;DATA GROUP 13 - CHARACTER 5
2282  4714  00          .BYTE   @000           ;DATA GROUP 13 - CHARACTER 5  PARITY
2283
2284  4715  FF          .BYTE   @377           ;DATA GROUP 14 - CHARACTER 1
2285  4716  01          .BYTE   @001           ;DATA GROUP 14 - CHARACTER 1  PARITY
2286  4717  FF          .BYTE   @377           ;DATA GROUP 14 - CHARACTER 2
2287  4718  01          .BYTE   @001           ;DATA GROUP 14 - CHARACTER 2  PARITY
2288  4719  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 3
2289  471A  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 3  PARITY
2290  471B  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 4
2291  471C  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 4  PARITY
2292  471D  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 5
2293  471E  80          .BYTE   @200           ;DATA GROUP 14 - CHARACTER 5  PARITY
2294
```

```
                                        .SBTTL  TABLE UNTRANSLATED INPUT DATA

                                ;INPUT DATA TABLE - THIS DATA IS LOADED INTO THE READ CHANNEL FIFO'S
                                ;ONE GROUP AT A TIME AND THEN TRANSLATED.

4720  01        M8950I: .BYTE   @001            ;DATA GROUP 1 - CHARACTER 1  PARITY
4721  FF                .BYTE   @377            ;DATA GROUP 1 - CHARACTER 2
4722  01                .BYTE   @001            ;DATA GROUP 1 - CHARACTER 2  PARITY
4723  00                .BYTE   @000            ;DATA GROUP 1 - CHARACTER 3
4724  00                .BYTE   @000            ;DATA GROUP 1 - CHARACTER 3  PARITY
4725  00                .BYTE   @000            ;DATA GROUP 1 - CHARACTER 4
4726  00                .BYTE   @000            ;DATA GROUP 1 - CHARACTER 4  PARITY
4727  FF                .BYTE   @377            ;DATA GROUP 1 - CHARACTER 5
4728  01                .BYTE   @001            ;DATA GROUP 1 - CHARACTER 5  PARITY

4729  FF                .BYTE   @377            ;DATA GROUP 2 - CHARACTER 1
472A  01                .BYTE   @001            ;DATA GROUP 2 - CHARACTER 1  PARITY
472B  FF                .BYTE   @377            ;DATA GROUP 2 - CHARACTER 2
472C  01                .BYTE   @001            ;DATA GROUP 2 - CHARACTER 2  PARITY
472D  00                .BYTE   @000            ;DATA GROUP 2 - CHARACTER 3
472E  00                .BYTE   @000            ;DATA GROUP 2 - CHARACTER 3  PARITY
472F  FF                .BYTE   @377            ;DATA GROUP 2 - CHARACTER 4
4730  01                .BYTE   @001            ;DATA GROUP 2 - CHARACTER 4  PARITY
4731  FF                .BYTE   @377            ;DATA GROUP 2 - CHARACTER 5
4732  01                .BYTE   @001            ;DATA GROUP 2 - CHARACTER 5  PARITY

4733  FF                .BYTE   @377            ;DATA GROUP 3 - CHARACTER 1
4734  01                .BYTE   @001            ;DATA GROUP 3 - CHARACTER 1  PARITY
4735  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 2
4736  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 2  PARITY
4737  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 3
4738  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 3  PARITY
4739  FF                .BYTE   @377            ;DATA GROUP 3 - CHARACTER 4
473A  01                .BYTE   @001            ;DATA GROUP 3 - CHARACTER 4  PARITY
473B  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 5
473C  00                .BYTE   @000            ;DATA GROUP 3 - CHARACTER 5  PARITY

473D  FF                .BYTE   @377            ;DATA GROUP 4 - CHARACTER 1
473E  01                .BYTE   @001            ;DATA GROUP 4 - CHARACTER 1  PARITY
473F  00                .BYTE   @000            ;DATA GROUP 4 - CHARACTER 2
4740  00                .BYTE   @000            ;DATA GROUP 4 - CHARACTER 2  PARITY
4741  00                .BYTE   @000            ;DATA GROUP 4 - CHARACTER 3
4742  00                .BYTE   @000            ;DATA GROUP 4 - CHARACTER 3  PARITY
4743  FF                .BYTE   @377            ;DATA GROUP 4 - CHARACTER 4
4744  01                .BYTE   @001            ;DATA GROUP 4 - CHARACTER 4  PARITY
4745  FF                .BYTE   @377            ;DATA GROUP 4 - CHARACTER 5
4746  01                .BYTE   @001            ;DATA GROUP 4 - CHARACTER 5  PARITY

4747  FF                .BYTE   @377            ;DATA GROUP 5 - CHARACTER 1
4748  01                .BYTE   @001            ;DATA GROUP 5 - CHARACTER 1  PARITY
4749  FF                .BYTE   @377            ;DATA GROUP 5 - CHARACTER 2
474A  01                .BYTE   @001            ;DATA GROUP 5 - CHARACTER 2  PARITY
474B  FF                .BYTE   @377            ;DATA GROUP 5 - CHARACTER 3
```

```
2350  474C  01          .BYTE   @001          ;DATA GROUP 5 - CHARACTER 3  PARITY
2351  474D  00          .BYTE   @000          ;DATA GROUP 5 - CHARACTER 4
2352  474E  0C          .BYTE   @000          ;DATA GROUP 5 - CHARACTER 4  PARITY
2353  474F  FF          .BYTE   @377          ;DATA GROUP 5 - CHARACTER 5
2354  4750  01          .BYTE   @001          ;DATA GROUP 5 - CHARACTER 5  PARITY
2355
2356  4751  FF          .BYTE   @377          ;DATA GROUP 6 - CHARACTER 1
2357  4752  01          .BYTE   @001          ;DATA GROUP 6 - CHARACTER 1  PARITY
2358  4753  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 2
2359  4754  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 2  PARITY
2360  4755  FF          .BYTE   @377          ;DATA GROUP 6 - CHARACTER 3
2361  4756  01          .BYTE   @001          ;DATA GROUP 6 - CHARACTER 3  PARITY
2362  4757  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 4
2363  4758  00          .BYTE   @000          ;DATA GROUP 6 - CHARACTER 4  PARITY
2364  4759  FF          .BYTE   @377          ;DATA GROUP 6 - CHARACTER 5
2365  475A  01          .BYTE   @001          ;DATA GROUP 6 - CHARACTER 5  PARITY
2366
2367  475B  FF          .BYTE   @377          ;DATA GROUP 7 - CHARACTER 1
2368  475C  01          .BYTE   @001          ;DATA GROUP 7 - CHARACTER 1  PARITY
2369  475D  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 2
2370  475E  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 2  PARITY
2371  475F  FF          .BYTE   @377          ;DATA GROUP 7 - CHARACTER 3
2372  4760  01          .BYTE   @001          ;DATA GROUP 7 - CHARACTER 3  PARITY
2373  4761  FF          .BYTE   @377          ;DATA GROUP 7 - CHARACTER 4
2374  4762  01          .BYTE   @001          ;DATA GROUP 7 - CHARACTER 4  PARITY
2375  4763  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 5
2376  4764  00          .BYTE   @000          ;DATA GROUP 7 - CHARACTER 5  PARITY
2377
2378  4765  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 1
2379  4766  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 1  PARITY
2380  4767  00          .BYTE   @000          ;DATA GROUP 8 - CHARACTER 2
2381  4768  00          .BYTE   @000          ;DATA GROUP 8 - CHARACTER 2  PARITY
2382  4769  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 3
2383  476A  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 3  PARITY
2384  476B  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 4
2385  476C  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 4  PARITY
2386  476D  FF          .BYTE   @377          ;DATA GROUP 8 - CHARACTER 5
2387  476E  01          .BYTE   @001          ;DATA GROUP 8 - CHARACTER 5  PARITY
2388
2389  476F  FF          .BYTE   @377          ;DATA GROUP 9 - CHARACTER 1
2390  4770  01          .BYTE   @001          ;DATA GROUP 9 - CHARACTER 1  PARITY
2391  4771  FF          .BYTE   @377          ;DATA GROUP 9 - CHARACTER 2
2392  4772  01          .BYTE   @001          ;DATA GROUP 9 - CHARACTER 2  PARITY
2393  4773  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 3
2394  4774  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 3  PARITY
2395  4775  FF          .BYTE   @377          ;DATA GROUP 9 - CHARACTER 4
2396  4776  01          .BYTE   @001          ;DATA GROUP 9 - CHARACTER 4  PARITY
2397  4777  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 5
2398  4778  00          .BYTE   @000          ;DATA GROUP 9 - CHARACTER 5  PARITY
2399
2400  4779  00          .BYTE   @000          ;DATA GROUP 10 - CHARACTER 1
2401  477A  00          .BYTE   @000          ;DATA GROUP 10 - CHARACTER 1  PARITY
2402  477B  FF          .BYTE   @377          ;DATA GROUP 10 - CHARACTER 2
2403  477C  01          .BYTE   @001          ;DATA GROUP 10 - CHARACTER 2  PARITY
```

```
2404  477D  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 3
2405  477E  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 3  PARITY
2406  477F  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 4
2407  4780  00          .BYTE   @000           ;DATA GROUP 10 - CHARACTER 4  PARITY
2408  4781  FF          .BYTE   @377           ;DATA GROUP 10 - CHARACTER 5
2409  4782  01          .BYTE   @001           ;DATA GROUP 10 - CHARACTER 5  PARITY
2410
2411  4783  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 1
2412  4784  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 1  PARITY
2413  4785  FF          .BYTE   @377           ;DATA GROUP 11 - CHARACTER 2
2414  4786  01          .BYTE   @001           ;DATA GROUP 11 - CHARACTER 2  PARITY
2415  4787  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 3
2416  4788  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 3  PARITY
2417  4789  FF          .BYTE   @377           ;DATA GROUP 11 - CHARACTER 4
2418  478A  01          .BYTE   @001           ;DATA GROUP 11 - CHARACTER 4  PARITY
2419  478B  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 5
2420  478C  00          .BYTE   @000           ;DATA GROUP 11 - CHARACTER 5  PARITY
2421
2422  478D  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 1
2423  478E  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 1  PARITY
2424  478F  FF          .BYTE   @377           ;DATA GROUP 12 - CHARACTER 2
2425  4790  01          .BYTE   @001           ;DATA GROUP 12 - CHARACTER 2  PARITY
2426  4791  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 3
2427  4792  00          .BYTE   @000           ;DATA GROUP 12 - CHARACTER 3  PARITY
2428  4793  FF          .BYTE   @377           ;DATA GROUP 12 - CHARACTER 4
2429  4794  01          .BYTE   @001           ;DATA GROUP 12 - CHARACTER 4  PARITY
2430  4795  FF          .BYTE   @377           ;DATA GROUP 12 - CHARACTER 5
2431  4796  01          .BYTE   @001           ;DATA GROUP 12 - CHARACTER 5  PARITY
2432
2433  4797  FF          .BYTE   @377           ;DATA GROUP 13 - CHARACTER 1
2434  4798  01          .BYTE   @001           ;DATA GROUP 13 - CHARACTER 1  PARITY
2435  4799  FF          .BYTE   @377           ;DATA GROUP 13 - CHARACTER 2
2436  479A  01          .BYTE   @001           ;DATA GROUP 13 - CHARACTER 2  PARITY
2437  479B  FF          .BYTE   @377           ;DATA GROUP 13 - CHARACTER 3
2438  479C  01          .BYTE   @001           ;DATA GROUP 13 - CHARACTER 3  PARITY
2439  479D  FF          .BYTE   @377           ;DATA GROUP 13 - CHARACTER 4
2440  479E  01          .BYTE   @001           ;DATA GROUP 13 - CHARACTER 4  PARITY
2441  479F  00          .BYTE   @000           ;DATA GROUP 13 - CHARACTER 5
2442  47A0  00          .BYTE   @000           ;DATA GROUP 13 - CHARACTER 5  PARITY
2443
2444  47A1  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 1
2445  47A2  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 1  PARITY
2446  47A3  FF          .BYTE   @377           ;DATA GROUP 14 - CHARACTER 2
2447  47A4  01          .BYTE   @001           ;DATA GROUP 14 - CHARACTER 2  PARITY
2448  47A5  FF          .BYTE   @377           ;DATA GROUP 14 - CHARACTER 3
2449  47A6  01          .BYTE   @001           ;DATA GROUP 14 - CHARACTER 3  PARITY
2450  47A7  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 4
2451  47A8  00          .BYTE   @000           ;DATA GROUP 14 - CHARACTER 4  PARITY
2452  47A9  FF          .BYTE   @377           ;DATA GROUP 14 - CHARACTER 5
2453  47AA  01          .BYTE   @001           ;DATA GROUP 14 - CHARACTER 5  PARITY
2454
2455  47AB  00          .BYTE   @000           ;DATA GROUP 15 - CHARACTER 1
2456  47AC  00          .BYTE   @000           ;DATA GROUP 15 - CHARACTER 1  PARITY
2457  47AD  FF          .BYTE   @377           ;DATA GROUP 15 - CHARACTER 2
```

```
2458  47AE  01          .BYTE   @001        ;DATA GROUP 15 - CHARACTER 2  PARITY
2459  47AF  FF          .BYTE   @377        ;DATA GROUP 15 - CHARACTER 3
2460  47B0  01          .BYTE   @001        ;DATA GROUP 15 - CHARACTER 3  PARITY
2461  47B1  FF          .BYTE   @377        ;DATA GROUP 15 - CHARACTER 4
2462  47B2  01          .BYTE   @001        ;DATA GROUP 15 - CHARACTER 4  PARITY
2463  47B3  00          .BYTE   @000        ;DATA GROUP 15 - CHARACTER 5
2464  47B4  00          .BYTE   @000        ;DATA GROUP 15 - CHARACTER 5  PARITY
2465
2466  47B5  00          .BYTE   @000        ;DATA GROUP 16 - CHARACTER 1
2467  47B6  00          .BYTE   @000        ;DATA GROUP 16 - CHARACTER 1  PARITY
2468  47B7  FF          .BYTE   @377        ;DATA GROUP 16 - CHARACTER 2
2469  47B8  01          .BYTE   @001        ;DATA GROUP 16 - CHARACTER 2  PARITY
2470  47B9  FF          .BYTE   @377        ;DATA GROUP 16 - CHARACTER 3
2471  47BA  01          .BYTE   @001        ;DATA GROUP 16 - CHARACTER 3  PARITY
2472  47BB  FF          .BYTE   @377        ;DATA GROUP 16 - CHARACTER 4
2473  47BC  01          .BYTE   @001        ;DATA GROUP 16 - CHARACTER 4  PARITY
2474  47BD  FF          .BYTE   @377        ;DATA GROUP 16 - CHARACTER 5
2475  47BE  81          .BYTE   @201        ;DATA GROUP 16 - CHARACTER 5  PARITY
2476
```

```
                                .SBTTL  TABLE TRANSLATED OUTPUT DATA REVERSE
                                ;OUTPUT DATA TABLE - THIS DATA IS COMPARED WITH THE OUTPUT OF THE READ
                                ;CHANNEL BOARDS WHEN DATA READY SETS.  IT REFLECTS THE TRANSLATED INPUT
                                ;DATA ON SUBGROUP BASIS.

47BF  FF                        M89500: .BYTE   @377            ;DATA GROUP 1 - DATA VALUE 1
47C0  01                                .BYTE   @001            ;DATA GROUP 1 - DATA VALUE 1 PARITY
47C1  FF                                .BYTE   @377            ;DATA GROUP 1 - DATA VALUE 2
47C2  01                                .BYTE   @001            ;DATA GROUP 1 - DATA VALUE 2 PARITY
47C3  00                                .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 3
47C4  00                                .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 3 PARITY
47C5  00                                .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 4
47C6  00                                .BYTE   @000            ;DATA GROUP 1 - DATA VALUE 4 PARITY

47C7  FF                                .BYTE   @377            ;DATA GROUP 2 - DATA VALUE 1
47C8  01                                .BYTE   @001            ;DATA GROUP 2 - DATA VALUE 1 PARITY
47C9  00                                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 2
47CA  00                                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 2 PARITY
47CB  00                                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 3
47CC  00                                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 3 PARITY
47CD  00                                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 4
47CE  00                                .BYTE   @000            ;DATA GROUP 2 - DATA VALUE 4 PARITY

47CF  FF                                .BYTE   @377            ;DATA GROUP 3 - DATA VALUE 1
47D0  01                                .BYTE   @001            ;DATA GROUP 3 - DATA VALUE 1 PARITY
47D1  00                                .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 2
47D2  00                                .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 2 PARITY
47D3  00                                .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 3
47D4  00                                .BYTE   @000            ;DATA GROUP 3 - DATA VALUE 3 PARITY
47D5  FF                                .BYTE   @377            ;DATA GROUP 3 - DATA VALUE 4
47D6  01                                .BYTE   @001            ;DATA GROUP 3 - DATA VALUE 4 PARITY

47D7  00                                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 1
47D8  00                                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 1 PARITY
47D9  00                                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 2
47DA  00                                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 2 PARITY
47DB  00                                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 3
47DC  00                                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 3 PARITY
47DD  00                                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 4
47DE  00                                .BYTE   @000            ;DATA GROUP 4 - DATA VALUE 4 PARITY

47DF  FF                                .BYTE   @377            ;DATA GROUP 5 - DATA VALUE 1
47E0  01                                .BYTE   @001            ;DATA GROUP 5 - DATA VALUE 1 PARITY
47E1  FF                                .BYTE   @377            ;DATA GROUP 5 - DATA VALUE 2
47E2  01                                .BYTE   @001            ;DATA GROUP 5 - DATA VALUE 2 PARITY
47E3  FF                                .BYTE   @377            ;DATA GROUP 5 - DATA VALUE 3
47E4  01                                .BYTE   @001            ;DATA GROUP 5 - DATA VALUE 3 PARITY
47E5  00                                .BYTE   @000            ;DATA GROUP 5 - DATA VALUE 4
47E6  00                                .BYTE   @000            ;DATA GROUP 5 - DATA VALUE 4 PARITY

47E7  FF                                .BYTE   @377            ;DATA GROUP 6 - DATA VALUE 1
47E8  01                                .BYTE   @001            ;DATA GROUP 6 - DATA VALUE 1 PARITY
47E9  00                                .BYTE   @000            ;DATA GROUP 6 - DATA VALUE 2
47EA  00                                .BYTE   @000            ;DATA GROUP 6 - DATA VALUE 2 PARITY
```

```
2532  47EB  FF          .BYTE   @377          ;DATA GROUP 6 - DATA VALUE 3
2533  47EC  01          .BYTE   @001          ;DATA GROUP 6 - DATA VALUE 3 PARITY
2534  47ED  00          .BYTE   @000          ;DATA GROUP 6 - DATA VALUE 4
2535  47EE  00          .BYTE   @000          ;DATA GROUP 6 - DATA VALUE 4 PARITY
2536
2537  47EF  FF          .BYTE   @377          ;DATA GROUP 7 - DATA VALUE 1
2538  47F0  01          .BYTE   @001          ;DATA GROUP 7 - DATA VALUE 1 PARITY
2539  47F1  00          .BYTE   @000          ;DATA GROUP 7 - DATA VALUE 2
2540  47F2  00          .BYTE   @000          ;DATA GROUP 7 - DATA VALUE 2 PARITY
2541  47F3  FF          .BYTE   @377          ;DATA GROUP 7 - DATA VALUE 3
2542  47F4  01          .BYTE   @001          ;DATA GROUP 7 - DATA VALUE 3 PARITY
2543  47F5  FF          .BYTE   @377          ;DATA GROUP 7 - DATA VALUE 4
2544  47F6  01          .BYTE   @001          ;DATA GROUP 7 - DATA VALUE 4 PARITY
2545
2546  47F7  00          .BYTE   @000          ;DATA GROUP 8 - DATA VALUE 1
2547  47F8  00          .BYTE   @000          ;DATA GROUP 8 - DATA VALUE 1 PARITY
2548  47F9  00          .BYTE   @000          ;DATA GROUP 8 - DATA VALUE 2
2549  47FA  00          .BYTE   @000          ;DATA GROUP 8 - DATA VALUE 2 PARITY
2550  47FB  FF          .BYTE   @377          ;DATA GROUP 8 - DATA VALUE 3
2551  47FC  01          .BYTE   @001          ;DATA GROUP 8 - DATA VALUE 3 PARITY
2552  47FD  00          .BYTE   @000          ;DATA GROUP 8 - DATA VALUE 4
2553  47FE  00          .BYTE   @000          ;DATA GROUP 8 - DATA VALUE 4 PARITY
2554
2555  47FF  FF          .BYTE   @377          ;DATA GROUP 9 - DATA VALUE 1
2556  4800  01          .BYTE   @001          ;DATA GROUP 9 - DATA VALUE 1 PARITY
2557  4801  FF          .BYTE   @377          ;DATA GROUP 9 - DATA VALUE 2
2558  4802  01          .BYTE   @001          ;DATA GROUP 9 - DATA VALUE 2 PARITY
2559  4803  00          .BYTE   @000          ;DATA GROUP 9 - DATA VALUE 3
2560  4804  00          .BYTE   @000          ;DATA GROUP 9 - DATA VALUE 3 PARITY
2561  4805  FF          .BYTE   @377          ;DATA GROUP 9 - DATA VALUE 4
2562  4806  01          .BYTE   @001          ;DATA GROUP 9 - DATA VALUE 4 PARITY
2563
2564  4807  00          .BYTE   @000          ;DATA GROUP 10 - DATA VALUE 1
2565  4808  00          .BYTE   @000          ;DATA GROUP 10 - DATA VALUE 1 PARITY
2566  4809  FF          .BYTE   @377          ;DATA GROUP 10 - DATA VALUE 2
2567  480A  01          .BYTE   @001          ;DATA GROUP 10 - DATA VALUE 2 PARITY
2568  480B  00          .BYTE   @000          ;DATA GROUP 10 - DATA VALUE 3
2569  480C  00          .BYTE   @000          ;DATA GROUP 10 - DATA VALUE 3 PARITY
2570  480D  00          .BYTE   @000          ;DATA GROUP 10 - DATA VALUE 4
2571  480E  00          .BYTE   @000          ;DATA GROUP 10 - DATA VALUE 4 PARITY
2572
2573  480F  00          .BYTE   @000          ;DATA GROUP 11 - DATA VALUE 1
2574  4810  00          .BYTE   @000          ;DATA GROUP 11 - DATA VALUE 1 PARITY
2575  4811  FF          .BYTE   @377          ;DATA GROUP 11 - DATA VALUE 2
2576  4812  01          .BYTE   @001          ;DATA GROUP 11 - DATA VALUE 2 PARITY
2577  4813  00          .BYTE   @000          ;DATA GROUP 11 - DATA VALUE 3
2578  4814  00          .BYTE   @000          ;DATA GROUP 11 - DATA VALUE 3 PARITY
2579  4815  FF          .BYTE   @377          ;DATA GROUP 11 - DATA VALUE 4
2580  4816  01          .BYTE   @001          ;DATA GROUP 11 - DATA VALUE 4 PARITY
2581
2582  4817  00          .BYTE   @000          ;DATA GROUP 12 - DATA VALUE 1
2583  4818  00          .BYTE   @000          ;DATA GROUP 12 - DATA VALUE 1 PARITY
2584  4819  00          .BYTE   @000          ;DATA GROUP 12 - DATA VALUE 2
2585  481A  00          .BYTE   @000          ;DATA GROUP 12 - DATA VALUE 2 PARITY
```

```
2586  481B  00          .BYTE   @000        ;DATA GROUP 12 - DATA VALUE 3
2587  481C  00          .BYTE   @000        ;DATA GROUP 12 - DATA VALUE 3 PARITY
2588  481D  FF          .BYTE   @377        ;DATA GROUP 12 - DATA VALUE 4
2589  481E  01          .BYTE   @001        ;DATA GROUP 12 - DATA VALUE 4 PARITY
2590
2591  481F  FF          .BYTE   @377        ;DATA GROUP 13 - DATA VALUE 1
2592  4820  01          .BYTE   @001        ;DATA GROUP 13 - DATA VALUE 1 PARITY
2593  4821  FF          .BYTE   @377        ;DATA GROUP 13 - DATA VALUE 2
2594  4822  01          .BYTE   @001        ;DATA GROUP 13 - DATA VALUE 2 PARITY
2595  4823  FF          .BYTE   @377        ;DATA GROUP 13 - DATA VALUE 3
2596  4824  01          .BYTE   @001        ;DATA GROUP 13 - DATA VALUE 3 PARITY
2597  4825  FF          .BYTE   @377        ;DATA GROUP 13 - DATA VALUE 4
2598  4826  01          .BYTE   @001        ;DATA GROUP 13 - DATA VALUE 4 PARITY
2599
2600  4827  00          .BYTE   @000        ;DATA GROUP 14 - DATA VALUE 1
2601  4828  00          .BYTE   @000        ;DATA GROUP 14 - DATA VALUE 1 PARITY
2602  4829  FF          .BYTE   @377        ;DATA GROUP 14 - DATA VALUE 2
2603  482A  01          .BYTE   @001        ;DATA GROUP 14 - DATA VALUE 2 PARITY
2604  482B  FF          .BYTE   @377        ;DATA GROUP 14 - DATA VALUE 3
2605  482C  01          .BYTE   @001        ;DATA GROUP 14 - DATA VALUE 3 PARITY
2606  482D  00          .BYTE   @000        ;DATA GROUP 14 - DATA VALUE 4
2607  482E  00          .BYTE   @000        ;DATA GROUP 14 - DATA VALUE 4 PARITY
2608
2609  482F  00          .BYTE   @000        ;DATA GROUP 15 - DATA VALUE 1
2610  4830  00          .BYTE   @000        ;DATA GROUP 15 - DATA VALUE 1 PARITY
2611  4831  FF          .BYTE   @377        ;DATA GROUP 15 - DATA VALUE 2
2612  4832  01          .BYTE   @001        ;DATA GROUP 15 - DATA VALUE 2 PARITY
2613  4833  FF          .BYTE   @377        ;DATA GROUP 15 - DATA VALUE 3
2614  4834  01          .BYTE   @001        ;DATA GROUP 15 - DATA VALUE 3 PARITY
2615  4835  FF          .BYTE   @377        ;DATA GROUP 15 - DATA VALUE 4
2616  4836  01          .BYTE   @001        ;DATA GROUP 15 - DATA VALUE 4 PARITY
2617
2618  4837  00          .BYTE   @000        ;DATA GROUP 16 - DATA VALUE 1
2619  4838  00          .BYTE   @000        ;DATA GROUP 16 - DATA VALUE 1 PARITY
2620  4839  00          .BYTE   @000        ;DATA GROUP 16 - DATA VALUE 2
2621  483A  00          .BYTE   @000        ;DATA GROUP 16 - DATA VALUE 2 PARITY
2622  483B  FF          .BYTE   @377        ;DATA GROUP 16 - DATA VALUE 3
2623  483C  01          .BYTE   @001        ;DATA GROUP 16 - DATA VALUE 3 PARITY
2624  483D  FF          .BYTE   @377        ;DATA GROUP 16 - DATA VALUE 4
2625  483E  01          .BYTE   @001        ;DATA GROUP 16 - DATA VALUE 4 PARITY
2626
2627
```

```
2629                                    .SBTTL  PROGRAM VARIABLES
2630  483F   00                         UNITMP: .BYTE   0               ;UNIT MAP
2631  4840   00                         SUNIT:  .BYTE   0               ;PORT/TU SELECT BYTE
2632  4841   00   00                    AMTMSK: .WORD   0               ;AMTIE MASK WORD
2633  4843   00   00                    INTBLTMP: .WORD 0               ;TEMP. STORAGE FOR THE INPUT TABLE POINTER
2634  4845   00   00                    INTBL:  .WORD   0               ;INPUT TABLE ADDRESS STORAGE
2635  4847   00   00                    OUTTBL: .WORD   0               ;OUTPUT TABLE ADDRESS STORAGE
2636  4849   00                         DATAA:  .BYTE   0               ;ACTUAL DATA AFTER TRANSLATION
2637  484A   00                         DATAAP: .BYTE   0               ;ACTUAL DATA PARITY AFTER TRANSLATION
2638  484B   00                         DATAE:  .BYTE   0               ;EXPECTED DATA AFTER TRANSLATION
2639  484C   00                         DATAEP: .BYTE   0               ;EXPECTED DATA PARITY AFTER TRANSLATION
2640  484D   00                         SUBCNT: .BYTE   0               ;SUBGROUP COUNTER
2641         0000                               .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
AMTMSK  4841           ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000          BADST  = 0090          BIT0   = 0001
BIT1   = 0002          BIT15  = 8000          BIT2   = 0004          BIT3   = 0008
BIT4   = 0010          BIT5   = 0020          BIT6   = 0040          BIT7   = 0080
BIT8   = 0100          BIT9   = 0200          BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60           BRKXCT= 4F00          BSAVE   4F9C          BYTCNT= 00D4
BYTEH   4F24           BYTEL   4F23          C      =%0001          CASCT   4F21
CASCTL= 00A0           CASSTA= 00A0          CATTH  = 0089          CATTL  = 0088
CBUSST= 00A1           CBYTH  = 008B          CBYTL  = 008A          CDG1H  = 0087
CDG1L  = 0086          CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095
CDG3L  = 0094          CDVTH  = 008D          CDVTL  = 008C          CHPTIE= 0028
CHOTIE= 0020           CH1TIE= 0021          CH2TIE= 0022          CH3TIE= 0023
CH4TIE= 0024           CH5TIE= 0025          CH6TIE= 0026          CH7TIE= 0027
CKAMP   468C  G        CKAM0   45C9  G       CKAM1   45E1  G       CKAM2   45F9  G
CKAM3   4611  G        CKAM4   4629  G       CKAM5   4641  G       CKAM6   4659  G
CKAM7   4671  G        CKILL   450B          CKILP   45A7  G       CKIL0   4517  G
CKIL1   4529  G        CKIL2   453B  G       CKIL3   454D  G       CKIL4   455F  G
CKIL5   4571  G        CKIL6   4583  G       CKIL7   4595  G       CKLOP  = 2815
CK5X4   45AE           CLEAR   44E0          CLKCTL= 00F0          CLOCK   4F26
CLRLP   44E4           CMC0H  = 0099          CMC0L  = 0098          CMC1H  = 009B
CMC1L  = 009A          CMC2H  = 009D          CMC2L  = 009C          CMC3H  = 009F
CMC3L  = 009E          CMINH  = 0097          CMINL  = 0096          CNTCTL= 00D7
CRCWRD= 0018           CSAVE   4F9D          CSRLH  = 0091          CSRLL  = 0090
CTCH   = 0085          CTCL   = 0084          CTSTH  = 008F          CTSTL  = 008E
CXCTH  = 0081          CXCTL  = 0080          CXINH  = 0083          CXINL  = 0082
C.     = 0001          C.AVAI= 0080          C.DP   = 0008          C.DSE  = 0010
C.DTU  = 0003          C.DVA  = 0008          C.FAIL= 00FC          C.FMT  = 0070
C.FNCT= 003E           C.GO   = 0001          C.INTC= 00FE          C.MAIN= 0020
C.NSA  = 0080          C.RCT  = 00FC          C.SER  = 0080          C.SHR  = 0040
C.SKPC= 000F           C.TAPE= 0040          C.WCS  = 0002          D      =%0002
DATAA   4849           DATAAP  484A          DATACT= 00D0          DATAE   484B
DATAEP  484C           DBUS    4F28          DBUSCT= 00C0          DBUSST= 00C0
DDRA   = 00D8          DDRAIN= 0010          DDRB   = 00D9          DDRBIN= 0002
DDRC   = 00DA          DDRCIN= 0001          DDRCO  = 0088          DDRCTL= 00DB
DIAFLG  4F22           DIAGPG= 4300          DIAGRM= 4F90          DIARD  = 000B
DONE1  = 0045          DONINT= 0010          DSAVE   4F9E          DSE    = 0006
DUMMX   446B  G        DUMMY   431E  G       D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010           D.LAGC= 0020          D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008           D.WR4  = 0080          E      =%0003          ECCBAD= 0042
FCCCOR= 0019           ECCOK  = 0041          ECCSTA= 001A          ECCTST= 000E
EDATA  = 0095          EOTCLR= 0003          ERFLG   4F93          ERLP   = 2809
ERLPA  = 280F          ERLPB  = 2812          ERLPE  = 280C          ERNUM   4F90
ERRCNT= 00D6           ESAVE   4F9F          EXIT    44DD          E.ACRC= 0010
E.AMT  = 0020          E.CDP  = 0080          E.CRC  = 0080          E.PNTR= 0008
E.RPE  = 0040          E.STEC= 0001          E.TTEC= 0002          E.UNC  = 0004
FIFORD= 006A           FORMAT  4F25          FOUND   4344          FWDTST= 0061
GCRID  = 0089          GCRSET= 0002          GOODTM= 0092          H      =%0004
HLSAVE  4FA0           IE     = 0008          INTBL   4845          INTBLT  4843
INTSTA= 00E0           ITERA   4F9A          I.PWR  = 0020          I.RMPE= 0040
I5.5   = 0010          I6.5   = 0020          I7.5   = 0040          KCALL  = 005F
KCLR   = 007B          KDEP   = 003F          KENAB  = 0078          KEXAM  = 003E
KEYBRD= 00C8           KEY1   = 0078          KEY10  = 006D          KEY11  = 006E
KEY12  = 006F          KEY13  = 005C          KEY14  = 005D          KEY15  = 005E
```

```
KEY16 = 005F            KEY17 = 003C            KEY18 = 003D            KEY19 = 003E
KEY2  = 0079            KEY20 = 003F            KEY3  = 007A            KEY4  = 007B
KEY5  = 0074            KEY6  = 0075            KEY7  = 0076            KEY8  = 0077
KEY9  = 006C            KINTA = 006F            KLDAD = 003D            KN0   = 003C
KN1   = 005C            KN2   = 005D            KN3   = 005E            KN4   = 006C
KN5   = 006D            KN6   = 006E            KN7   = 0074            KN8   = 0075
KN9   = 0076            KU2   = 0079            KU3   = 007A            KU8   = 0077
L     =%0005            LBLANK= 000F            LCE   = 000B            LCH   = 000C
LCL   = 000D            LCP   = 000E            LC0   = 0000            LC1   = 0001
LC2   = 0002            LC3   = 0003            LC4   = 0004            LC5   = 0005
LC6   = 0006            LC7   = 0007            LC8   = 0008            LC9   = 0009
LDLEDA= 00CA            LDLEDB= 00CB            LDLEDC= 00CC            LDLEDD= 00CD
LDLEDE= 00CE            LDLEDF= 00CF            LKDIAG= 2800            LKKBD = 004C
LKKEY = 0049            LKLWMG= 0058            LKLWMP= 0055            LKLWPG= 0052
LKLWPP= 004F            LKMOD7= 0046            LKOPR = 0046            LPFLG   4F94
LPNUM   4F92            M     =%0006            MBSEL = 00E0            MB.A  = 0008
MB.B  = 0004            MEMTOP= 4FFF            MINUS = 000A            MM    = 8000
MSE   = 0008            MSGN  = 0014            MTACLR= 0000            MT.ARA= 0020
MT.CPE= 0080            MT.DSE= 0001            MT.FWD= 0040            MT.INH= 0008
MT.LWR= 0004            MT.MOT= 0002            MT.NWT= 0080            MT.PEC= 0040
MT.PSB= 0004            MT.PS0= 0001            MT.PS1= 0002            MT.REV= 0020
MT.WRT= 0010            MT.Z  = 0008            M.ATA = 0080            M.CAPE= 0020
M.CONT= 0080            M.DEM = 0020            M.EBL = 0010            M.EXC = 0008
M.FAIL= 0008            M.ILR = 0010            M.INIT= 0010            M.OCC = 0020
M.ONLI= 0001            M.PE  = 0040            M.PORT= 0080            M.RDEN= 0002
M.RDPE= 0008            M.RUN = 0004            M.SCLK= 0001            M.TRA = 0040
M.UNIT= 0007            M.WCLK= 0040            M.WCLN= 0080            M.WREN= 0080
M5.5  = 0001            M6.5  = 0002            M7.5  = 0004            M8950E  4693
M8950I  471F            M89500  47BF            NOTCAP= 0088            OKAY  = 00FF
OPRRAM= 4300            OPSTRT= 0058            OPVER = 0040            OUT     4692
OUTILL  45AD            OUTTBL  4847            PADCNT= 00D5            PADCRC= 0080
PDIAG = 0048            PEID  = 008A            PENAB = 004C            PESET = 0001
PL    = 00B1            PRDD  = 004C            PRENF = 009C            PS    = 00B2
PSTAT = 0048            PSW   =%0009            P.AMTP= 0001            P.BCTC= 0040
P.CMDP= 0020            P.INTE= 0080            P.LCS = 0040            P.LWR = 0020
P.RDP = 0002            P.RPEN= 0004            P.RPST= 0002            P.RPOE= 0020
P.RP1E= 0010            P.RP2E= 0020            P.RP3E= 0010            P.SING= 0080
P.STAT= 0002            P.STPE= 0080            P.TACH= 0008            P.TUPR= 0010
P.WCSP= 0004            P.WDS = 0040            P.WFLP= 0001            P.WPEN= 0010
P.WPOE= 0008            P.WP1E= 0004            P.WP2E= 0008            P.WP3E= 0004
P.5VOK= 0002            QUE   = 281B            QUEM  = 281E            RAMT  = 0010
RARA  = 0006            RARAI = 0004            RCHBD0= 0048            RCHBD1= 0047
RCHOK = 0046            RCHTST= 000C            RCLRT = 000D            RCMD  = 000B
RCMLP = 0003            RCONT = 0080            RDATA = 0017            RDCLK = 0010
RDON  = 0011            READG = 0007            REND  = 0014            REQST = 2806
RESCHR= 00D1            REVTST= 0064            REWIND= 0004            RFIFOL= 0008
RGCLK = 0002            RGCRI = 0003            RIBG  = 0001            RILL  = 0012
RINST = 000C            RMCTST= 0008            RMK2  = 0013            RNOP  = 0000
RPATH = 0001            RPBAD = 0044            RPCHI = 0001            RPCLK = 0003
RPCTL = 0009            RPEI  = 0002            RPFAIL= 0000            RPF1  = 009D
RPF2  = 009E            RPOK  = 0043            RPOSTN= 0016            RPSTA = 0015
RRCMT = 000A            RSTAT = 0002            RTIEB = 000A            RTIER = 0030
RTM   = 0005            RUNKI = 0009            RUPTST= 005E            RWDUNL= 0005
R.AMT = 0001            R.BOP = 0008            R.DATA= 0040            R.DON = 0002
```

```
R.DRDY= 0010        R.END = 0010        R.ILL = 0004        R.JVOK= 0004
R.MK2 = 0008        R.PLOD= 0008        R.PL00= 0010        R.PL01= 0020
R.POST= 0020        R.STNM= 0002        R.STOP= 0004        R.STPC= 0001
R.TBJN= 0080        R.TSTD= 0040        R.VOK = 0080        R00H  = 0081
R00L  = 0080        R01H  = 0083        R01L  = 0082        R02H  = 0085
R02L  = 0084        R03H  = 0087        R03L  = 0086        R04H  = 0089
R04L  = 0088        R05H  = 008B        R05L  = 008A        R06H  = 008D
R06L  = 008C        R07H  = 008F        R07L  = 008E        R10H  = 0091
R10L  = 0090        R11H  = 0093        R11L  = 0092        R12H  = 0095
R12L  = 0094        R13H  = 0097        R13L  = 0096        R14H  = 0099
R14L  = 0098        R15H  = 009B        R15L  = 009A        R16H  = 009D
R16L  = 009C        R17H  = 009F        R17L  = 009E        R7.5  = 0010
SELCLR= 0000        SETATA= 00A1        SID   = 0080        SOD   = 0080
SOE   = 0040        SP    =%0008        SSCLK = 0040        SSTEP = 0005
STACK = 4F1F        STATRM= 4F20        STPCT   4F20        STRSP = 5000
SUBCNT  484D        SUNIT   4840        TADR00= 0080        TADR01= 0081
TADR02= 0082        TADR03= 0083        TADR04= 0084        TADR05= 0085
TADR06= 0086        TADR07= 0087        TADR10= 0088        TADR11= 0C89
TADR12= 008A        TADR13= 008B        TAMT  = 0044        TASEL = 0080
TCMD  = 0040        TC.FWD= 0040        TC.INH= 0008        TC.LWR= 0004
TC.REV= 0020        TC.WRT= 0010        TEMP    4F99        TEST1   4300
TEST2   43E0        TMF   = 0099        TMRDY = 0040        TRKENA= 00D2
TSET  = 2803        TSTEND= 2818        TSTS  = 0040        TST01A  43B8
TST01C  4377        TST01X  4351        TST02A  4474        TST02B  447C
TST02C  441A        TST02X  43E5        TUSEL0= 00D1        TUSEL1= 00D2
TU78  = 0010        T.ATH0= 0001        T.ATH1= 0002        T.BOT = 0004
T.EOT = 0002        T.FPT = 0001        T.NTHR= 0004        T.ONL = 0020
T.PES = 0008        T.PSBJ= 0020        T.PS0J= 0008        T.PS1J= 0010
T.RDY = 0080        T.RDY0= 0040        T.RWD = 0010        T.SCLK= 0002
UIBG  = 00A1        UNITMP  483F        UNITSL  4327        VALFC   4F98
VALTB   4F95        VELTST= 005B        WDR.P = 0010        WMCCTL= 00D3
WMCERR= 00DA        WMCSTA= 00D0        WRTCLK= 0000        WRTDAT= 00D3
W.ACKC= 0004        W.CRC = 0008        W.DIAG= 0002        W.DONN= 0040
W.ECC = 0010        W.ENAB= 0080        W.ERR = 0020        W.FMT = 0070
W.GCR = 0010        W.LEFT= 0004        W.ONES= 0020        W.RESI= 0002
W.REV = 0004        W.ROME= 0010        W.RST = 0001        W.SKIP= 000F
W.WRIT= 0008        W.XFER= 0020        X     =%000A        X.DONN= 0080
X.ENAB= 0040        X.PEPE= 0002        X.ROME= 0001        X.WCLK= 0001
Y     =%000B        .     = 484E


 ERRORS DETECTED:  0

 *RPM4.A78/PTP,RPM4=NLIST,PARAM,MACRO,LIST,RPM4
 RUN-TIME: 4 6 0 SECONDS
 CORE USED:  10K
```

.MAIN. CROSS - MICRO PROCESSOR ASSEMBLER 5C(25) 15-OCT-80 09:22
PARAM.M80 TABLE OF 'ONTENTS SEQ 0862

3 'MEMORY MA '' EXTENDED ADDRESS BIT
19 READ MICRO CONTROLLER CONTROL & STATUS WORDS
139 TRANSPORT/PORT COMMAND + STATUS WORDS
380 COMMON ADDRESS SPACE REGISTERS
444 MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485 MASSBUS CONTROL WORD DEFINITIONS
557 WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709 KEYBOARD/DISPLAY CONTROL + STATUS WORD
797 MISCELLANEOUS VARIABLE DEFINITIONS
801 RAM ADDRESS DEFINITIONS
819 LINKAGE TABLE ADDRESSES
842 BIT DEFINITIONS
856 RIM AND SIM INSTRUCTION BIT DEFINITIONS
877 INTERRUPTABLE INSTRUCTION MACROS
924 DIAGNOSTIC VARIABLES
945 COMMON VARIABLE STORAGE
959 MACRO CALL DEFINITIONS
1047 TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101 8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1333 TEST 01 - MARK TWO DETECT
1553 TEST 02 - END MARK DETECT TEST
1738 TEST 03 - FIFO DEPTH TEST
1921 SUBROUTINE CLEAR ALL TU PORTS
1965 SUBROUTINE CHECK MARK 2
2066 SUBROUTINE CHECK END MARK
2170 SUBROUTINE CHECK AMTIE
2312 TABLE INPUT DATA MARK TWO
2325 TABLE INPUT DATA END MARK
2337 PROGRAM VARIABLES

```
;*************************************************************************
;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
;ERROR MACRO CALLS
;
;       1  -    REQUEST HOST CPU TO PRINT:
;               "BYTE/SCLK COUNT NUMBER = LLL"
;               WHERE:
;               LLL = THE VALUE STORED IN CAS REGISTER 5
;                     (THE BYTE COUNT REGISTER 16 BITS).
;
;       2  -    REQUEST HOST CPU TO PRINT:
;               DATA FORMAT = MM
;               SKIP COUNT  = NN
;               WHERE:
;               MM = DATA FORMAT FROM CAS REGISTER 2
;               NN = SKIP COUNT FROM CAS REGISTER 2
;
;       3  -    REQUEST HOST CPU TO PRINT:
;               BYTE-SCLK COUNT = LLL
;               DATA FORMAT = MM
;               SKIP COUNT  = NN
;               WHERE:
;               LLL = AS ABOVE
;               MM  = AS ABOVE
;               NN  = AS ABOVE
;
;       4  -    REQUEST HOST TO PRINT:
;               TRANSITION COUNT = LLL
;               WHERE: LLL = COUNT FROM CAS REGISTER 05
;
;       5  -    REQUEST HOST CPU TO PRINT:
;               EXPECTED 18 BITS =      E EEEEEE
;               ACTUAL 18 BITS =        A AAAAAA
;
;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
;               OF CAS REG 15 LOW BYTE.
;
;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
;               17 LOW BYTE SIGN BIT.
;
;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
;               AND/OR ACTUAL DAT^.
;
;       7  -    REQUEST HOST CPU TO PRINT:
;               "SUBGROUP NUMBER = LLL"
;               WHERE:
;               LLL = THE VALUE STORED IN CAS REGISTER 5
;                     (THE BYTE COUNT REGISTER 16 BITS).
;
;       10 -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
;*************************************************************************
```

```
1016                    ;*****************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ;*****************************************************************
1035                    ;*****************************************************************
1036                    ;DATA PATTERN CODES
1037
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ;*****************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
1332                            .TITLE  RPM5 - READ PATH MICRO CONTROLLER PART #5
1333                            .SBTTL  TEST 01 - MARK TWO DETECT
1334                            ;ID     RPM5-READ PATH MICRO CONTROLLER PART #5
1335  4300                      ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1336                            ;*MARK TWO DETECT
1337  4300                      SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1338                            ;*THIS TEST CHECKS THE DATA HANDSHAKE BETWEEN THE ECC AND READ CHANNELS
1339                            ;*AS WELL AS THE READ CHANNELS ABILITY TO DETECT A MARK TWO TRANSLATION.
1340  4300                      SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1341                            ;*BGNTST
1342                            ;*      SET NORMAL READ PATH CLOCK
1343                            ;*      CALL SUBROUTINE CLEAR
1344                            ;*      SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
1345                            ;*      CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
1346                            ;*      DELAY 50 MICROSECONDS
1347                            ;*      CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
1348                            ;*      CLEAR THE TIE BUS
1349                            ;*      SET UP INPUT TABLE POINTER
1350                            ;*      ENABLE THE READ PATH CLOCK
1351                            ;*      SET PLO BYPASS MODE 2,TIE BUS JAM AND PLO DISABLE
1352                            ;*      CLOCK THE FIFO
1353                            ;*      ISSUE CLEAR ALL COMMAND
1354                            ;*      WAIT
1355                            ;*      STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
1356                            ;*      INIT THE LOOP COUNT TO 5
1357                            ;*      BGNDO
1358                            ;*      :       MOVE 8 BITS OF DATA FROM THE INPUT TABLE TO THE TU PORT COMMAND REGISTER
1359                            ;*      :       INCREMENT THE INPUT TABLE POINTER
1360                            ;*      :       MOVE THE PARITY BIT FROM THE INPUT TABLE TO THE TU PORT CONTROL REGISTER
1361                            ;*      :       CLOCK THE DATA INTO THE FIFO
1362                            ;*      :       DECREMENT THE LOOP COUNT
1363                            ;*      :       INCREMENT THE INPUT TABLE POINTER
1364                            ;*      :       DO UNTIL LOOP COUNT = 0
1365                            ;*      ENDDO
1366                            ;*      ISSUE DIAGNOSTIC READ COMMAND
1367                            ;*      SET UP WATCHDOG TIMER COUNT
1368                            ;*      BGNDO
1369                            ;*      :       SINGLE STEP THE READ PATH
1370                            ;*      :       DECREMENT THE WATCHDOG TIMER
1371                            ;*      :       DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1372                            ;*      ENDDO
1373                            ;*      IF WATCHDOG TIMER=0
1374                            ;*      :       THEN-ERROR
1375                            ;*      :       ELSE-CONTINUE
1376                            ;*      ENDIF
```

```
1377                                ;*    SINGLE STEP THE READ PATH
1378                                ;*    CALL SUBROUTINE CKMK
1379                                ;*    DO UNTIL END OF INPUT DATA TABLE
1380                                ;*ENDTST
1381  4300                          SE
 (1)                                ; ***************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1382                                ;*RPM5 MICRO TEST 01
1383                                ;*RPM5 MICRO ERROR 01
1384                                ;*RPM5-READ PATH-MARK TWO DETECT TEST
1385                                ;*M8950'S, M8953
1386                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1387                                ;*FATAL ERROR - TEST ABORTED
1388                                ;*
1389                                ;*RPM5 MICRO TEST 01
1390                                ;*RPM5 MICRO ERROR 02
1391                                ;*RPM5-READ PATH-MARK TWO DETECT TEST
1392                                ;*M8950'S, M8953
1393                                ;*CHANNEL 0 NOT MARK TWO
1394                                ;*
1395                                ;*RPM5 MICRO TEST 01
1396                                ;*RPM5 MICRO ERROR 03
1397                                ;*RPM5-READ PATH-MARK TWO DETECT TEST
1398                                ;*M8950'S, M8953
1399                                ;*CHANNEL 1 NOT MARK TWO
1400                                ;*
1401                                ;*RPM5 MICRO TEST 01
1402                                ;*RPM4 MICRO ERROR 04
1403                                ;*RPM5-READ PATH-MARK TWO DETECT TEST
1404                                ;*M8950'S, M8953
1405                                ;*CHANNEL 2 NOT MARK TWO
1406                                ;*
1407                                ;*RPM5 MICRO TEST 01
1408                                ;*RPM5 MICRO ERROR 05
1409                                ;*RPM5-READ PATH-MARK TWO DETECT TEST
1410                                ;*M8950'S, M8953
1411                                ;*CHANNEL 3 NOT MARK TWO
1412                                ;*
1413                                ;*RPM5 MICRO TEST 01
1414                                ;*RPM5 MICRO ERROR 06
1415                                ;*RPM5-READ PATH-MARK TWO DETECT TEST
1416                                ;*M8950'S, M8953
1417                                ;*CHANNEL 4 NOT MARK TWO
1418                                ;*
1419                                ;*RPM5 MICRO TEST 01
1420                                ;*RPM5 MICRO ERROR 07
1421                                ;*RPM5-READ PATH-MARK TWO DETECT TEST
1422                                ;*M8950'S, M8953
1423                                ;*CHANNEL 5 NOT MARK TWO
1424                                ;*
1425                                ;*RPM5 MICRO TEST 01
1426                                ;*RPM5 MICRO ERROR 10
1427                                ;*RPM5-READ PATH-MARK TWO DETECT TEST
```

```
                                        ;*M8950'S, M8953
                                        ;*CHANNEL 6 NOT MARK TWO
                                        ;*
                                        ;*RPM5 MICRO TEST 01
                                        ;*RPM5 MICRO ERROR 11
                                        ;*RPM5-READ PATH-MARK TWO DETECT TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL 7 NOT MARK TWO
                                        ;*
                                        ;*RPM5 MICRO TEST 01
                                        ;*RPM5 MICRO ERROR 12
                                        ;*RPM5-READ PATH-MARK TWO DETECT TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL P NOT MARK TWO
4300                                    $
 (1)                                    ; ************************************************************************
                                        TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1) 4300  3E  01          7.0                   MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1) 4302  CD  03  28     18.0                   CALL    TSET            ;SETUP THE TEST
                                        ;%RPM5-READ PATH-MARK TWO DETECT TEST
                                        ;&M8950'S, M8953
4305                                            REQ     @7,0,0,0,0
 (1) 4305  CD  06  28     18.0                   CALL    REQST
 (1) 4308  00                                    .BYTE   0               ;DATA PATTERN NUMBER
 (1) 4309  00  00                                .WORD   0               ;SYSTEM "0" COUNT
 (1) 430B  00  00                                .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1) 430D  00                                    .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1) 430E  07                                    .BYTE   @7              ;REQUEST CODE
430F                                            RIN     R12L
 (1) 430F  DB  94         10.0                   IN      R12L            ;READ R12L INTO AC
 (1) 4311  7F              4.0                   MOV     A,A             ;RETRY LINK
4312  32  78  47          13.0           STA     UNITMP
4315  A7                   4.0           ANA     A
4316  C2  27  43          10.0           JNZ     UNITSL
4319                                    ERR     EXIT,DUMMY
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1) 4319  CD  09  28     18.0                   CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0001                                  MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1) 431C  01                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1) 431D  00                                    .BYTE
 (1) 431E  CD  15  28     18.0           DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1) 4321  DA  0B  45     10.0                   JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                        ;>FATAL ERROR - TEST ABORTED
4324  C3  0B  45          10.0           JMP     EXIT

4327  06  00                7.0   UNITSL: MVI     B,@0            ;LOAD THE BASE UNIT NUMBER
4329  3A  78  47          13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
432C  E6  01                7.0           ANI     @01             ;PORT 0?
432E  C2  44  43          10.0           JNZ     FOUND           ;YES-GO USE PORT #0
4331  04                    4.0           INR     B               ;NO-UPDATE POINTER TO PORT #1
4332  3A  78  47          13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
4335  E6  02                7.0           ANI     @02             ;PORT 1?
```

```
1464  4337  C2  44  43   10.0          JNZ   FOUND          ;YES-GO USE PORT #1
1465  433A  04            4.0          INR   B              ;NO-UPDATE POINTER TO PORT #2
1466  433B  3A  78  47    13.0          LDA   UNITMP         ;DID THE USER SPECIFY TU
1467  433E  E6  04         7.0          ANI   @04            ;PORT 2?
1468  4340  C2  44  43   10.0          JNZ   FOUND          ;YES-GO USE PORT #2
1469  4343  04            4.0          INR   B              ;NO-ASSUME PORT #3
1470  4344  CD  0E  45   18.0  FOUND:  CALL  CLEAR          ;CLEAR ALL THE TU PORTS
1471  4347  DB  E0        10.0          IN    INTSTA         ;GET THE MASS BUS SELECT BIT
1472  4349  E6  80         7.0          ANI   BIT7           ;
1473  434B  B0            4.0          ORA   B              ;OR IN THE TU PORT SELECT BITS
1474  434C  D3  E0        10.0          OUT   MBSEL          ;SELECT THE MASS BUS AND TU PORTS
1475  434E  32  79  47    13.0          STA   SUNIT          ;STORE THE UNIT NUMBER FOR LATER USE
1476
1477  4351  3E  10         7.0  TST01X: MVI   A,RDCLK       ;SET NORMAL READ PATH CLOCKS
1478  4353  D3  F0        10.0          OUT   CLKCTL
1479
1480  4355  3E  04         7.0          MVI   A,P.RPEN              ;ENABLE THE READ PATH FROM THE TU PORT
1481  4357  D3  4C        10.0          OUT   PENAB          ;
1482  4359  3E  10         7.0          MVI   A,W.GCR        ;SET GCR MODE
1483  435B  D3  D3        10.0          OUT   WMCCTL
1484  435D  3E  60         7.0          MVI   A,P.LWR!P.LCS  ;SET THE PORT CONTROL TO
1485  435F  D3  48        10.0          OUT   PDIAG          ;LCS, LWR AND DPEN
1486  4361  AF            4.0          XRA   A
1487  4362  D3  44        10.0          OUT   TAMT           ;
1488  4364  3E  0A         7.0          MVI   A,10           ;DELAY
1489  4366  3D            4.0  1$:     DCR   A              ;50
1490  4367  C2  66  43   10.0          JNZ   1$             ;MICROSECONDS
1491
1492  436A  D3  D2        10.0          OUT   TRKENA         ;CLEAR ALL TRACKS FROM
1493  436C  D3  D2        10.0          OUT   TRKENA         ;THE TRANSLATOR
1494
1495  436E  AF            4.0          XRA   A              ;CLEAR THE TIE BUS
1496  436F  D3  0A        10.0          OUT   RTIEB
1497
1498  4371  21  64  47   10.0          LXI   H,M8950MK       ;SET UP POINTER TO INPUT DATA TABLE
1499  4374  22  7E  47   16.0          SHLD  INTBL          ;SAVE IT
1500  4377  3E  A8         7.0  TST01C: MVI   A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1501                                                       ;TIE BUS JAM AND PLO DISABLE
1502  4379  D3  09        10.0          OUT   RPCTL
1503  437B  D3  08        10.0          OUT   RFIFOL         ;CLOCK THE FIFO
1504  437D  3E  0D         7.0          MVI   A,RCLRT        ;ISSUE CLEAR ALL COMMAND
1505  437F  D3  0B        10.0          OUT   RCMD
1506
1507  4381  00            4.0          NOP                  ;WAIT
1508  4382  00            4.0          NOP
1509  4383  00            4.0          NOP
1510  4384  00            4.0          NOP
1511
1512  4385  3E  A9         7.0          MVI   A,R.PLO1!R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
1513                                                                 ;SET TIE BUS JAM AND PLO DISABLE
1514  4387  D3  09        10.0          OUT   RPCTL
1515
1516  4389  0E  02         7.0          MVI   C,2            ;SET UP THE FIFO LOAD COUNTER
1517  438B  2A  7E  47   16.0  3$:     LHLD  INTBL          ;GET POINTER TO INPUT DATA TABLE
```

```
1518  438E  06  05        7.0        MVI   B,@5          ;SET UP LOOP COUNT
1519  4390  7E            7.0  1$:   MOV   A,M           ;GET A DATA BYTE
1520  4391  D3  40       10.0        OUT   TCMD          ;STORE DATA IN COMMAND ADDRESS
1521  4393  23            6.0        INX   H             ;POINT TO DATA PARITY
1522  4394  7E            7.0        MOV   A,M           ;GET THE DATA PARITY
1523  4395  07            4.0        RLC                 ;POSITION FOR OUTPUT
1524  4396  F6  60        7.0        ORI   P.LWR!P.LCS   ;OR IN CONTROL BITS
1525  4398  D3  48       10.0        OUT   PDIAG         ;OUTPUT THE DATA PARITY
1526  439A  D3  08       10.0        OUT   RFIFOL        ;CLOCK DATA INTO THE FIFO'S
1527
1528  439C  23            6.0        INX   H             ;UPDATE THE TABLE POINTER
1529  439D  05            4.0        DCR   B             ;DECREMENT LOOP COUNT
1530  439E  C2  90  43   10.0        JNZ   1$            ;DO UNTIL LOOP COUNT = 0
1531
1532  43A1  0D            4.0        DCR   C             ;DECREMENT THE FIFO LOAD LOOP COUNTER
1533  43A2  C2  8B  43   10.0        JNZ   3$            ;CONTINUE UNTIL ZERO
1534                              ;THE FIFO IS NOW FILLED WITH THE INPUT DATA TABLE
1535
1536  43A5  22  7E  47   16.0        SHLD  INTBL         ;SAVE THE INPUT DATA TABLE POINTER
1537  43A8  3E  0B        7.0        MVI   A,DIARD       ;LOAD THE DIAGNOSTIC READ
1538  43AA  D3  0B       10.0        OUT   RCMD          ;COMMAND
1539  43AC  11  01  00   10.0        LXI   D,1           ;SET WATCH DOG INCREMENT
1540  43AF  21  A8  FD   10.0        LXI   H,-600        ;SET WATCH DOG COUNT TO 600
1541  43B2  D3  0C       10.0  2$:   OUT   RINST         ;STEP THE READ PATH
1542  43B4  DB  01       10.0        IN    RPCHI         ;GET THE READ PATH STATUS
1543  43B6  E6  10        7.0        ANI   R.DRDY        ;DATA READY SET?
1544  43B8  C2  BF  43   10.0        JNZ   TST01A        ;YES - GO CHECK
1545  43BB  19           10.0        DAD   D             ;WATCH DOG TIMEOUT?
1546  43BC  D2  B2  43   10.0        JNC   2$            ;NO-CONTINUE
1547  43BF  D3  0C       10.0  TST01A: OUT   RINST       ;SINGLE STEP THE READ PATH
1548  43C1  CD  39  45   18.0        CALL  CKMK          ;CHECK FOR FAILING READ CHANNELS
1549  43C4  DA  51  43   10.0        JC    TST01X        ;LOOP ON ERROR
1550  43C7                        ENDTST  TST01X
 (1)                          ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43C7                        REQ     7              ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43C7  CD  06  28   18.0                    CALL    REQST
 (2)  43CA  00                                   .BYTE                   ;DATA PATTERN NUMBER
 (2)  43CB  00  00                               .WORD                   ;SYSTEM '*' COUNT
 (2)  43CD  00  00                               .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43CF  00                                   .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  43D0  07                                   .BYTE   7               ;REQUEST CODE
 (1)  43D1  3A  9A  4F   13.0        LDA   ITERA         ;GET ITERATION COUNT
 (1)  43D4  3D           4.0        DCR   A             ;DOWNCOUNT
 (1)  43D5  32  9A  4F   13.0        STA   ITERA         ;SAVE COUNT
 (1)  43D8  F2  51  43   10.0        JP    TST01X        ;DO TEST UNTIL TILL = 0
1551  43DB  CD  0E  45   18.0        CALL  CLEAR         ;CLEAR ALL TU PORTS
```

```
                                .SBTTL  TEST 02 - END MARK DETECT TEST
43DE                            ST
                                ;*****************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*END MARK DETECT TEST
43DE                            SD
                                ;*****************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE DATA HANDSHAKE BETWEEN THE ECC AND READ CHANNELS
                                ;*AS WELL AS THE READ CHANNELS ABILITY TO DETECT AN END MARK TRANSLATION.
43DE                            SP
                                ;*****************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*     SET NORMAL READ PATH CLOCK
                                ;*     CALL SUBROUTINE CLEAR
                                ;*     SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
                                ;*     CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
                                ;*     DELAY 50 MICROSECONDS
                                ;*     CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
                                ;*     CLEAR THE TIE BUS
                                ;*     SET UP INPUT TABLE POINTER
                                ;*     ENABLE THE READ PATH CLOCK
                                ;*     SET PLO BYPASS MODE 2,TIE BUS JAM AND PLO DISABLE
                                ;*     CLOCK THE FIFO
                                ;*     ISSUE CLEAR ALL COMMAND
                                ;*     WAIT
                                ;*     STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
                                ;*     INIT THE LOOP COUNT TO 5
                                ;*     BGNDO
                                ;*     :     MOVE 8 BITS OF DATA FROM THE INPUT TABLE TO THE TU PORT COMMAND REGISTER
                                ;*     :     INCREMENT THE INPUT TABLE POINTER
                                ;*     :     MOVE THE PARITY BIT FROM THE INPUT TABLE TO THE TU PORT CONTROL REGISTER
                                ;*     :     CLOCK THE DATA INTO THE FIFO
                                ;*     :     DECREMENT THE LOOP COUNT
                                ;*     :     INCREMENT THE INPUT TABLE POINTER
                                ;*     :     DO UNTIL LOOP COUNT = 0
                                ;*     ENDDO
                                ;*     ISSUE DIAGNOSTIC READ COMMAND
                                ;*     SET UP WATCHDOG TIMER COUNT
                                ;*     BGNDO
                                ;*     :     SINGLE STEP THE READ PATH
                                ;*     :     DECREMENT THE WATCHDOG TIMER
                                ;*     :     DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
                                ;*     ENDDO
                                ;*     IF WATCHDOG TIMER=0
                                ;*     :     THEN-ERROR
                                ;*     :     ELSE-CONTINUE
                                ;*     ENDIF
                                ;*     SINGLE STEP THE READ PATH
                                ;*     CALL SUBROUTINE CKEMK
```

```
1598                                    ;*    DO UNTIL END OF INPUT DATA TABLE
1599                                    ;*ENDTST
1600  43DE                              SE
 (1)                                    ; ***************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1601                                    ;*RPM5 MICRO TEST 02
1602                                    ;*RPM5 MICRO ERROR 13
1603                                    ;*RPM5-READ PATH-END MARK DETECT TEST
1604                                    ;*M8950'S, M8953
1605                                    ;*CHANNEL 0 NOT END MARK
1606                                    ;*
1607                                    ;*RPM5 MICRO TEST 02
1608                                    ;*RPM5 MICRO ERROR 14
1609                                    ;*RPM5-READ PATH-END MARK DETECT TEST
1610                                    ;*M8950'S, M8953
1611                                    ;*CHANNEL 1 NOT END MARK
1612                                    ;*
1613                                    ;*RPM5 MICRO TEST 02
1614                                    ;*RPM5 MICRO ERROR 15
1615                                    ;*RPM5-READ PATH-END MARK DETECT TEST
1616                                    ;*M8950'S, M8953
1617                                    ;*CHANNEL 2 NOT END MARK
1618                                    ;*
1619                                    ;*RPM5 MICRO TEST 02
1620                                    ;*RPM5 MICRO ERROR 16
1621                                    ;*RPM5-READ PATH-END MARK DETECT TEST
1622                                    ;*M8950'S, M8953
1623                                    ;*CHANNEL 3 NOT END MARK
1624                                    ;*
1625                                    ;*RPM5 MICRO TEST 02
1626                                    ;*RPM5 MICRO ERROR 17
1627                                    ;*RPM5-READ PATH-END MARK DETECT TEST
1628                                    ;*M8950'S, M8953
1629                                    ;*CHANNEL 4 NOT END MARK
1630                                    ;*
1631                                    ;*RPM5 MICRO TEST 02
1632                                    ;*RPM5 MICRO ERROR 20
1633                                    ;*RPM5-READ PATH-END MARK DETECT TEST
1634                                    ;*M8950'S, M8953
1635                                    ;*CHANNEL 5 NOT END MARK
1636                                    ;*
1637                                    ;*RPM5 MICRO TEST 02
1638                                    ;*RPM5 MICRO ERROR 21
1639                                    ;*RPM5-READ PATH-END MARK DETECT TEST
1640                                    ;*M8950'S, M8953
1641                                    ;*CHANNEL 6 NOT END MARK
1642                                    ;*
1643                                    ;*RPM5 MICRO TEST 02
1644                                    ;*RPM5 MICRO ERROR 22
1645                                    ;*RPM5-READ PATH-END MARK DETECT TEST
1646                                    ;*M8950'S, M8953
1647                                    ;*CHANNEL 7 NOT END MARK
1648                                    ;*
```

```
                                        ;*RPM5 MICRO TEST 02
                                        ;*RPM5 MICRO ERROR 23
                                        ;*RPM5-READ PATH-END MARK DETECT TEST
                                        ;*M8950'S, M8953
                                        ;*CHANNEL P NOT END MARK
43DE                                    $
 (1)                                    ; ************************************************************************************
                                        TEST2:  TESTX   @2              ;INITIALIZE THE TEST
 (1) 43DE  3E  02           7.0                  MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1) 43E0  CD  03  28      18.0                  CALL    TSET            ;SETUP THE TEST
                                        ;%RPM5-READ PATH-END MARK DETECT TEST
                                        ;&M8950'S, M8953

43E3  3A  79  47           13.0         LDA     SUNIT
43E6  D3  E0               10.0         OUT     MBSEL
43E8  3E  10                7.0  TST02X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
43EA  D3  F0               10.0         OUT     CLKCTL

43EC  3E  04                7.0         MVI     A,P.RPEN                ;ENABLE THE READ PATH FROM THE TU PORT
43EE  D3  4C               10.0         OUT     PENAB           ;
43F0  3E  10                7.0         MVI     A,W.GCR         ;SET GCR MODE
43F2  D3  D3               10.0         OUT     WMCCTL
43F4  3E  60                7.0         MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
43F6  D3  48               10.0         OUT     PDIAG           ;LCS, LWR AND DPEN
43F8  AF                    4.0         XRA     A
43F9  D3  44               10.0         OUT     TAMT            ;
43FB  3E  0A                7.0         MVI     A,10            ;DELAY
43FD  3D                    4.0  1$:     DCR     A               ;50
43FE  C2  FD  43           10.0         JNZ     1$              ;MICROSECONDS

4401  D3  D2               10.0         OUT     TRKENA          ;CLEAR ALL TRACKS FROM
4403  D3  D2               10.0         OUT     TRKENA          ;THE TRANSLATOR

4405  AF                    4.0         XRA     A               ;CLEAR THE TIE BUS
4406  D3  0A               10.0         OUT     RTIEB

4408  21  6E  47           10.0         LXI     H,M8950EM       ;SET UP POINTER TO INPUT DATA TABLE
440B  22  7E  47           16.0         SHLD    INTBL           ;SAVE IT
440E  3E  A8                7.0  TST02C: MVI     A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
                                                                ;TIE BUS JAM AND PLO DISABLE
4410  D3  08               10.0         OUT     RFIFOL          ;CLOCK THE FIFO
4412  D3  09               10.0         OUT     RPCTL
4414  3E  0D                7.0         MVI     A,RCLRT         ;ISSUE CLEAR ALL COMMAND
4416  D3  0B               10.0         OUT     RCMD

4418  00                    4.0         NOP                     ;WAIT
4419  00                    4.0         NOP
441A  00                    4.0         NOP
441B  00                    4.0         NOP

441C  3E  A9                7.0         MVI     A,R.PLO1!R.STPC.R.PLOD!R.TBJN   ;STOP THE READ PATH
                                                                ;SET TIE BUS JAM AND PLO DISABLE
441E  D3  09               10.0         OUT     RPCTL
```

```
1700
1701  4420  0E  02        7.0          MVI   C,2              ;SET UP THE FIFO LOAD COUNTER
1702  4422  2A  7E  47    16.0  3$:    LHLD  INTBL            ;GET POINTER TO INPUT DATA TABLE
1703  4425  06  05        7.0          MVI   B,@5             ;SET UP LOOP COUNT
1704  4427  7E            7.0  1$:    MOV   A,M              ;GET A DATA BYTE
1705  4428  D3  40       10.0          OUT   TCMD             ;STORE DATA IN COMMAND ADDRESS
1706  442A  23            6.0          INX   H                ;POINT TO DATA PARITY
1707  442B  7E            7.0          MOV   A,M              ;GET THE DATA PARITY
1708  442C  07            4.0          RLC                    ;POSITION FOR OUTPUT
1709  442D  F6  60        7.0          ORI   P.LWR!P.LCS  ;OR IN CONTROL BITS
1710  442F  D3  48       10.0          OUT   PDIAG            ;OUTPUT THE DATA PARITY
1711  4431  D3  08       10.0          OUT   RFIFOL           ;CLOCK DATA INTO THE FIFO'S
1712
1713  4433  23            6.0          INX   H                ;UPDATE THE TABLE POINTER
1714  4434  05            4.0          DCR   B                ;DECREMENT LOOP COUNT
1715  4435  C2  27  44    10.0          JNZ   1$               ;DO UNTIL LOOP COUNT = 0
1716
1717  4438  0D            4.0          DCR   C                ;DECREMENT THE FIFO LOAD LOOP COUNTER
1718  4439  C2  22  44    10.0          JNZ   3$               ;CONTINUE UNTIL ZERO
1719                                ;THE FIFO IS NOW FILLED WITH THE INPUT DATA TABLE
1720
1721  443C  22  7E  47    16.0          SHLD  INTBL            ;SAVE THE INPUT DATA TABLE POINTER
1722  443F  3E  0B        7.0          MVI   A,DIARD          ;LOAD THE DIAGNOSTIC READ
1723  4441  D3  0B       10.0          OUT   RCMD             ;COMMAND
1724  4443  11  01  00    10.0          LXI   D,1              ;SET WATCH DOG INCREMENT
1725  4446  21  A8  FD    10.0          LXI   H,-600           ;SET WATCH DOG COUNT TO 600
1726  4449  D3  0C       10.0  2$:    OUT   RINST            ;STEP THE READ PATH
1727  444B  DB  01       10.0          IN    RPCHI            ;GET THE READ PATH STATUS
1728  444D  E6  10        7.0          ANI   R.DRDY           ;DATA READY SET?
1729  444F  C2  56  44    10.0          JNZ   TST02A           ;YES - GO CHECK
1730  4452  19            10.0          DAD   D                ;WATCH DOG TIMEOUT?
1731  4453  D2  49  44    10.0          JNC   2$               ;NO-CONTINUE
1732  4456  D3  0C       10.0  TST02A: OUT   RINST            ;SINGLE STEP THE READ PATH
1733  4458  CD  DC  45    18.0          CALL  CKEMK            ;CHECK FOR FAILING READ CHANNELS
1734  445B  DA  E8  43    10.0          JC    TST02X           ;LOOP ON ERROR
1735  445E                              ENDTST TST02X
 (1)                            ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  445E                              REQ   7                ;FAKE CALL TO KEEP TEST ALIVE
 (2)  445E  CD  06  28    18.0                        CALL    REQST
 (2)  4461  00                                        .BYTE                   ;DATA PATTERN NUMBER
 (2)  4462  00  00                                    .WORD                   ;SYSTEM '"' COUNT
 (2)  4464  00  00                                    .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4466  00                                        .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4467  07                                        .BYTE   7               ;REQUEST CODE
 (1)  4468  3A  9A  4F    13.0          LDA   ITERA            ;GET ITERATION COUNT
 (1)  446B  3D            4.0          DCR   A                ;DOWNCOUNT
 (1)  446C  32  9A  4F    13.0          STA   ITERA            ;SAVE COUNT
 (1)  446F  F2  E8  43    10.0          JP    TST02X           ;DO TEST UNTIL TILL = 0
1736  4472  CD  0E  45    18.0          CALL  CLEAR            ;CLEAR ALL TU PORTS
```

```
                                .SBTTL  TEST 03 - FIFO DEPTH TEST
                               ST
                               ; ***************************************************************************
                               ;*TEST TITLE
                               ;*---- -----
                               ;*FIFO DEPTH TEST
                               SD
                               ; ***************************************************************************
                               ;*DESCRIPTION
                               ;*-----------
                               ;*THIS TEST IS DESIGNED TO PROVIDE A COMPLETE TEST OF THE FIFO DEPTH TO
                               ;*INSURE THAT THE READ CHANNELS CAN RUN WITH THEIR SPECIFIED SKEW VALUE.
                               SP
                               ; ***************************************************************************
                               ;*PROCEDURE
                               ;*---------
                               ;*BGNTST
                               ;*  SET NORMAL READ PATH CLOCKS
                               ;*  CLEAR THE EXPECTED DATA AND EXPECTED DATA PARITY
                               ;*  SET PLO BYPASS MODE 3,TIE BUS JAM AND PLO DISABLE
                               ;*  CLOCK THE FIFO
                               ;*  ISSUE CLEAR ALL FIFO'S COMMAND
                               ;*  WAIT
                               ;*  BGNDO
                               ;*  :  SET UP FIFO DATA OF ZERO, TIE BUS JAM AND PLO DISABLE
                               ;*  :  CLOCK THE DATA INTO THE FIFO
                               ;*  :  SET UP FIFO DATA OF ONE, TIE BUS JAM AND PLO DISABLE
                               ;*  :  CLOCK THE DATA INTO THE FIFO
                               ;*  :  DECREMENT THE LOOP COUNT
                               ;*  :  DO UNTIL THE LOOP COUNT=0
                               ;*  ENDDO
                               ;*  CLEAR THE LOOP COUNT
                               ;*  BGNDO
                               ;*  :  PUT THE TM78 IN PE MODE
                               ;*  :  ISSUE THE FIFO READ COMMAND
                               ;*  :  LET THE RMC RUN, SET TIE BUS JAM AND PLO DISABLE
                               ;*  :  WAIT
                               ;*  :  STOP THE RMC, SET TIE BUS JAM AND PLO DISABLE
                               ;*  :  INPUT THE FIFO DATA
                               ;*  :  INPUT THE FIFO DATA PARITY
                               ;*  :  CALL SUBROUTINE CKAMT
                               ;*  :  INCREMENT THE LOOP COUNT
                               ;*  :  DO UNTIL THE LOOP COUNT=62(10)
                               ;*  ENDDO
                               ;*ENDTST
                               SE
                               ; ***************************************************************************
                               ;*ERRORS
                               ;*------
                               ;*RPM5 MICRO TEST 03
                               ;*RPM5 MICRO ERROR 24
                               ;*RPM5-FIFO DEPTH TEST
                               ;*M8950, M8953
                               ;*CHANNEL 0 FAILED
```

```
1780                    ;*ACTUAL = NNNN
1781                    ;*EXPECTED = NNNN
1782                    ;*
1783                    ;*RPM5 MICRO TEST 03
1784                    ;*RPM5 MICRO ERROR 25
1785                    ;*RPM5-FIFO DEPTH TEST
1786                    ;*M8950, M8953
1787                    ;*CHANNEL 1 FAILED
1788                    ;*ACTUAL = NNNN
1789                    ;*EXPECTED = NNNN
1790                    ;*
1791                    ;*RPM5 MICRO TEST 03
1792                    ;*RPM5 MICRO ERROR 26
1793                    ;*RPM5-FIFO DEPTH TEST
1794                    ;*M8950, M8953
1795                    ;*CHANNEL 2 FAILED
1796                    ;*ACTUAL = NNNN
1797                    ;*EXPECTED = NNNN
1798                    ;*
1799                    ;*RPM5 MICRO TEST 03
1800                    ;*RPM5 MICRO ERROR 27
1801                    ;*RPM5-FIFO DEPTH TEST
1802                    ;*M8950, M8953
1803                    ;*CHANNEL 3 FAILED
1804                    ;*ACTUAL = NNNN
1805                    ;*EXPECTED = NNNN
1806                    ;*
1807                    ;*RPM5 MICRO TEST 03
1808                    ;*RPM5 MICRO ERROR 30
1809                    ;*RPM5-FIFO DEPTH TEST
1810                    ;*M8950, M8953
1811                    ;*CHANNEL 4 FAILED
1812                    ;*ACTUAL = NNNN
1813                    ;*EXPECTED = NNNN
1814                    ;*
1815                    ;*RPM5 MICRO TEST 03
1816                    ;*RPM5 MICRO ERROR 31
1817                    ;*RPM5-FIFO DEPTH TEST
1818                    ;*M8950, M8953
1819                    ;*CHANNEL 5 FAILED
1820                    ;*ACTUAL = NNNN
1821                    ;*EXPECTED = NNNN
1822                    ;*
1823                    ;*RPM5 MICRO TEST 03
1824                    ;*RPM5 MICRO ERROR 32
1825                    ;*RPM5-FIFO DEPTH TEST
1826                    ;*M8950, M8953
1827                    ;*CHANNEL 6 FAILED
1828                    ;*ACTUAL = NNNN
1829                    ;*EXPECTED = NNNN
1830                    ;*
1831                    ;*RPM5 MICRO TEST 03
1832                    ;*RPM5 MICRO ERROR 33
1833                    ;*RPM5-FIFO DEPTH TEST
```

```
                                        ;*M8950, M8953
                                        ;*CHANNEL 7 FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*RPM5 MICRO TEST 03
                                        ;*RPM5 MICRO ERROR 34
                                        ;*RPM5-FIFO DEPTH TEST
                                        ;*M8950, M8953
                                        ;*CHANNEL P FAILED
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
1847  4475                              S
 (1)                                    ; ************************************************************************************
1848  4475                              TEST3:  TESTX   @3              ;INITIALIZE THE TEST
 (1)  4475  3E  03            7.0                 MVI     A,@3               ;DEFINE THE TEST NUMBER
 (1)  4477  CD  03  28       18.0                 CALL    TSET               ;SETUP THE TEST
1849                                    ;%RPM5-FIFO DEPTH TEST
1850                                    ;&M8950, M8953
1851
1852  447A  3E  10               7.0   TST03X: MVI     A,RDCLK            ;SET NORMAL READ PATH CLOCKS
1853  447C  D3  F0              10.0           OUT     CLKCTL
1854  447E  AF                   4.0           XRA     A                  ;CLEAR
1855  447F  32  87  47           13.0          STA     PASCNT             ;PASS COUNT
1856  4482  32  84  47           13.0          STA     DATAE              ;EXPECTED DATA
1857  4485  32  85  47           13.0          STA     DATAEP             ;EXPECTED DATA PARITY
1858  4488                                    ROUT    RO5H               ;CAS REGISTER 05 HIGH
 (1)  4488  D3  8B              10.0                   OUT     RO5H               ;WRITE AC INTO RO5H
 (1)  448A  7F                   4.0                   MOV     A,A                ;RETRY LINK
1859  448B  3E  B8               7.0           MVI     A,R.PLO1!R.PLO0!R.PLOD!R.TBJN   ;SET PLO BYPASS MODE 3
1860                                                                           ;AND TIE BUS JAM AND PLO DISABLE
1861  448D  D3  09              10.0           OUT     RPCTL              ;
1862  448F  D3  08              10.0           OUT     RFIFOL             ;CLOCK THE FIFO
1863  4491  3E  0D               7.0           MVI     A,RCLRT            ;CLEAR THE FIFO'S
1864  4493  D3  0B              10.0           OUT     RCMD
1865  4495  00                   4.0           NOP                        ;DELAY
1866  4496  00                   4.0           NOP                        ;
1867  4497  00                   4.0           NOP                        ;
1868  4498  00                   4.0           NOP                        ;
1869  4499  06  20               7.0           MVI     B,32               ;LOAD THE LOOP COUNT
1870  449B  3E  B9               7.0   2$:     MVI     A,R.PLO0!R.PLO1!R.STPC!R.PLOD!R.TBJN  ;LOAD FIFO DATA OF ZERO
1871                                                                           ;AND SET TIE BUS JAM AND PLO DISABLE
1872  449D  D3  09              10.0           OUT     RPCTL
1873  449F  D3  08              10.0           OUT     RFIFOL             ;CLOCK THE DATA INTO THE FIFO
1874  44A1  3E  F9               7.0           MVI     A,R.PLO0!R.PLO1!R.STPC!$C8  ;LOAD FIFO DATA ONE
1875                                                                           ;AND SET TIE BUS JAM AND PLO DISABLE
1876  44A3  D3  09              10.0           OUT     RPCTL
1877  44A5  D3  08              10.0           OUT     RFIFOL             ;CLOCK ONES INTO THE FIFO
1878  44A7  05                   4.0           DCR     B                  ;DECREMENT THE LOOP COUNT
1879  44A8  C2  9B  44           10.0          JNZ     2$                 ;CONTINUE UNTIL ZERO
1880  44AB  3A  87  47           13.0   3$:    LDA     PASCNT             ;LOAD THE PASS # IN CAS
1881  44AE                                    ROUT    RO5L
 (1)  44AE  D3  8A              10.0                   OUT     RO5L               ;WRITE AC INTO RO5L
```

```
 (1)  44B0  7F                  4.0                 MOV     A,A             ;RETRY LINK
1882  44B1  AF                  4.0         XRA     A               ;PUT THE MACHINE IN PE MODE
1883  44B2  D3  D3              10.0         OUT     WMCCTL
1884  44B4  3E  6A               7.0         MVI     A,FIFORD        ;LOAD THE FIFO READ COMMAND
1885  44B6  D3  0B              10.0         OUT     RCMD
1886  44B8  3E  B8               7.0         MVI     A,R.PLOO!R.PLO1.R.PLOD!R.TBJN   ;RUN THE RMC
1887                                                                 ;AND SET TIE BUS JAM AND PLO DISABLE
1888  44BA  D3  09              10.0         OUT     RPCTL
1889  44BC  00                   4.0         NOP                     ;WAIT
1890  44BD  00                   4.0         NOP                     ;
1891  44BE  00                   4.0         NOP                     ;
1892  44BF  00                   4.0         NOP                     ;
1893  44C0  3E  B9               7.0         MVI     A,R.PLOO.R.PLO1.R.STPC!R.PLOD!R.TBJN  ;STOP THE RMC
1894                                                                 ;AND SET TIE BUS JAM AND PLO DISABLE
1895  44C2  D3  09              10.0         OUT     RPCTL
1896  44C4  DB  17              10.0         IN      RDATA
1897  44C6  32  82  47          13.0         STA     DATAA
1898  44C9  DB  15              10.0         IN      RPSTA
1899  44CB  E6  40               7.0         ANI     R.DATA
1900  44CD  07                   4.0         RLC
1901  44CE  07                   4.0         RLC
1902  44CF  32  83  47          13.0         STA     DATAAP
1903  44D2  CD  7F  46          18.0         CALL    CKAMT
1904  44D5  DA  7A  44          10.0         JC      TST03X
1905
1906  44D8  3A  87  47          13.0         LDA     PASCNT
1907  44DB  3C                   4.0         INR     A
1908  44DC  32  87  47          13.0         STA     PASCNT
1909  44DF  3A  84  47          13.0         LDA     DATAE
1910  44E2  EE  FF               7.0         XRI     $FF
1911  44E4  32  84  47          13.0         STA     DATAE
1912  44E7  3A  85  47          13.0         LDA     DATAEP
1913  44EA  EE  01               7.0         XRI     $01
1914  44EC  32  85  47          13.0         STA     DATAEP
1915  44EF  3A  87  47          13.0         LDA     PASCNT
1916  44F2  FE  3E               7.0         CPI     62
1917  44F4  C2  AB  44          10.0         JNZ     3$
1918  44F7                                   ENDTST  TST03X
 (1)                                 ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  44F7                                   REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  44F7  CD  06  28          18.0                                 CALL    REQST
 (2)  44FA  00                                                       .BYTE                   ;DATA PATTERN NUMBER
 (2)  44FB  00  00                                                   .WORD                   ;SYSTEM "" COUNT
 (2)  44FD  00  00                                                   .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  44FF  00                                                       .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4500  07                                                       .BYTE   7               ;REQUEST CODE
 (1)  4501  3A  9A  4F          13.0                 LDA     ITERA           ;GET ITERATION COUNT
 (1)  4504  3D                   4.0                 DCR     A               ;DOWNCOUNT
 (1)  4505  32  9A  4F          13.0                 STA     ITERA           ;SAVE COUNT
 (1)  4508  F2  7A  44          10.0                 JP      TST03X          ;DO TEST UNTIL TILL = 0
1919  450B  C3  18  28          10.0         EXIT:   JMP     TSTEND
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ; ************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ; ************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ; ************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*    CLEAR PORT SELECT FOR TRANSPORT
                                        ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*    CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ; ************************************************************************
                        12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
                        12.0          PUSH    B
                         7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
                        10.0  CLPLP:  IN      INTSTA          ;GET MB SELECT INFO
                         7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
                         4.0          ORA     B               ;ADD IN THE SELECTED PORT #
                        10.0          OUT     MBSEL           ;RESET TO THIS PORT
                         7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
                        10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
                         4.0          XRA     A               ;CLEAR TU COMMAND A
                        10.0          OUT     TCMD            ;
                         7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
                        10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
                         7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
                        10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
                         4.0          XRA     A
                        10.0          OUT     TAMT            ;CLEAR AMTIE WORD
                        10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
                        10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
                         4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
                         4.0          MOV     A,B
                         7.0          CPI     4               ;DONE?
                        10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
                        10.0          POP     B               ;RESET B & C
                        10.0          POP     PSW             ;ALL DONE
                        10.0          RET                     ;EXIT
```

```
                                .SBTTL  SUBROUTINE CHECK MARK 2
4539                            SSUB
                                ;*************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*CHECK READ CHANNELS MARK 2
4539                            SD
                                ;*************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO CHECK THAT ALL READ CHANNEL "MARK TWO"
                                ;*BITS EQUAL ONE AND TO FLAG ANY EXCEPTIONS WITH AN ERROR MESSAGE.
4539                            SP
                                ;*************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*  IF CHANNEL 0=MARK TWO
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 0
                                ;*  ENDIF
                                ;*  IF CHANNEL 1=MARK TWO
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 1
                                ;*  ENDIF
                                ;*  IF CHANNEL 2=MARK TWO
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 2
                                ;*  ENDIF
                                ;*  IF CHANNEL 3=MARK TWO
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 3
                                ;*  ENDIF
                                ;*  IF CHANNEL 4=MARK TWO
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 4
                                ;*  ENDIF
                                ;*  IF CHANNEL 5=MARK TWO
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 5
                                ;*  ENDIF
                                ;*  IF CHANNEL 6=MARK TWO
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 6
                                ;*  ENDIF
                                ;*  IF CHANNEL 7=MARK TWO
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 7
                                ;*  ENDIF
                                ;*  IF CHANNEL P=MARK TWO
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL P
                                ;*  ENDIF
                                ;*ENDSUB
```

```
2010  4539                                  S
 (1)                                        ; ************************************************************************
2011
2012  4539  DB  13          10.0  CKMK:    IN      RMK2            ;GET THE READ CHANNEL ILLEGAL BITS
2013  453B  E6  01           7.0          ANI     $01             ;CHANNEL 0 ILLEGAL?
2014  453D  C2  45  45       10.0         JNZ     CKMK0           ;YES-CONTINUE
2015  4540                               ERR     OUTMK2,CKMK0
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4540  CD  09  28       18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4543  02                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4544  00                           .BYTE
 (1)  4545  CD  15  28       18.0  CKMK0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4548  DA  DB  45       10.0         JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2016                          ;>CHANNEL 0 NOT MARK TWO
2017
2018  454B  DB  13          10.0          IN      RMK2            ;GET THE READ CHANNEL ILLEGAL BITS
2019  454D  E6  02           7.0          ANI     $02             ;CHANNEL 1 ILLEGAL?
2020  454F  C2  57  45       10.0         JNZ     CKMK1           ;YES-CONTINUE
2021  4552                               ERR     OUTMK2,CKMK1
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4552  CD  09  28       18.0                 CALL    ERLP            ,PROCESS ERROR - DO 2.3
 (1)       0003                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4555  03                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4556  00                           .BYTE
 (1)  4557  CD  15  28       18.0  CKMK1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  455A  DA  DB  45       10.0         JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2022                          ;>CHANNEL 1 NOT MARK TWO
2023
2024  455D  DB  13          10.0          IN      RMK2            ;GET THE READ CHANNEL ILLEGAL BITS
2025  455F  E6  04           7.0          ANI     $04             ;CHANNEL 2 ILLEGAL?
2026  4561  C2  69  45       10.0         JNZ     CKMK2           ;YES-CONTINUE
2027  4564                               ERR     OUTMK2,CKMK2
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4564  CD  09  28       18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0004                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4567  04                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4568  00                           .BYTE
 (1)  4569  CD  15  28       18.0  CKMK2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  456C  DA  DB  45       10.0         JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2028                          ;>CHANNEL 2 NOT MARK TWO
2029  456F  DB  13          10.0          IN      RMK2            ;GET THE READ CHANNEL ILLEGAL BITS
2030  4571  E6  08           7.0          ANI     $08             ;CHANNEL 3 ILLEGAL?
2031  4573  C2  7B  45       10.0         JNZ     CKMK3           ;YES-CONTINUE
2032  4576                               ERR     OUTMK2,CKMK3
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4576  CD  09  28       18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0005                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4579  05                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  457A  00                           .BYTE
 (1)  457B  CD  15  28       18.0  CKMK3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  457E  DA  DB  45       10.0         JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2033                          ;>CHANNEL 3 NOT MARK TWO
```

```
2035  4581  DB  13        10.0          IN      RMK2            ;GET THE READ CHANNEL ILLEGAL BITS
2036  4583  E6  10         7.0          ANI     $10             ;CHANNEL 4 ILLEGAL?
2037  4585  C2  8D  45    10.0          JNZ     CKMK4           ;YES-CONTINUE
2038  4588                               ERR     OUTMK2,CKMK4
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4588  CD  09  28    18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0006                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  458B  06                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  458C  00                                        .BYTE
 (1)  458D  CD  15  28    18.0                CKMK4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4590  DA  DB  45    10.0                        JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2039                            ;>CHANNEL 4 NOT MARK TWO
2040
2041  4593  DB  13        10.0          IN      RMK2            ;GET THE READ CHANNEL ILLEGAL BITS
2042  4595  E6  20         7.0          ANI     $20             ;CHANNEL 5 ILLEGAL?
2043  4597  C2  9F  45    10.0          JNZ     CKMK5           ;YES-CONTINUE
2044  459A                               ERR     OUTMK2,CKMK5
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  459A  CD  09  28    18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0007                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  459D  07                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  459E  00                                        .BYTE
 (1)  459F  CD  15  28    18.0                CKMK5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45A2  DA  DB  45    10.0                        JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2045                            ·>CHANNEL 5 NOT MARK TWO
2046  45A5  DB  13        10.0          IN      RMK2            ;GET THE READ CHANNEL ILLEGAL BITS
2047  45A7  E6  40         7.0          ANI     $40             ;CHANNEL 6 ILLEGAL?
2048  45A9  C2  B1  45    10.0          JNZ     CKMK6           ;YES-CONTINUE
2049  45AC                               ERR     OUTMK2,CKMK6
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45AC  CD  09  28    18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0008                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45AF  08                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45B0  00                                        .BYTE
 (1)  45B1  CD  15  28    18.0                CKMK6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45B4  DA  DB  45    10.0                        JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2050                            ;>CHANNEL 6 NOT MARK TWO
2051
2052  45B7  DB  13        10.0          IN      RMK2            ;GET THE READ CHANNEL ILLEGAL BITS
2053  45B9  E6  80         7.0          ANI     $80             ;CHANNEL 7 ILLEGAL?
2054  45BB  C2  C3  45    10.0          JNZ     CKMK7           ;YES-CONTINUE
2055  45BE                               ERR     OUTMK2,CKMK7
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45BE  CD  09  28    18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0009                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45C1  09                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45C2  00                                        .BYTE
 (1)* 45C3  CD  15  28    18.0                CKMK7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45C6  DA  DB  45    10.0                        JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2056                            ;>CHANNEL 7 NOT MARK TWO
```

```
2058  45C9  DB  15        10.0          JN      RPSTA           ;GET THE CHANNEL P BIT
2059  45CB  E6  08         7.0          ANI     R.MK2           ;CHANNEL P ILLEGAL?
2060  45CD  C2  D5  45     10.0          JNZ     CKMKP           ;YES-CONTINUE
2061  45D0                               ERR     OUTMK2,CKMKP
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45D0  CD  09  28     18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000A                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45D3  0A                                  .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45D4  00                                  .BYTE
 (1)  45D5  CD  15  28     18.0          CKMKP:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45D8  DA  DB  45     10.0                  JC      OUTMK2          ;LOOP ADDRESS IF LOOP SPECIFIED
2062                                ;>CHANNEL P NOT MARK TWO
2063
2064  45DB  C9                10.0  OUTMK2: RET                    ;RETURN
```

```
                                .SBTTL  SUBROUTINE CHECK END MARK
45DC                            SSUB
                                ; ***************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*CHECK READ CHANNELS END MARK
45DC                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO CHECK THAT ALL READ CHANNEL ''END MARK''
                                ;*BITS EQUAL ONE AND TO FLAG ANY EXCEPTIONS WITH AN ERROR MESSAGE.
45DC                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*  IF CHANNEL 0=END MARK
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 0
                                ;*  ENDIF
                                ;*  IF CHANNEL 1=END MARK
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 1
                                ;*  ENDIF
                                ;*  IF CHANNEL 2=END MARK
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 2
                                ;*  ENDIF
                                ;*  IF CHANNEL 3=END MARK
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 3
                                ;*  ENDIF
                                ;*  IF CHANNEL 4=END MARK
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 4
                                ;*  ENDIF
                                ;*  IF CHANNEL 5=END MARK
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 5
                                ;*  ENDIF
                                ;*  IF CHANNEL 6=END MARK
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 6
                                ;*  ENDIF
                                ;*  IF CHANNEL 7=END MARK
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL 7
                                ;*  ENDIF
                                ;*  IF CHANNEL P=END MARK
                                ;*    THEN-CONTINUE
                                ;*    ELSE-ERROR CHANNEL P
                                ;*  ENDIF
                                ;*ENDSUB
```

```
2111  45DC                                  S
 (1)                                        ; ************************************************************************************
2112
2113  45DC   DB   14          10.0  CKEMK:  IN      REND            ;GET THE READ CHANNEL ILLEGAL BITS
2114  45DE   E6   01           7.0          ANI     $01             ;CHANNEL 0 ILLEGAL?
2115  45E0   C2   E8   45     10.0          JNZ     CKEN0           ;YES-CONTINUE
2116  45E3                                 ERR     OUTEND,CKEN0
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45E3   CD   09   28     18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000B                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45E6   0B                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45E7   00                                        .BYTE
 (1)  45E8   CD   15   28     18.0                CKEN0:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45EB   DA   7E   46     10.0                        JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2117                                     ;>CHANNEL 0 NOT END MARK
2118
2119  45EE   DB   14          10.0          IN      REND            ;GET THE READ CHANNEL ILLEGAL BITS
2120  45F0   E6   02           7.0          ANI     $02             ;CHANNEL 1 ILLEGAL?
2121  45F2   C2   FA   45     10.0          JNZ     CKEN1           ;YES-CONTINUE
2122  45F5                                 ERR     OUTEND,CKEN1
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  45F5   CD   09   28     18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000C                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45F8   0C                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  45F9   00                                        .BYTE
 (1)  45FA   CD   15   28     18.0                CKEN1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  45FD   DA   7E   46     10.0                        JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2123                                     ;>CHANNEL 1 NOT END MARK
2124
2125  4600   DB   14          10.0          IN      REND            ;GET THE READ CHANNEL ILLEGAL BITS
2126  4602   E6   04           7.0          ANI     $04             ;CHANNEL 2 ILLEGAL?
2127  4604   C2   0C   46     10.0          JNZ     CKEN2           ;YES-CONTINUE
2128  4607                                 ERR     OUTEND,CKEN2
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4607   CD   09   28     18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000D                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  460A   0D                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  460B   00                                        .BYTE
 (1)  460C   CD   15   28     18.0                CKEN2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  460F   DA   7E   46     10.0                        JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2129                                     ;>CHANNEL 2 NOT END MARK
2130
2131  4612   DB   14          10.0          IN      REND            ;GET THE READ CHANNEL ILLEGAL BITS
2132  4614   E6   08           7.0          ANI     $08             ;CHANNEL 3 ILLEGAL?
2133  4616   C2   1E   46     10.0          JNZ     CKEN3           ;YES-CONTINUE
2134  4619                                 ERR     OUTEND,CKEN3
 (1)                                       ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4619   CD   09   28     18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000E                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  461C   0E                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  461D   00                                        .BYTE
 (1)  461E   CD   15   28     18.0                CKEN3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4621   DA   7E   46     10.0                        JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2135                                     ;>CHANNEL 3 NOT END MARK
```

```
2136
2137  4624  DB  14        10.0        IN      REND            ;GET THE READ CHANNEL ILLEGAL BITS
2138  4626  E6  10         7.0        ANI     $10             ;CHANNEL 4 ILLEGAL?
2139  4628  C2  30  46    10.0        JNZ     CKEN4           ;YES-CONTINUE
2140  462B                           ERR     OUTEND,CKEN4
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  462B  CD  09  28    18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000F                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  462E  0F                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  462F  00                                .BYTE
 (1)  4630  CD  15  28    18.0        CKEN4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4633  DA  7E  46    10.0                JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2141                          ;>CHANNEL 4 NOT END MARK
2142
2143  4636  DB  14        10.0        IN      REND            ;GET THE READ CHANNEL ILLEGAL BITS
2144  4638  E6  20         7.0        ANI     $20             ;CHANNEL 5 ILLEGAL?
2145  463A  C2  42  46    10.0        JNZ     CKEN5           ;YES-CONTINUE
2146  463D                           ERR     OUTEND,CKEN5
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  463D  CD  09  28    18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0010                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4640  10                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4641  00                                .BYTE
 (1)  4642  CD  15  28    18.0        CKEN5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4645  DA  7E  46    10.0                JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2147                          ;>CHANNEL 5 NOT END MARK
2148
2149  4648  DB  14        10.0        IN      REND            ;GET THE READ CHANNEL ILLEGAL BITS
2150  464A  E6  40         7.0        ANI     $40             ;CHANNEL 6 ILLEGAL?
2151  464C  C2  54  46    10.0        JNZ     CKEN6           ;YES-CONTINUE
2152  464F                           ERR     OUTEND,CKEN6
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  464F  CD  09  28    18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0011                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4652  11                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4653  00                                .BYTE
 (1)  4654  CD  15  28    18.0        CKEN6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4657  DA  7E  46    10.0                JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2153                          ;>CHANNEL 6 NOT END MARK
2154
2155  465A  DB  14        10.0        IN      REND            ;GET THE READ CHANNEL ILLEGAL BITS
2156  465C  E6  80         7.0        ANI     $80             ;CHANNEL 7 ILLEGAL?
2157  465E  C2  66  46    10.0        JNZ     CKEN7           ;YES-CONTINUE
2158  4661                           ERR     OUTEND,CKEN7
 (1)                               ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4661  CD  09  28    18.0                CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0012                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4664  12                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4665  00                                .BYTE
 (1)  4666  CD  15  28    18.0        CKEN7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4669  DA  7E  46    10.0                JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2159                          ;>CHANNEL 7 NOT END MARK
```

```
2161  466C  DB  15        10.0          IN      RPSTA           ;GET THE CHANNEL P BIT
2162  466E  E6  04         7.0          ANI     R.ILL           ;CHANNEL P ILLEGAL?
2163  4670  C2  78  46     10.0          JNZ     CKENP           ;YES-CONTINUE
2164  4673                               ERR     OUTEND,CKENP
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4673  CD  09  28     18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0013                                 MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4676  13                                   .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4677  00                                   .BYTE
 (1)  4678  CD  15  28     18.0          CKENP:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  467B  DA  7E  46     10.0                  JC      OUTEND          ;LOOP ADDRESS IF LOOP SPECIFIED
2165                                ;>CHANNEL P NOT END MARK
2166
2167  467E  C9                10.0  OUTEND: RET                    ;RETURN
2168
```

```
                                .SBTTL  SUBROUTINE CHECK AMTIE
                                SSUB
                                ; ***************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*CHECK AMTIE SUBROUTINE
                                SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO ISOLATE A BIT FAILURE IN THE READ PATH DATA
                                ;*OR AMTIE BYTES TO A SPECIFIC READ CHANNEL BY COMPARING THE ACTUAL AND
                                ;*EXPECTED INPUTS TO THE SUBROUTINE ON A BIT BY BIT BASIS.
                                SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*    IF CHANNEL 0 ACTUAL INFORMATION NOT = CHANNEL 0 EXPECTED
                                ;*      THEN-ERROR CHANNEL 0
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 1 ACTUAL INFORMATION NOT = CHANNEL 1 EXPECTED
                                ;*      THEN-ERROR CHANNEL 1
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 2 ACTUAL INFORMATION NOT = CHANNEL 2 EXPECTED
                                ;*      THEN-ERROR CHANNEL 2
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 3 ACTUAL INFORMATION NOT = CHANNEL 3 EXPECTED
                                ;*      THEN-ERROR CHANNEL 3
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 4 ACTUAL INFORMATION NOT = CHANNEL 4 EXPECTED
                                ;*      THEN-ERROR CHANNEL 4
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 5 ACTUAL INFORMATION NOT = CHANNEL 5 EXPECTED
                                ;*      THEN-ERROR CHANNEL 5
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 6 ACTUAL INFORMATION NOT = CHANNEL 6 EXPECTED
                                ;*      THEN-ERROR CHANNEL 6
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 7 ACTUAL INFORMATION NOT = CHANNEL 7 EXPECTED
                                ;*      THEN-ERROR CHANNEL 7
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL P ACTUAL INFORMATION NOT = CHANNEL P EXPECTED
                                ;*      THEN-ERROR CHANNEL P
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
```

```
2215                                        ;*ENDSUB
2216  467F                                  $
 (1)                                        ; ************************************************************************
2217  467F   21   84   47      10.0  CKAMT:  LXI     H,DATAE          ;LOAD EXPECTED DATA ADDRESS
2218  4682   3A   82   47      13.0          LDA     DATAA            ;GET THE ACTUAL DATA
2219  4685                                  ROUT    ADATA            ;WRITE THE ACTUAL DATA TO THE CAS
 (1)  4685   D3   94              10.0                  OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  4687   7F                    4.0                  MOV     A,A              ;RETRY LINK
2220  4688   E6   01               7.0          ANI     $01              ;REMOVE UNWANTED BITS
2221  468A   47                    4.0          MOV     B,A              ;SAVE IN B
2222  468B   7E                    7.0          MOV     A,M              ;GET EXPECTED DATA
2223  468C                                  ROUT    EDATA            ;WRITE THE EXPECTED DATA TO THE CAS
 (1)  468C   D3   95              10.0                  OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  468E   7F                    4.0                  MOV     A,A              ;RETRY LINK
2224  468F   E6   01               7.0          ANI     $01              ;REMOVE UNWANTED BITS
2225  4691   B8                    4.0          CMP     B                ;COMPARE
2226  4692   CA   9A   46         10.0          JZ      CKAM0            ;CONTINUE IF EQUAL
2227  4695                                  ERRB    OUT,CKAM0,1
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4695   CD   12   28      18.0                  CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)       0014                                     MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4698   14                                     .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  4699   01                                     .BYTE   1                ;PRINT ROUTINE NUMBER
 (1)  469A   CD   15   28      18.0          CKAM0:: CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  469D   DA   63   47      10.0                  JC      OUT              ;LOOP ADDRESS IF LOOP SPECIFIED
2228                                    ;>CHANNEL 0 FAILED
2229
2230  46A0   3A   82   47      13.0          LDA     DATAA            ;GET THE ACTUAL DATA
2231  46A3   E6   02               7.0          ANI     $02              ;REMOVE UNWANTED BITS
2232  46A5   47                    4.0          MOV     B,A              ;SAVE IN B
2233  46A6   7E                    7.0          MOV     A,M              ;GET EXPECTED DATA
2234  46A7   E6   02               7.0          ANI     $02              ;REMOVE UNWANTED BITS
2235  46A9   B8                    4.0          CMP     B                ;COMPARE
2236  46AA   CA   B2   46         10.0          JZ      CKAM1            ;CONTINUE IF EQUAL
2237  46AD                                  ERRB    OUT,CKAM1,1
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46AD   CD   12   28      18.0                  CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)       0015                                     MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46B0   15                                     .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  46B1   01                                     .BYTE   1                ;PRINT ROUTINE NUMBER
 (1)  46B2   CD   15   28      18.0          CKAM1:: CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46B5   DA   63   47      10.0                  JC      OUT              ;LOOP ADDRESS IF LOOP SPECIFIED
2238                                    ;>CHANNEL 1 FAILED
```

```
2240  46B8  3A 82 47      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2241  46BB  E6 04          7.0          ANI     $04             ;REMOVE UNWANTED BITS
2242  46BD  47              4.0          MOV     B,A             ;SAVE IN B
2243  46BE  7E              7.0          MOV     A,M             ;GET EXPECTED DATA
2244  46BF  E6 04           7.0          ANI     $04             ;REMOVE UNWANTED BITS
2245  46C1  B8              4.0          CMP     B               ;COMPARE
2246  46C2  CA CA 46       10.0          JZ      CKAM2           ;CONTINUE IF EQUAL
2247  46C5                               ERPB    OUT,CKAM2,1
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46C5  CD 12 28       18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0016                                     MSGN    =       MSGN+1          ,UPDATE MESSAGE NUMBER FOR THIS
 (1)  46C8  16                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46C9  01                                      .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  46CA  CD 15 28       18.0              CKAM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46CD  DA 63 47 *     10.0                      JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2248                            ;>CHANNEL 2 FAILED
2249  46D0  3A 82 47      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2250  46D3  E6 08           7.0          ANI     $08             ;REMOVE UNWANTED BITS
2251  46D5  47              4.0          MOV     B,A             ;SAVE IN B
2252  46D6  7E              7.0          MOV     A,M             ;GET EXPECTED DATA
2253  46D7  E6 08           7.0          ANI     $08             ;REMOVE UNWANTED BITS
2254  46D9  B8              4.0          CMP     B               ;COMPARE
2255  46DA  CA E2 46       10.0          JZ      CKAM3           ;CONTINUE IF EQUAL
2256  46DD                               ERRB    OUT,CKAM3,1
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46DD  CD 12 28       18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0017                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46E0  17                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46E1  01                                      .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  46E2  CD 15 28       18.0              CKAM3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46E5  DA 63 47       10.0                      JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2257                            ;>CHANNEL 3 FAILED
2258
2259  46E8  3A 82 47      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2260  46EB  E6 10           7.0          ANI     $10             ;REMOVE UNWANTED BITS
2261  46ED  47              4.0          MOV     B,A             ;SAVE IN B
2262  46EE  7E              7.0          MOV     A,M             ;GET EXPECTED DATA
2263  46EF  E6 10           7.0          ANI     $10             ;REMOVE UNWANTED BITS
2264  46F1  B8              4.0          CMP     B               ;COMPARE
2265  46F2  CA FA 46       10.0          JZ      CKAM4           ;CONTINUE IF EQUAL
2266  46F5                               ERRB    OUT,CKAM4,1
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46F5  CD 12 28       18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0018                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46F8  18                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46F9  01                                      .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  46FA  CD 15 28       18.0              CKAM4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46FD  DA 63 47       10.0                      JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2267                            ;>CHANNEL 4 FAILED
```

```
2269  4700  3A 82 47      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2270  4703  E6 20           7.0          ANI     $20             ;REMOVE UNWANTED BITS
2271  4705  47              4.0          MOV     B,A             ;SAVE IN B
2272  4706  7E              7.0          MOV     A,M             ;GET EXPECTED DATA
2273  4707  E6 20           7.0          ANI     $20             ;REMOVE UNWANTED BITS
2274  4709  B8              4.0          CMP     B               ;COMPARE
2275  470A  CA 12 47       10.0          JZ      CKAM5           ;CONTINUE IF EQUAL
2276  470D                               ERRB    OUT,CKAM5,1
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  470D  CD 12 28   18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0019                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4710  19                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4711  01                                    .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4712  CD 15 28   18.0                CKAM5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4715  DA 63 47   10.0                        JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2277                            ;>CHANNEL 5 FAILED
2278
2279  4718  3A 82 47      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2280  471B  E6 40           7.0          ANI     $40             ;REMOVE UNWANTED BITS
2281  471D  47              4.0          MOV     B,A             ;SAVE IN B
2282  471E  7E              7.0          MOV     A,M             ;GET EXPECTED DATA
2283  471F  E6 40           7.0          ANI     $40             ;REMOVE UNWANTED BITS
2284  4721  B8              4.0          CMP     B               ;COMPARE
2285  4722  CA 2A 47       10.0          JZ      CKAM6           ;CONTINUE IF EQUAL
2286  4725                               ERRB    OUT,CKAM6,1
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4725  CD 12 28   18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       001A                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4728  1A                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4729  01                                    .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  472A  CD 15 28   18.0                CKAM6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  472D  DA 63 47   10.0                        JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2287                            ;>CHANNEL 6 FAILED
2288
2289  4730  3A 82 47      13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2290  4733  E6 80           7.0          ANI     $80             ;REMOVE UNWANTED BITS
2291  4735  47              4.0          MOV     B,A             ;SAVE IN B
2292  4736  7E              7.0          MOV     A,M             ;GET EXPECTED DATA
2293  4737  E6 80           7.0          ANI     $80             ;REMOVE UNWANTED BITS
2294  4739  B8              4.0          CMP     B               ;COMPARE
2295  473A  CA 42 47       10.0          JZ      CKAM7           ;CONTINUE IF EQUAL
2296  473D                               ERRB    OUT,CKAM7,1
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  473D  CD 12 28   18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       001B                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4740  1B                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4741  01                                    .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4742  CD 15 28   18.0                CKAM7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4745  DA 63 47   10.0                        JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2297                            ;>CHANNEL 7 FAILED
```

```
2299  4748  21  85  47   10.0          LXI     H,DATAEP        ;GET THE EXPECTED DATA PARITY ADDRESS
2300  474B  3A  83  47   13.0          LDA     DATAAP          ;GET THE ACTUAL DATA
2301  474E  E6  01        7.0          ANI     $01             ;REMOVE UNWANTED BITS
2302  4750  47            4.0          MOV     B,A             ;SAVE IN B
2303  4751  7E            7.0          MOV     A,M             ;GET EXPECTED DATA
2304  4752  E6  01        7.0          ANI     $01             ;REMOVE UNWANTED BITS
2305  4754  B8            4.0          CMP     B               ;COMPARE
2306  4755  CA  5D  47    10.0          JZ      CKAMP           ;CONTINUE IF EQUAL
2307  4758                             ERRB    OUT,CKAMP,1
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4758  CD  12  28   18.0                       CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       001C                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  475B  1C                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  475C  01                                     .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  475D  CD  15  28   18.0                CKAMP:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4760  DA  63  47   10.0                        JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2308                             ;>CHANNEL P FAILED
2309
2310  4763  C9            10.0  OUT:    RET
```

```
2312                                    .SBTTL  TABLE INPUT DATA MARK TWO
2313
2314  4764  FF          M8950MK: .BYTE  @377            ;DATA GROUP 1 - CHARACTER 1
2315  4765  01                   .BYTE  @001            ;DATA GROUP 1 - CHARACTER 1  PARITY
2316  4766  FF                   .BYTE  @377            ;DATA GROUP 1 - CHARACTER 2
2317  4767  01                   .BYTE  @001            ;DATA GROUP 1 - CHARACTER 2  PARITY
2318  4768  FF                   .BYTE  @377            ;DATA GROUP 1 - CHARACTER 3
2319  4769  01                   .BYTE  @001            ;DATA GROUP 1 - CHARACTER 3  PARITY
2320  476A  00                   .BYTE  @000            ;DATA GROUP 1 - CHARACTER 4
2321  476B  00                   .BYTE  @000            ;DATA GROUP 1 - CHARACTER 4  PARITY
2322  476C  00                   .BYTE  @000            ;DATA GROUP 1 - CHARACTER 5
2323  476D  80                   .BYTE  @200            ;DATA GROUP 1 - CHARACTER 5  PARITY
2324
2325                                    .SBTTL  TABLE INPUT DATA END MARK
2326  476E  FF          M8950EM: .BYTE  @377            ;DATA GROUP 1 - CHARACTER 1
2327  476F  01                   .BYTE  @001            ;DATA GROUP 1 - CHARACTER 1  PARITY
2328  4770  FF                   .BYTE  @377            ;DATA GROUP 1 - CHARACTER 2
2329  4771  01                   .BYTE  @001            ;DATA GROUP 1 - CHARACTER 2  PARITY
2330  4772  FF                   .BYTE  @377            ;DATA GROUP 1 - CHARACTER 3
2331  4773  01                   .BYTE  @001            ;DATA GROUP 1 - CHARACTER 3  PARITY
2332  4774  FF                   .BYTE  @377            ;DATA GROUP 1 - CHARACTER 4
2333  4775  01                   .BYTE  @001            ;DATA GROUP 1 - CHARACTER 4  PARITY
2334  4776  FF                   .BYTE  @377            ;DATA GROUP 1 - CHARACTER 5
2335  4777  81                   .BYTE  @201            ;DATA GROUP 1 - CHARACTER 5  PARITY
```

```
2337                                    .SBTTL  PROGRAM VARIABLES
2338  4778   00                         UNITMP: .BYTE   0               ;UNIT MAP
2339  4779   00                         SUNIT:  .BYTE   0               ;PORT/TU SELECT BYTE
2340  477A   00   00                    AMTMSK: .WORD   0               ;AMTIE MASK WORD
2341  477C   00   00                    INTBLTMP: .WORD 0               ;TEMP. STORAGE FOR THE INPUT TABLE POINTER
2342  477E   00   00                    INTBL:  .WORD   0               ;INPUT TABLE ADDRESS STORAGE
2343  4780   00   00                    OUTTBL: .WORD   0               ;OUTPUT TABLE ADDRESS STORAGE
2344  4782   00                         DATAA:  .BYTE   0               ;ACTUAL DATA AFTER TRANSLATION
2345  4783   00                         DATAAP: .BYTE   0               ;ACTUAL DATA PARITY AFTER TRANSLATION
2346  4784   00                         DATAE:  .BYTE   0               ;EXPECTED DATA AFTER TRANSLATION
2347  4785   00                         DATAEP: .BYTE   0               ;EXPECTED DATA PARITY AFTER TRANSLATION
2348  4786   00                         SUBCNT: .BYTE   0               ;SUBGROUP COUNTER
2349  4787   00                         PASCNT: .BYTE   0               ;PASS COUNTER
2350        0000                                .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
AMTMSK  477A           ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000         BADST  = 0090          BIT0   = 0001
BIT1   = 0002          BIT15  = 8000         BIT2   = 0004          BIT3   = 0008
BIT4   = 0010          BIT5   = 0020         BIT6   = 0040          BIT7   = 0080
BIT8   = 0100          BIT9   = 0200         BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60           BRKXCT= 4F00          BSAVE   4F9C          BYTCNT= 00D4
BYTEH   4F24           BYTEL   4F23          C      =%0001         CASCT   4F21
CASCTL= 00A0           CASSTA= 00A0          CATTH  = 0089          CATTL  = 0088
CBUSST= 00A1           CBYTH  = 008B         CBYTL  = 008A          CDG1H  = 0087
CDG1L  = 0086          CDG2H  = 0093         CDG2L  = 0092          CDG3H  = 0095
CDG3L  = 0094          CDVTH  = 008D         CDVTL  = 008C          CHPTIE= 0028
CHOTIE= 0020           CH1TIE= 0021          CH2TIE= 0022          CH3TIE= 0023
CH4TIE= 0024           CH5TIE= 0025          CH6TIE= 0026          CH7TIE= 0027
CKAMP   475D  G        CKAMT   467F          CKAM0   469A  G       CKAM1   46B2  G
CKAM2   46CA  G        CKAM3   46E2  G       CKAM4   46FA  G       CKAM5   4712  G
CKAM6   472A  G        CKAM7   4742  G       CKEMK   45DC          CKENP   4678  G
CKEN0   45E8  G        CKEN1   45FA  G       CKEN2   460C  G       CKEN3   461E  G
CKEN4   4630  G        CKEN5   4642  G       CKEN6   4654  G       CKEN7   4666  G
CKLOP  = 2815          CKMK    4539          CKMKP   45D5  G       CKMK0   4545  G
CKMK1   4557  G        CKMK2   4569  G       CKMK3   457B  G       CKMK4   458D  G
CKMK5   459F  G        CKMK6   45B1  G       CKMK7   45C3  G       CLEAR   450E
CLKCTL= 00F0           CLOCK   4F26          CLRLP   4512          CMC0H  = 0099
CMC0L  = 0098          CMC1H  = 009B         CMC1L  = 009A         CMC2H  = 009D
CMC2L  = 009C          CMC3H  = 009F         CMC3L  = 009E         CMINH  = 0097
CMINL  = 0096          CNTCTL= 00D7          CRCWRD= 0018          CSAVE   4F9D
CSRLH  = 0091          CSRLL  = 0090         CTCH   = 0085         CTCL   = 0084
CTSTH  = 008F          CTSTL  = 008E         CXCTH  = 0081         CXCTL  = 0080
CXINH  = 0083          CXINL  = 0082         C.     = 0001         C.AVAI= 0080
C.DP   = 0008          C.DSE  = 0010         C.DTU  = 0003         C.DVA  = 0008
C.FAIL= 00FC           C.FMT  = 0070         C.FNCT= 003E          C.GO   = 0001
C.INTC= 00FE           C.MAIN= 0020          C.NSA  = 0080         C.RCT  = 00FC
C.SEK  = 0080          C.SHR  = 0040         C.SKPC= 000F          C.TAPE= 0040
C.WCS  = 0002          D      =%0002         DATAA   4782          DATAAP  4783
DATACT= 00D0           DATAE   4784          DATAEP  4785          DBUS    4F28
DBUSCT= 00C0           DBUSST= 00C0          DDRA   = 00D8         DDRAIN= 0010
DDRB   = 00D9          DDRBIN= 0002          DDRC   = 00DA         DDRCIN= 0001
DDRCO  = 0088          DDRCTL= 00DB          DIAFLG  4F22          DIAGPG= 4300
DIAGRM= 4F90           DIARD  = 0008         DONE1  = 0045         DONINT= 0010
DSAVE   4F9E           DSE    = 0006         DUMMY   431E  G       D.ATH0= 0001
D.ATH1= 0002           D.EOTD= 0010          D.LAGC= 0020          D.NOTW= 0040
D.NTHR= 0004           D.TACH= 0008          D.WR4  = 0080         E      =%0003
ECCBAD= 0042           ECCCOR= 0019          ECCOK  = 0041         ECCSTA= 001A
ECCTST= 000E           EDATA  = 0095         EOTCLR= 0003          ERFLG   4F93
ERLP   = 2809          ERLPA  = 280F         ERLPB  = 2812         ERLPE  = 280C
ERNUM   4F90           ERRCNT= 00D6          ESAVE   4F9F          EXIT    450B
E.ACRC= 0010           E.AMT  = 0020         E.CDP  = 0080         E.CRC  = 0080
E.PNTR= 0008           E.RPE  = 0040         E.STEC= 0001          E.TTEC= 0002
E.UNC  = 0004          FIFORD= 006A          FORMAT  4F25          FOUND   4344
FWDTST= 0061           GCRID  = 0089         GCRSET= 0002          GOODTM= 0092
H      =%0004          HLSAVE  4FA0          IE     = 0008         INTBL   477E
INTBLT  477C           INTSTA= 00E0          ITERA   4F9A          I.PWR  = 0020
I.RMPE= 0040           I5.5   = 0010         I6.5   = 0020         I7.5   = 0040
KCALL  = 005F          KCLR   = 007B         KDEP   = 003F         KENAB  = 0078
```

```
KEXAM = 003E    KEYBRD= 00C8    KEY1  = 0078    KEY10 = 006D
KEY11 = 006E    KEY12 = 006F    KEY13 = 005C    KEY14 = 005D
KEY15 = 005E    KEY16 = 005F    KEY17 = 003C    KEY18 = 003D
KEY19 = 003E    KEY2  = 0079    KEY20 = 003F    KEY3  = 007A
KEY4  = 007B    KEY5  = 0074    KEY6  = 0075    KEY7  = 0076
KEY8  = 0077    KEY9  = 006C    KINTA = 006F    KLDAD = 003D
KN0   = 003C    KN1   = 005C    KN2   = 005D    KN3   = 005E
KN4   = 006C    KN5   = 006D    KN6   = 006E    KN7   = 0074
KN8   = 0075    KN9   = 0076    KU2   = 0079    KU3   = 007A
KU8   = 0077    L     =%0005    LBLANK= 000F    LCE   = 000B
LCH   = 000C    LCL   = 000D    LCP   = 000E    LC0   = 0000
LC1   = 0001    LC2   = 0002    LC3   = 0003    LC4   = 0004
LC5   = 0005    LC6   = 0006    LC7   = 0007    LC8   = 0008
LC9   = 0009    LDLEDA= 00CA    LDLEDB= 00CB    LDLEDC= 00CC
LDLEDD= 00CD    LDLEDE= 00CE    LDLEDF= 00CF    LKDIAG= 2800
LKKBD = 004C    LKKEY = 0049    LKLWMG= 0058    LKLWMP= 0055
LKLWPG= 0052    LKLWFP= 004F    LKMOD7= 0046    LKOPR = 0046
LPFLG   4F94    LPNUM   4F92    M     =%0006    MBSEL = 00E0
MB.A  = 0008    MB.B  = 0004    MEMTOP= 4FFF    MINUS = 000A
MM    = 8000    MSE   = 0008    MSGN  = 001C    MTACLR= 0000
MT.ARA= 0020    MT.CPE= 0080    MT.DSE= 0001    MT.FWD= 0040
MT.INH= 0008    MT.LWR= 0004    MT.MOT= 0002    MT.NWT= 0080
MT.PEC= 0040    MT.PSB= 0004    MT.PS0= 0001    MT.PS1= 0002
MT.REV= 0020    MT.WRT= 0010    MT.Z  = 0008    M.ATA = 0080
M.CAPE= 0020    M.CONT= 0080    M.DEM = 0020    M.EBL = 0010
M.EXC = 0008    M.FAIL= 0008    M.ILR = 0010    M.INIT= 0010
M.OCC = 0020    M.ONLI= 0001    M.PE  = 0040    M.PORT= 0080
M.RDEN= 0002    M.RDPE= 0008    M.RUN = 0004    M.SCLK= 0001
M.TRA = 0040    M.UNIT= 0007    M.WCLK= 0040    M.WCLN= 0080
M.WREN= 0080    M5.5  = 0001    M6.5  = 0002    M7.5  = 0004
M8950E  476E    M8950M  4764    NOTCAP= 0088    OKAY  = 00FF
OPRRAM= 4300    OPSTRT= 0058    OPVER = 0040    OUT     4763
OUTEND  467E    OUTMK2  45DB    OUTTBL  4780    PADCNT= 00D5
PADCRC= 0087    PASCNT  4787    PDIAG = 0048    PEID  = 008A
PENAB = 004C    PESET = 0001    PL    = 00B1    PRDD  = 004C
PRENF = 009C    PS    = 00B2    PSTAT = 0048    PSW   =%0009
P.AMTP= 0001    P.BCTC= 0040    P.CMDP= 0020    P.INTE= 0080
P.LCS = 0040    P.LWR = 0020    P.RDP = 0002    P.RPEN= 0004
P.RPST= 0002    P.RP0E= 0020    P.RP1E= 0010    P.RP2E= 0020
P.RP3E= 0010    P.SING= 0080    P.STAT= 0002    P.STPE= 0080
P.TACH= 0008    P.TUPR= 0010    P.WCSP= 0004    P.WDS = 0040
P.WFLP= 0001    P.WPEN= 0010    P.WP0F= 0008    P.WP1E= 0004
P.WP2E= 0008    P.WP3E= 0004    P.5VOK= 0002    QUE   = 281B
QUEM  = 281E    RAMT  = 0010    RARA  = 0006    RARAI = 0004
RCHBD0= 0048    RCHBD1= 0047    RCHOK = 0046    RCHTST= 000C
RCLRT = 000D    RCMD  = 000B    RCMLP = 0003    RCONT = 0080
RDATA = 0017    RDCLK = 0010    RDON  = 0011    READG = 0007
REND  = 0014    REQST = 2806    RESCHR= 00D1    REVTST= 0064
REWIND= 0004    RFIFOL= 0008    RGCLK = 0002    RGCRI = 0003
RIBG  = 0001    RILL  = 0012    RINST = 000C    RMCTST= 0008
RMK2  = 0013    RNOP  = 0000    RPATH = 0001    RPBAD = 0044
RPCHI = 0001    RPCLK = 0003    RPCTL = 0009    RPEI  = 0002
RPFAIL= 0000    RPF1  = 009D    RPF2  = 009E    RPOK  = 0043
RPOSTN= 0016    RPSTA = 0015    RRCMT = 000A    RSTAT = 0002
```

```
RTIEB = 000A            RTIER = 0030            RTM   = 0005            RUNKI = 0009
RUPTST= 005E            RWDUNL= 0005            R.AMT = 0001            R.BOP = 0008
R.DATA= 0040            R.DON = 0002            R.DRDY= 0010            R.END = 0010
R.ILL = 0004            R.JVOK= 0004            R.MK2 = 0008            R.PLOD= 0008
R.PLO0= 0010            R.PLO1= 0020            R.POST= 0020            R.STNM= 0002
R.STOP= 0004            R.STPC= 0001            R.TBJN= 0080            R.TSTD= 0040
R.VOK = 0080            R00H  = 0081            R00L  = 0080            R01H  = 0083
R01L  = 0082            R02H  = 0085            R02L  = 0084            R03H  = 0087
R03L  = 0086            R04H  = 0089            R04L  = 0088            R05H  = 008B
R05L  = 008A            R06H  = 008D            R06L  = 008C            R07H  = 008F
R07L  = 008E            R10H  = 0091            R10L  = 0090            R11H  = 0093
R11L  = 0092            R12H  = 0095            R12L  = 0094            R13H  = 0097
R13L  = 0096            R14H  = 0099            R14L  = 0098            R15H  = 009B
R15L  = 009A            R16H  = 009D            R16L  = 009C            R17H  = 009F
R17L  = 009E            R7.5  = 0010            SELCLR= 0000            SETATA= 00A1
SID   = 0080            SOD   = 0080            SOE   = 0040            SP    =%0008
SSCLK = 0040            SSTE  = 0005            STACK = 4FFF            STATRM= 4F20
STPCT   4F20            STRSP = 5000            SUBCNT  4786            SUNIT   4779
TADR00= 0080            TADR01= 0081            TADR02= 0082            TADR03= 0083
TADR04= 0084            TADR05= 0085            TADR06= 0086            TADR07= 0087
TADR10= 0088            TADR11= 0089            TADR12= 008A            TADR13= 008B
TAMT  = 0044            TASEL = 0080            TCMD  = 0040            TC.FWD= 0040
TC.INH= 0008            TC.LWR= 0004            TC.REV= 0020            TC.WRT= 0010
TEMP    4F99            TEST1   4300            TEST2   43DE            TEST3   4475
TMF   = 0099            TMRDY = 0040            TRKENA= 00D2            TSET  = 2803
TSTEND= 2818            TSTS  = 0040            TST01A  43BF            TST01C  4377
TST01X  4351            TST02A  4456            TST02C  440E            TST02X  43E8
TST03X  447A            TUSEL0= 00D1            TUSEL1= 00D2            TU78  = 0010
T.ATH0= 0001            T.ATH1= 0002            T.BOT = 0004            T.EOT = 0002
T.FPT = 0001            T.NTHR= 0004            T.ONL = 0020            T.PES = 0008
T.PSBJ= 0020            T.PSOJ= 0008            T.PS1J= 0010            T.RDY = 0080
T.RDYO= 0040            T.RWD = 0010            T.SCLK= 0002            UIBG  = 00A1
UNIIMP  4778            UNITSL  4327            VALFC   4F98            VALTB   4F95
VELTST= 005B            WDR.P = 0010            WMCCTL= 00D3            WMCERR= 00DA
WMCSTA= 00D0            WRTCLK= 0000            WRTDAT= 00D3            W.ACRC= 0004
W.CRC = 0008            W.DIAG= 0002            W.DONN= 0040            W.ECC = 0010
W.ENAB= 0080            W.ERR = 0020            W.FMT = 0070            W.GCR = 0010
W.LEFT= 0004            W.ONES= 0020            W.RESI= 0002            W.REV = 0004
W.ROME= 0010            W.RST = 0001            W.SKIP= 000F            W.WRIT= 0008
W.XFER= 0020            X     =%000A            X.DONN= 0080            X.ENAB= 0040
X.PEPE= 0002            X.ROME= 0001            X.WCLK= 0001            Y     =%000B
.     = 4788

 ERRORS DETECTED:  0

 *RPM5.A78/PTP,RPM5=NLIST,PARAM,MACRO,LIST,RPM5
 RUN-TIME: 4 6 0 SECONDS
 CORE USED:  10K
```

3 "MEMORY MAP" EXTENDED ADDRESS BIT
19 READ MICRO CONTROLLER CONTROL & STATUS WORDS
139 TRANSPORT/PORT COMMAND + STATUS WORDS
380 COMMON ADDRESS SPACE REGISTERS
444 MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485 MASSBUS CONTROL WORD DEFINITIONS
557 WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709 KEYBOARD/DISPLAY CONTROL + STATUS WORD
797 MISCELLANEOUS VARIABLE DEFINITIONS
801 RAM ADDRESS DEFINITIONS
819 LINKAGE TABLE ADDRESSES
842 BIT DEFINITIONS
856 RIM AND SIM INSTRUCTION BIT DEFINITIONS
877 INTERRUPTABLE INSTRUCTION MACROS
924 DIAGNOSTIC VARIABLES
945 COMMON VARIABLE STORAGE
959 MACRO CALL DEFINITIONS
1047 TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101 8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1333 TEST 1 - ECC NO CORRECTION TEST
1904 TEST 2 - ECC SOFT POINTERS NO BAD DATA
2460 SUBROUTINE CLEAR ALL TU PORTS
2505 SUBROUTINE 4X5 TRANSLATE A SUBGROUP
2617 TABLE 4 X 5 TRANSLATION
2658 SUBROUTINE VARIABLES
2668 SUBROUTINE CLEAR ECC
2680 SUBROUTINE CALCUALTE ECC CHARACTER
2716 SUBROUTINE POLYNOMIAL BIT TRANSLATION
2738 SUBROUTINE CHECK 5X4
2885 TABLE TIE BUS VECTOR TABLE
2905 TABLE TIE BUS DATA VALUES
3066 PROGRAM VARIABLES

```
                ;*************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;               "BYTE/SCLK COUNT NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               MM = DATA FORMAT FROM CAS REGISTER 2
                ;               NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;               BYTE-SCLK COUNT = LLL
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               LLL = AS ABOVE
                ;               MM  = AS ABOVE
                ;               NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;               TRANSITION COUNT = LLL
                ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;               EXPECTED 18 BITS =      E EEEEEE
                ;               ACTUAL 18 BITS =        A AAAAAA
                ;
                ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;               OF CAS REG 15 LOW BYTE.
                ;
                ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;               17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;               AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;               "SUBGROUP NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;*************************************************************************
```

```
1016                    ; ************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ; ************************************************************************
1035                    ; ************************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ; ************************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  ECC1 - ECC CONTROLLER PART #1
                                .SBTTL  TEST 1 - ECC NO CORRECTION TEST
                                ;ID     ECC1-ERROR CORRECTION CONTROLLER PART #1
4300                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*ECC NO CORRECTION TEST
4300                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE ECC CONTROLLER TO PASS DATA THROUGH
                                ;*WITHOUT PERFORMING ANY CORRECTION ON THE DATA.  THE DATA INPUT IS CORRECT
                                ;*IN EVERY DETAIL I.E. 5 TO 4 TRANSLATION, PROPER ECC CHARACTER, AND NO
                                ;*TRACK IN ERROR POINTERS.
4300.                           SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
                                ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
                                ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
                                ;*    CLEAR THE TIE BUS
                                ;*    CLEAR THE INPUT DATA
                                ;*  BGNDO
                                ;*  : ENABLE THE READ PATH CLOCK
                                ;*  : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
                                ;*  : CLOCK THE FIFO'S
                                ;*  : ISSUE CLEAR ALL COMMAND
                                ;*  : WAIT
                                ;*  : STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
                                ;*  : CALL SUBROUTINE CLECC
                                ;*  : SET THE LOOP COUNT TO 7
                                ;*  : BGNDO
                                ;*  : :  GET THE INPUT DATA BYTE
                                ;*  : :  CALL SUBROUTINE ECC
                                ;*  : :  DECREMENT THE LOOP COUNT
                                ;*  : :  DO UNTIL THE LOOP COUNT=0
                                ;*  : ENDDO
                                ;*  : SET THE LOOP COUNT TO 3
                                ;*  : BGNDO
                                ;*  : :  FILL THE TRANSLATOR SUBROUTINE BUFFER WITH THE DATA BYTE
                                ;*  : :  CALL SUBROUTINE T4X5
                                ;*  : :  INIT THE LOOP COUNT TO 5
                                ;*  : :  BGNDO
                                ;*  : :  :  MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
                                ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
                                ;*  : :  :  MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
                                ;*  : :  :  CLOCK THE DATA INTO THE FIFO
                                ;*  : :  :  DECREMENT THE LOOP COUNT
```

```
1377                                    ;*  : : :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1378                                    ;*  : : :  DO UNTIL LOOP COUNT = 0
1379                                    ;*  : : ENDDO
1380                                    ;*  : : FILL THREE BYTES OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE DATA BYTE
1381                                    ;*  : : FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
1382                                    ;*  : : CALL SUBROUTINE T4X5
1383                                    ;*  : : INIT THE LOOP COUNT TO 5
1384                                    ;*  : : BGNDO
1385                                    ;*  : : :  MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
1386                                    ;*  : : :  MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
1387                                    ;*  : : :  CLOCK THE DATA INTO THE FIFO
1388                                    ;*  : : :  DECREMENT THE LOOP COUNT
1389                                    ;*  : : :  DO UNTIL THE LOOP COUNT = 0
1390                                    ;*  : : ENDDO
1391                                    ;*  : : DECREMENT THE LOOP COUNT
1392                                    ;*  : : DO UNTIL THE LOOP COUNT = 0
1393                                    ;*  : ENDDO
1394                                    ;*  : ISSUE DIAGNOSTIC READ COMMAND
1395                                    ;*  : SET UP WATCHDOG TIMER COUNT
1396                                    ;*  : BGNDO
1397                                    ;*  : :    SINGLE STEP THE READ PATH
1398                                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1399                                    ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1400                                    ;*  : ENDDO
1401                                    ;*  : IF WATCHDOG TIMER=0
1402                                    ;*  : :    THEN-ERROR
1403                                    ;*  : :    ELSE-CONTINUE
1404                                    ;*  : ENDIF
1405                                    ;*  : SINGLE STEP THE READ PATH
1406                                    ;*  : CALL SUBROUTINE CKDONE
1407                                    ;*  : INIT THE LOOP COUNT TO 8
1408                                    ;*  : BGNDO
1409                                    ;*  : :    SINGLE STEP THE READ PATH
1410                                    ;*  : :    INPUT THE ACTUAL DATA
1411                                    ;*  : :    INPUT THE ACTUAL PARITY
1412                                    ;*  : :    CALL SUBROUTINE C[illegible]X4
1413                                    ;*  : :    DECREMENT THE LOOP COUNT
1414                                    ;*  : :    DO UNTIL LOOP [illegible]UNT = 0
1415                                    ;*  : ENDDO
1416                                    ;*  : INPUT THE ECC CHARACTER FROM THE READ CHANNELS
1417                                    ;*  : IF ACTUAL ECC CHARACTER=EXPECTED ECC CHARACTER
1418                                    ;*  : :  THEN-CONTINUE
1419                                    ;*  : :  ELSE-ERROR
1420                                    ;*  : ENDIF
1421                                    ;*  : SET UP WATCHDOG TIMER COUNT
1422                                    ;*  : BGNDO
1423                                    ;*  : :  SINGLE STEP THE READ PATH
1424                                    ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
1425                                    ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
1426                                    ;*  : ENDDO
1427                                    ;*  : IF WATCHDOG TIMER=0
1428                                    ;*  : :  THEN-ERROR-EXIT TEST
1429                                    ;*  : :  ELSE-CONTINUE
1430                                    ;*  : ENDIF
```

```
                                ;*  : SINGLE STEP THE READ PATH
                                ;*  : INIT THE  LOOP COUNT TO 7
                                ;*  : BGNDO
                                ;*  : :  SINGLE STEP THE READ PATH
                                ;*  : :  COMPARE CORRECTED DATA WITH INPUT DATA
                                ;*  : :  IF NOT EQUAL
                                ;*  : :  :  THEN-ECC ERROR
                                ;*  : :  :  ELSE-CONTINUE
                                ;*  : :  ENDIF
                                ;*  : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
                                ;*  : :  IF NOT EQUAL
                                ;*  : :  :  THEN-ECC ERROR
                                ;*  : :  :  ELSE-CONTINUE
                                ;*  : :  ENDIF
                                ;*  : :  DECREMENT THE LOOP COUNT
                                ;*  : :  DO UNTIL THE LOOP COUNT = 0
                                ;*  : ENDDO
                                ;*  : SINGLE STEP THE READ PATH
                                ;*  : IF EXPECTED NOT EQUAL
                                ;*  : :  THEN-ECC ERROR
                                ;*  : :  ELSE-CONTINUE
                                ;*  : ENDIF
                                ;*  : INPUT THE ECC STATUS
                                ;*  : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
                                ;*  : IF THE RESULTING STATUS=0
                                ;*  : :  THEN-CONTINUE
                                ;*  : :  ELSE-ERROR
                                ;*  : ENDIF
                                ;*  : INCREMENT THE INPUT DATA
                                ;*  : DO UNTIL INPUT DATA = 0.
                                ;*  ENDDO
                                ;*ENDTST
4300                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*ECC1 MICRO TEST 01
                                ;*ECC1 MICRO ERROR 01
                                ;*ECC1-CONTROLLER-NO CORRECTION TEST
                                ;*M8951, M8950'S
                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*ECC1 MICRO TEST 01
                                ;*ECC1 MICRO ERROR 02
                                ;*ECC1-CONTROLLER-NO CORRECTION TEST
                                ;*M8951, M8950'S
                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
                                ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
                                ;*BYTE/SCLK COUNT NUMBER = LLL
                                ;*FATAL ERROR - MICRO TEST ABORTED
                                ;*
                                ;*ECC1 MICRO TEST 01
                                ;*ECC1 MICRO ERROR 03
```

```
1482                            ;*ECC1-CONTROLLER-NO CORRECTION TEST
1483                            ;*M8951, M8950'S
1484                            ;*ECC CHAPACTER INCORRECT AT OUTPUT OF M8950'S
1485                            ;*BYTE/SCLK COUNT NUMBER = LLL
1486                            ;*ACTUAL = NNNN
1487                            ;*EXPECTED = NNNN
1488                            ;*
1489                            ;*ECC1 MICRO TEST 01
1490                            ;*ECC1 MICRO ERROR 04
1491                            ;*ECC1-CONTROLLER-NO CORRECTION TEST
1492                            ;*M8951, M8950'S
1493                            ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
1494                            ;*SENT TO THE ECC
1495                            ;*BYTE/SCLK COUNT NUMBER = LLL
1496                            ;*FATAL ERROR - MICRO TEST ABORTED
1497                            ;*
1498                            ;*ECC1 MICRO TEST 01
1499                            ;*ECC1 MICRO ERROR 05
1500                            ;*ECC1-CONTROLLER-NO CORRECTION TEST
1501                            ;*M8951, M8950'S
1502                            ;*CORRECTED DATA FROM THE ECC-INCORRECT
1503                            ;*NO CORRECTION SHOULD HAVE TAKEN PLACE
1504                            ;*BYTE/SCLK COUNT NUMBER = LLL
1505                            ;*ACTUAL = NNNN
1506                            ;*EXPECTED = NNNN
1507                            ;*
1508                            ;*ECC1 MICRO TEST 01
1509                            ;*ECC1 MICRO ERROR 06
1510                            ;*ECC1-CONTROLLER-NO CORRECTION TEST
1511                            ;*M8951, M8950'S
1512                            ;*CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
1513                            ;*BYTE/SCLK COUNT NUMBER = LLL
1514                            ;*
1515                            ;*ECC1 MICRO TEST 01
1516                            ;*ECC1 MICRO ERROR 07
1517                            ;*ECC1-CONTROLLER-NO CORRECTION TEST
1518                            ;*M8951, M8950'S
1519                            ;*CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
1520                            ;*BYTE/SCLK COUNT NUMBER = LLL
1521                            ;*
1522                            ;*ECC1 MICRO TEST 01
1523                            ;*ECC1 MICRO ERROR 10
1524                            ;*ECC1-CONTROLLER-NO CORRECTION TEST
1525                            ;*M8951, M8950'S
1526                            ;*CORRECTED ECC CHARACTER INCORRECT
1527                            ;*BYTE/SCLK COUNT NUMBER = LLL
1528                            ;*
1529                            ;*ECC1 MICRO TEST 01
1530                            ;*ECC1 MICRO ERROR 11
1531                            ;*ECC1-CONTROLLER-NO CORRECTION TEST
1532                            ;*M8951, M8950'S
1533                            ;*ECC STATUS INCORRECT AFTER A DATA GROUP
1534                            ;*BYTE/SCLK COUNT NUMBER = LLL
1535                            ;*ACTUAL = NNNN
```

```
1536                            ;*EXPECTED = NNNN
1537                            ;*
1538                            ;*ECC1 MICRO TEST 01
1539                            ;*ECC1 MICRO ERROR 22
1540                            ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
1541                            ;*M8951, M8950'S
1542                            ;*CHANNEL 0 FAILED
1543                            ;*BYTE/SCLK COUNT NUMBER = LLL
1544                            ;*ACTUAL = NNNN
1545                            ;*EXPECTED = NNNN
1546                            ;*
1547                            ;*ECC1 MICRO TEST 01
1548                            ;*ECC1 MICRO ERROR 23
1549                            ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
1550                            ;*M8951, M8950'S
1551                            ;*CHANNEL 1 FAILED
1552                            ;*BYTE/SCLK COUNT NUMBER = LLL
1553                            ;*ACTUAL = NNNN
1554                            ;*EXPECTED = NNNN
1555                            ;*
1556                            ;*ECC1 MICRO TEST 01
1557                            ;*ECC1 MICRO ERROR 24
1558                            ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
1559                            ;*M8951, M8950'S
1560                            ;*CHANNEL 2 FAILED
1561                            ;*BYTE/SCLK COUNT NUMBER = LLL
1562                            ;*ACTUAL = NNNN
1563                            ;*EXPECTED = NNNN
1564                            ;*
1565                            ;*ECC1 MICRO TEST 01
1566                            ;*ECC1 MICRO ERROR 25
1567                            ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
1568                            ;*M8951, M8950'S
1569                            ;*CHANNEL 3 FAILED
1570                            ;*BYTE/SCLK COUNT NUMBER = LLL
1571                            ;*ACTUAL = NNNN
1572                            ;*EXPECTED = NNNN
1573                            ;*
1574                            ;*ECC1 MICRO TEST 01
1575                            ;*ECC1 MICRO ERROR 26
1576                            ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
1577                            ;*M8951, M8950'S
1578                            ;*CHANNEL 4 FAILED
1579                            ;*BYTE/SCLK COUNT NUMBER = LLL
1580                            ;*ACTUAL = NNNN
1581                            ;*EXPECTED = NNNN
1582                            ;*
1583                            ;*ECC1 MICRO TEST 01
1584                            ;*ECC1 MICRO ERROR 27
1585                            ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
1586                            ;*M8951, M8950'S
1587                            ;*CHANNEL 5 FAILED
1588                            ;*BYTE/SCLK COUNT NUMBER = LLL
1589                            ;*ACTUAL = NNNN
```

```
1590                                    ;*EXPECTED = NNNN
1591                                    ;*
1592                                    ;*ECC1 MICRO TEST 01
1593                                    ;*ECC1 MICRO ERROR 30
1594                                    ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
1595                                    ;*M8951, M8950'S
1596                                    ;*CHANNEL 6 FAILED
1597                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1598                                    ;*ACTUAL = NNNN
1599                                    ;*EXPECTED = NNNN
1600                                    ;*
1601                                    ;*ECC1 MICRO TEST 01
1602                                    ;*ECC1 MICRO ERROR 31
1603                                    ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
1604                                    ;*M8951, M8950'S
1605                                    ;*CHANNEL 7 FAILED
1606                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1607                                    ;*ACTUAL = NNNN
1608                                    ;*EXPECTED = NNNN
1609                                    ;*
1610                                    ;*ECC1 MICRO TEST 01
1611                                    ;*ECC1 MICRO ERROR 32
1612                                    ;*ECC1-ECC CONTROLLER-NO CORRECTION TEST
1613                                    ;*M8951, M8950'S
1614                                    ;*CHANNEL P FAILED
1615                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1616                                    ;*ACTUAL = NNNN
1617                                    ;*EXPECTED = NNNN
1618  4300                              $
 (1)                                    ; ***************************************************************************
1619
1620  4300                              TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1)  4300  3E  01          7.0                  MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28     18.0                  CALL    TSET            ;SETUP THE TEST
1621                                    ;%ECC1-ECC CONTROLLER-NO CORRECTION TEST
1622                                    ;&M8951, M8950'S, M8952
1623  4305                                      REQ     @7,0,0,0,0
 (1)  4305  CD  06  28     18.0                  CALL    REQST
 (1)  4308  00                                   .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4309  00  00                               .WORD   0               ;SYSTEM "0" COUNT
 (1)  430B  00  00                               .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                   .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1)  430E  07                                   .BYTE   @7              ;REQUEST CODE
1624  430F                                      RIN     R12L
 (1)  430F  DB  94         10.0                  IN      R12L            ;READ R12L INTO AC
 (1)  4311  7F              4.0                  MOV     A,A             ;RETRY LINK
1625  4312  32  26  4A     13.0          STA     UNITMP
1626  4315  A7              4.0          ANA     A
1627  4316  C2  27  43     10.0          JNZ     UNITSL
```

```
1629  4319                                        ERR     EXIT,DUMMY
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0001                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                            .BYTE
 (1)  431E  CD  15  28      18.0                DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  92  47      10.0                        JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1630                                        ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1631                                        ;>FATAL ERROR - TEST ABORTED
1632  4324  C3  92  47      10.0                JMP     EXIT
1633
1634  4327  06  00           7.0  UNITSL: MVI     B,@0
1635  4329  3A  26  4A       13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1636  432C  E6  01           7.0           ANI     @01             ;PORT 0?
1637  432E  C2  44  43       10.0           JNZ     FOUND           ;YES-GO USE PORT #0
1638  4331  04               4.0           INR     B               ;NO-UPDATE POINTER TO PORT #1
1639  4332  3A  26  4A       13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1640  4335  E6  02           7.0           ANI     @02             ;PORT 1?
1641  4337  C2  44  43       10.0           JNZ     FOUND           ;YES-GO USE PORT #1
1642  433A  04               4.0           INR     B               ;NO-UPDATE POINTER TO PORT #2
1643  433B  3A  26  4A       13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1644  433E  E6  04           7.0           ANI     @04             ;PORT 2?
1645  4340  C2  44  43       10.0           JNZ     FOUND           ;YES-GO USE PORT #2
1646  4343  04               4.0           INR     B               ;NO-ASSUME PORT #3
1647  4344  CD  95  47       18.0  FOUND:  CALL    CLEAR
1648  4347  DB  E0           10.0           IN      INTSTA
1649  4349  E6  80           7.0           ANI     BIT7
1650  434B  B0               4.0           ORA     B
1651  434C  D3  E0           10.0           OUT     MBSEL
1652  434E  32  27  4A       13.0           STA     SUNIT
1653  4351  3E  04           7.0           MVI     A,P.RPEN
1654  4353  D3  4C           10.0           OUT     PENAB
1655
1656  4355  3E  10           7.0  TST01X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
1657  4357  D3  F0           10.0           OUT     CLKCTL
1658
1659  4359  3E  10           7.0           MVI     A,W.GCR         ;SET GCR MODE
1660  435B  D3  D3           10.0           OUT     WMCCTL
1661
1662  435D  3E  60           7.0           MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
1663  435F  D3  48           10.0           OUT     PDIAG           ;LCS, LWR AND DPEN
1664  4361  AF               4.0           XRA     A
1665  4362  D3  44           10.0           OUT     TAMT
1666  4364  D3  D2           10.0           OUT     TRKENA          ;CLEAR ALL TRACKS FROM
1667  4366  D3  D2           10.0           OUT     TRKENA          ;THE TRANSLATOR
1668  4368  32  24  4A       13.0           STA     INDAT
1669  436B                                   ROUT    R05H
 (1)  436B  D3  8B           10.0                   OUT     R05H            ;WRITE AC INTO R05H
 (1)  436D  7F               4.0                   MOV     A,A             ;RETRY LINK
1670
1671
1672  436E  3E  A8           7.0  TST01C: MVI     A,R.PL01!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1673                                                                       ;SET TIE BUS JAM AND PLO DISABLE
```

```
1674  4370  D3  09        10.0          OUT     RPCTL
1675  4372  AF             4.0          XRA     A
1676  4373  D3  0A        10.0          OUT     RTIEB           ;CLEAR THE TIE BUS
1677  4375  D3  08        10.0          OUT     RFIFOL          ;CLOCK THE FIFO'S
1678  4377  3E  0D         7.0          MVI     A,RCLRT         ;ISSUE CLEAR ALL COMMAND
1679  4379  D3  0B        10.0          OUT     RCMD
1680
1681  437B  00             4.0          NOP                     ;WAIT
1682  437C  00             4.0          NOP                     ;WAIT
1683  437D  00             4.0          NOP
1684  437E  00             4.0          NOP
1685  437F  00             4.0          NOP
1686
1687  4380  3E  A9         7.0          MVI     A,R.PLO1!R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
1688                                                                ;SET TIE BUS JAM AND PLO DISABLE
1689  4382  D3  09        10.0          OUT     RPCTL
1690  4384  CD  50  48     18.0          CALL    CLECC           ;CLEAR THE ECC CHARACTER
1691  4387  16  07         7.0          MVI     D,@7            ;SET UP THE LOOP COUNT
1692  4389  3A  24  4A     13.0  TST01F: LDA     INDAT           ;GET THE INPUT DATA
1693  438C  CD  55  48     18.0          CALL    ECC             ;CALCULATE ECC FOR
1694  438F  15             4.0          DCR     D               ;THE 7 INPUT BYTES
1695  4390  C2  89  43     10.0          JNZ     TST01F
1696
1697  4393  0E  03         7.0          MVI     C,@3            ;INIT THE LOOP COUNT
1698  4395  3A  24  4A     13.0  TST01I: LDA     INDAT           ;GET THE INPUT DATA
1699  4398                               ROUT    RO5L
 (1)  4398  D3  8A        10.0                  OUT     RO5L            ;WRITE AC INTO RO5L
 (1)  439A  7F             4.0                  MOV     A,A             ;RETRY LINK
1700  439B  32  40  48     13.0          STA     TRNIN           ;FILL THE TRANSLATOR
1701  439E  32  41  48     13.0          STA     TRNIN+1         ;SUBROUTINE INPUT
1702  43A1  32  42  48     13.0          STA     TRNIN+2         ;BUFFER
1703  43A4  32  43  48     13.0          STA     TRNIN+3         ;
1704  43A7  CD  C0  47     18.0          CALL    T4X5            ;TRANSLATE THE SUBGROUP
1705
1706  43AA  21  44  48     10.0          LXI     H,TRNOUT        ;GET POINTER TO TRANSLATE DATA TABLE
1707  43AD  06  05         7.0          MVI     B,@5            ;SET UP LOOP COUNT
1708  43AF  7E             7.0  TST01G: MOV     A,M             ;GET A DATA BYTE
1709  43B0  D3  40        10.0          OUT     TCMD            ;STORE DATA IN COMMAND ADDRESS
1710  43B2  23             6.0          INX     H               ;POINT TO DATA PARITY
1711  43B3  7E             7.0          MOV     A,M             ;GET THE DATA PARITY
1712
1713  43B4  07             4.0          RLC                     ;POSITION FOR OUTPUT
1714  43B5  F6  60         7.0          ORI     P.LWR!P.LCS     ;OR IN CONTROL BITS
1715  43B7  D3  48        10.0          OUT     PDIAG           ;OUTPUT THE DATA PARITY
1716
1717  43B9  D3  08        10.0          OUT     RFIFOL          ;CLOCK DATA INTO THE FIFO'S
1718
1719  43BB  23             6.0          INX     H               ;UPDATE THE TABLE POINTER
1720  43BC  05             4.0          DCR     B               ;DECREMENT LOOP COUNT
1721  43BD  C2  AF  43     10.0          JNZ     TST01G          ;DO UNTIL LOOP COUNT = 0
1722
1723  43C0  3A  24  4A     13.0          LDA     INDAT           ;GET THE INPUT DATA
1724  43C3  32  40  48     13.0          STA     TRNIN           ;FILL THE FIRST THREE
1725  43C6  32  41  48     13.0          STA     TRNIN+1         ;BYTES OF THE TRANSLATOR
```

```
1726  43C9  32  42  48   13.0          STA   TRNIN+2          ;SUBROUTINE WITH THE INPUT DATA
1727  43CC  3A  78  48   13.0          LDA   ECCCHR           ;STORE THE CALCULATED
1728  43CF  32  43  48   13.0          STA   TRNIN+3          ;ECC CHARACTER AS THE LAST CHARACTER
1729
1730  43D2  CD  C0  47   18.0          CALL  T4X5             ;TRANSLATE THE SECOND SUBGROUP
1731
1732  43D5  21  44  48   10.0          LXI   H,TRNOUT         ;GET THE POINTER TO INPUT DATA TABLE
1733  43D8  06  05        7.0          MVI   B,@5             ;SET UP THE LOOP COUNT
1734  43DA  7E            7.0  TST01H: MOV   A,M             ;GET A DATA BYTE
1735  43DB  D3  40       10.0          OUT   TCMD             ;STORE DATA IN CMD ADDRESS
1736  43DD  23            6.0          INX   H                ;POINT TO DATA PARITY BIT
1737  43DE  7E            7.0          MOV   A,M              ;GET THE DATA PARITY
1738  43DF  07            4.0          RLC                    ;POSITION FOR OUTPUT
1739  43E0  F6  60        7.0          ORI   P.LWR!P.LCS      ;OR IN CONTROL BITS
1740  43E2  D3  48       10.0          OUT   PDIAG            ;OUTPUT THE DATA PARITY
1741  43E4  D3  08       10.0          OUT   RFIFOL           ;CLOCK DATA INTO THE FIFOS
1742  43E6  23            6.0          INX   H                ;UPDATE THE TABLE POINTER
1743  43E7  05            4.0          DCR   B                ;DECREMENT THE LOOP COUNT
1744  43E8  C2  DA  43    10.0          JNZ   TST01H           ;DO UNTIL LOOP COUNT=0
1745
1746  43EB  0D            4.0          DCR   C                ;PUT THE GROUP IN
1747  43EC  C2  95  43    10.0          JNZ   TST01I           ;THE FIFO TWICE
1748
1749  43EF  3E  0B        7.0          MVI   A,DIARD          ;LOAD THE DIAGNOSTIC READ
1750  43F1  D3  0B       10.0          OUT   RCMD             ;COMMAND
1751  43F3  11  01  00   10.0          LXI   D,1              ;SET WATCH DOG INCREMENT
1752  43F6  21  A8  FD   10.0          LXI   H,-600           ;SET WATCH DOG COUNT TO 600
1753  43F9  D3  0C       10.0  2$:     OUT   RINST            ;STEP THE READ PATH
1754  43FB  DB  01       10.0          IN    RPCHI            ;DATA READY SET?
1755  43FD  E6  10        7.0          ANI   R.DRDY
1756  43FF  C2  14  44    10.0          JNZ   TST01A           ;YES-GO PROCESS
1757  4402  19           10.0          DAD   D                ;WATCH DOG TIMEOUT?
1758  4403  D2  F9  43    10.0          JNC   2$               ;NO-CONTINUE
1759  4406                             ERR   TST01C,DUMMX,1
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4406  CD  09  28    18.0                CALL  ERLP             ;PROCESS ERROR - DO 2.3
 (1)       0002                              MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4409  02                              .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  440A  01                              .BYTE 1
 (1)  440B  CD  15  28    18.0          DUMMX:: CALL  CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  440E  DA  6E  43    10.0                JC    TST01C           ;LOOP ADDRESS IF LOOP SPECIFIED
1760                               ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
1761                               ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1762                               ;<FATAL ERROR - MICRO TEST ABORTED
1763  4411  C3  92  47    10.0          JMP   EXIT
1764
1765  4414  0E  08        7.0  TST01A: MVI   C,$8             ;INIT THE LOOP COUNT
1766  4416  00            4.0  TST01B: NOP
1767  4417  DB  17       10.0          IN    PDATA            ;INPUT THE ACTUAL DATA BYTE
1768  4419  32  2A  4A    13.0          STA   DATAA            ;STORE IT
1769  441C  DB  15       10.0          IN    RPSTA            ;INPUT THE ACTUAL PARITY BIT
1770  441E  E6  40        7.0          ANI   R.DATA           ;REMOVE JUNK BITS
1771  4420  07            4.0          RLC                    ;POSITION THE BIT FOR COMPARE
1772  4421  07            4.0          RLC
```

```
1773  4422  32  2B  4A      13.0          STA     DATAAP          ;STORE IT
1774  4425  3A  24  4A      13.0          LDA     INDAT           ;GET THE EXPECTED DATA BYTE
1775  4428  32  2C  4A      13.0          STA     DATAE           ;STORE IT
1776  442B  A7                4.0          ANA     A               ;SET CONDITION BITS
1777  442C  E2  34  44       10.0          JPO     1$              ;ODD # OF ONES - CLEAR PARITY
1778  442F  3E  01            7.0          MVI     A,@1            ;EVEN # OF ONES - SET PARITY
1779  4431  C3  35  44       10.0          JMP     2$
1780  4434  AF                4.0  1$:     XRA     A
1781  4435  32  2D  4A      13.0  2$:     STA     DATAEP
1782  4438  CD  8B  48      18.0          CALL    CK5X4           ;GO CHECK FOR FAILING READ CHANNELS
1783  443B  DA  6E  43       10.0          JC      TST01C          ;IF LOOP ON ERROR REPEAT THE TEST
1784  443E  0D                4.0          DCR     C               ;DECREMENT THE LOOP COUNT
1785  443F  D3  0C            10.0          OUT     RINST           ;SINGLE STEP THE READ PATH
1786  4441  C2  16  44       10.0          JNZ     TST01B          ;DO UNTIL LOOP COUNT = ZERO
1787  4444  DB  17            10.0          IN      RDATA           ;INPUT THE ACTUAL ECC CHARACTER
1788  4446                                  ROUT    ADATA           ;STORE IN CAS
 (1)  4446  D3  94            10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4448  7F                4.0                  MOV     A,A             ;RETRY LINK
1789  4449  47                4.0          MOV     B,A             ;SAVE IN B
1790  444A  3A  78  48       13.0          LDA     ECCCHR          ;GET THE EXPECTED ECC CHARACTER
1791  444D                                  ROUT    EDATA           ;STORE IN CAS
 (1)  444D  D3  95            10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  444F  7F                4.0                  MOV     A,A             ;RETRY LINK
1792  4450  B8                4.0          CMP     B               ;ACTUAL=EXPECTED?
1793  4451  CA  59  44       10.0          JZ      ECCCK1          ;YES-CONTINUE
1794  4454                                  ERRB    TST01C,ECCCK1,1
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4454  CD  12  28       18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)  0003                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4457  03                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4458  01                                     .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4459  CD  15  28       18.0          ECCCK1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  445C  DA  6E  43       10.0                  JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1795                                  ;>ECC CHARACTER INCORRECT AT OUTPUT OF M8950'S
1796                                  ;<
1797  445F  11  01  00       10.0          LXI     D,1             ;SET WATCHDOG TIMER INCREMENT
1798  4462  21  A8  FD       10.0          LXI     H,-600          ;SET WATCHDOG COUNT TO 600
1799  4465  D3  0C            10.0  3$:     OUT     RINST           ;SINGLE STEP THE READ PATH
1800  4467  DB  01            10.0          IN      RPCHI           ;DATA READY SET?
1801  4469  E6  10             7.0          ANI     R.DRDY
1802  446B  C2  80  44       10.0          JNZ     TST01K          ;YES-GO PROCESS
1803  446E  19                10.0          DAD     D               ;NO-WATCHDOG TIMEOUT?
1804  446F  D2  65  44       10.0          JNC     3$              ;NO-CONTINUE
1805  4472                                  ERR     TST01C,DUMMW,1
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4472  CD  09  28       18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)  0004                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4475  04                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4476  01                                     .BYTE   1
 (1)  4477  CD  15  28       18.0          DUMMW:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  447A  DA  6E  43       10.0                  JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1806                                  ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
1807                                  ;>SENT TO THE ECC
1808                                  ;<FATAL ERROR - MICRO TEST ABORTED
```

```
1809  447D  C3  92  47      10.0          JMP     EXIT
1810
1811  4480  16  01           7.0  TST01K: MVI     D,@1            ;INIT THE LOOP COUNT
1812  4482  AF               4.0  TST01Y: XRA     A
1813  4483                          ROUT    RO5H
 (1)  4483  D3  8B          10.0                  OUT     RO5H            ;WRITE AC INTO RO5H
 (1)  4485  7F               4.0                  MOV     A,A             ;RETRY LINK
1814  4486  3A  24  4A      13.0          LDA     INDAT           ;GET THE INPUT DATA
1815  4489                          ROUT    EDATA
 (1)  4489  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  448B  7F               4.0                  MOV     A,A             ;RETRY LINK
1816  448C  A7               4.0          ANA     A               ;SET CONDITION BITS
1817  448D  3E  00           7.0          MVI     A,0             ;CLEAR THE ACC.
1818  448F  E2  93  44      10.0          JPO     1$              ;ODD # OF ONES - CONTINUE
1819  4492  3C               4.0          INR     A               ;EVEN # OF ONES - SET PARITY
1820  4493  32  25  4A      13.0  1$:     STA     INDATP          ;STORE THE PARITY BIT
1821  4496  D3  0C          10.0  TST01L: OUT     RINST           ;SINGLE STEP THE READ MACHINE
1822  4498  3A  24  4A      13.0          LDA     INDAT
1823  449B  47               4.0          MOV     B,A
1824  449C  DB  19          10.0          IN      ECCCOR
1825  449E  B8               4.0          CMP     B
1826  449F  CA  AE  44      10.0          JZ      TST01M
1827  44A2                          ROUT    ADATA
 (1)  44A2  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44A4  7F               4.0                  MOV     A,A             ;RETRY LINK
1828  44A5  78               4.0          MOV     A,B             ;GET THE BYTE COUNT
1829  44A6                          ROUT    RO5L            ;STORE IN CAS
 (1)  44A6  D3  8A          10.0                  OUT     RO5L            ;WRITE AC INTO RO5L
 (1)  44A8  7F               4.0                  MOV     A,A             ;RETRY LINK
1830  44A9                          ERR3    TST01C,TST01M,@1
 (1)                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44A9  CD  12  28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0005                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44AC  05                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44AD  01                               .BYTE   @1              ;PRINT ROUTINE NUMBER
 (1)  44AE  CD  15  28      18.0          TST01M::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44B1  DA  6E  43      10.0                  JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1831                          ;>CORRECTED DATA FROM THE ECC-INCORRECT
1832                          ;>NO CORRECTION SHOULD HAVE TAKEN PLACE
1833  44B4  DB  15          10.0          IN      RPSTA           ;GET THE ECC PARITY BIT
1834  44B6  E6  80           7.0          ANI     E.CDP           ;GET THE CORRECTED PARITY BIT
1835  44B8  07               4.0          RLC                     ;POSITION BIT FOR COMPARE
1836  44B9  4F               4.0          MOV     C,A             ;SAVE IN REG. C
1837  44BA  3A  25  4A      13.0          LDA     INDATP          ;GET THE EXPECTED PARITY BIT
1838  44BD  A7               4.0          ANA     A               ;SET THE CONDITION BITS
1839  44BE  CA  D9  44      10.0          JZ      TST01QO         ;GO CHECK FOR PARITY=0
1840  44C1  79               4.0          MOV     A,C             ;ELSE CHECK FOR PARITY=1
1841  44C2  FE  01           7.0          CPI     @1
1842  44C4  CA  D0  44      10.0          JZ      TST01RG
1843  44C7  78               4.0          MOV     A,B
1844  44C8                          ROUT    RO5L
 (1)  44C8  D3  8A          10.0                  OUT     RO5L            ;WRITE AC INTO RO5L
 (1)  44CA  7F               4.0                  MOV     A,A             ;RETRY LINK
```

```
1846  44CB                                    ERR     TST01C,TST01RG,@1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44CB  CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0006                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44CE  06                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44CF  01                                        .BYTE   @1
 (1)  44D0  CD  15  28        18.0            TST01RG::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44D3  DA  6E  43        10.0                    JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1847                                    ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
1848                                    ;<
1849
1850  44D6  C3  ED  44        10.0            JMP     TST01SZ         ;CONTINUE WITH TEST
1851  44D9  79                 4.0    TST01QO: MOV     A,C             ;CHECK FOR PARITY = ZERO
1852  44DA  A7                 4.0            ANA     A               ;SET CONDITION BITS
1853  44DB  CA  E7  44        10.0            JZ      TST01TP         ;CONTINUE IF ZERO
1854  44DE  78                 4.0            MOV     A,B
1855  44DF                                   ROUT    R05L
 (1)  44DF  D3  8A            10.0                    OUT     R05L            ;WRITE AC INTO R05L
 (1)  44E1  7F                 4.0                    MOV     A,A             ;RETRY LINK
1856  44E2                                    ERR     TST01C,TST01TP,@1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44E2  CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44E5  07                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44E6  01                                        .BYTE   @1
 (1)  44E7  CD  15  28        18.0            TST01TP::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44EA  DA  6E  45        10.0                    JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1857                                    ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
1858                                    ;<
1859
1860  44ED  7A                 4.0    TST01SZ: MOV     A,D             ;GET THE CHARACTER COUNT
1861  44EE  3C                 4.0             INR     A               ;INCREMENT
1862  44EF  FE  08             7.0             CPI     $8              ;DONE?
1863  44F1  CA  FA  44        10.0             JZ      TST01N          ;YES-GO CHECK ECC
1864  44F4  57                 4.0             MOV     D,A             ;NO-CONTINUE
1865  44F5  00                 4.0             NOP
1866  44F6  00                 4.0             NOP
1867  44F7  C3  82  44        10.0             JMP     TST01Y
1868
1869  44FA  D3  0C             10.0    TST01N: OUT     RINST           ;SINGLE STEP THE READ PATH
1870  44FC  3A  78  48         13.0             LDA     ECCCHR          ;GET THE EXPECTED ECC CHARACTER
1871  44FF                                    ROUT    EDATA           ;SAVE IN CAS
 (1)  44FF  D3  95             10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4501  7F                 4.0                    MOV     A,A             ;RETRY LINK
1872  4502  47                 4.0             MOV     B,A             ;SAVE IN REGISTER B
1873  4503  DB  19             10.0             IN      ECCCOR
1874  4505                                    ROUT    ADATA
 (1)  4505  D3  94             10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4507  7F                 4.0                    MOV     A,A             ;RETRY LINK
1875  4508  B8                 4.0             CMP     B
1876  4509  CA  1B  45        10.0             JZ      TST01U
1877  450C  3A  24  4A         13.0             LDA     INDAT           ;GET THE DATA/COUNT
1878  450F                                    ROUT    R05L
 (1)  450F  D3  8A             10.0                    OUT     R05L            ;WRITE AC INTO R05L
```

```
  (1)  4511  7F                    4.0                    MOV     A,A             ;RETRY LINK
1879   4512  AF                    4.0            XRA     A
1880   4513                                       ROUT    RO5H
  (1)  4513  D3  8B               10.0                    OUT     RO5H            ;WRITE AC INTO RO5H
  (1)  4515  7F                    4.0                    MOV     A,A             ;RETRY LINK
1881   4516                                       ERR     TST01C,TST01U,1
  (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  4516  CD  09  28           18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
  (1)       0008                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4519  08                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  451A  01                                          .BYTE   1
  (1)  451B  CD  15  28           18.0            TST01U::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
  (1)  451E  DA  6E  43           10.0                    JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1882                                       ;>CORRECTED ECC CHARACTER INCORRECT
1883                                       ;<
1884
1885   4521  DB  1A                10.0            IN      ECCSTA          ;GET THE ECC STATUS
1886   4523  E6  0F                 7.0            ANI     @17             ;REMOVE UNWANTED BITS
1887   4525  CA  3A  45            10.0            JZ      TST01V
1888   4528                                       ROUT    ADATA
  (1)  4528  D3  94               10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  452A  7F                    4.0                    MOV     A,A             ;RETRY LINK
1889   452B  AF                    4.0            XRA     A
1890   452C                                       ROUT    EDATA
  (1)  452C  D3  95               10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  452E  7F                    4.0                    MOV     A,A             ;RETRY LINK
1891   452F  3A  24  4A            13.0            LDA     INDAT           ;GET THE DATA/COUNT
1892   4532                                       ROUT    RO5L            ;SAVE IT
  (1)  4532  D3  8A               10.0                    OUT     RO5L            ;WRITE AC INTO RO5L
  (1)  4534  7F                    4.0                    MOV     A,A             ;RETRY LINK
1893   4535                                       ERRB    TST01C,TST01V,1
  (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  4535  CD  12  28           18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0009                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4538  09                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4539  01                                          .BYTE   1               ;PRINT ROUTINE NUMBER
  (1)  453A  CD  15  28           18.0            TST01V::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  453D  DA  6E  43           10.0                    JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1894                                       ;>ECC STATUS INCORRECT AFTER A DATA GROUP
1895                                       ;<
1896
1897   4540  3A  24  4A            13.0            LDA     INDAT
1898   4543  3C                     4.0            INR     A
1899   4544  32  24  4A            13.0            STA     INDAT
1900   4547  C2  6E  43            10.0            JNZ     TST01C
1901   454A                                       ENDTST  TST01X
  (1)                                          ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  454A                                    REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  454A  CD  06  28           18.0                    CALL    REQST
  (2)  454D  00                                          .BYTE                   ;DATA PATTERN NUMBER
  (2)  454E  00  00                                      .WORD                   ;SYSTEM "" COUNT
  (2)  4550  00  00                                      .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  4552  00                                          .BYTE                   ;DATA COMPARE FLAG IF -1
  (2)  4553  07                                          .BYTE   7               ;REQUEST CODE
```

```
   (1)  4554  3A  9A  4F    13.0    LDA   ITERA    ;GET ITERATION COUNT
   (1)  4557  3D              4.0    DCR   A        ;DOWNCOUNT
   (1)  4558  32  9A  4F    13.0    STA   ITERA    ;SAVE COUNT
   (1)  455B  F2  55  43    10.0    JP    TST01X   ;DO TEST UNTIL TILL = 0
1902    455E  CD  95  47    18.0    CALL  CLEAR    ;CLEAR ALL TU PORTS
```

```
                                .SBTTL  TEST 2 - ECC SOFT POINTERS NO BAD DATA
1905  4561                      ST
 (1)                            ; ******************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1906                            ;*ECC SOFT POINTERS NO BAD DATA
1907  4561                      SD
 (1)                            ; ******************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1908                            ;*THIS TEST CHECKS THE ABILITY OF THE ECC CONTROLLER TO PASS GOOD DATA THROUGH
1909                            ;*WITHOUT PERFORMING ANY CORRECTION ON THE DATA, IN SPITE OF THE
1910                            ;*PRESENCE OF SOFT TIE POINTERS.  THE DATA INPUT IS CORRECT
1911                            ;*IN EVERY DETAIL I.E. 5 TO 4 TRANSLATION, PROPER ECC CHARACTER, AND
1912                            ;*DIFFERENT TRACK IN ERROR POINTERS.
1913  4561                      SP
 (1)                            ; ******************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1914                            ;*BGNTST
1915                            ;*    SET NORMAL READ PATH CLOCK
1916                            ;*    CALL SUBROUTINE CLEAR
1917                            ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
1918                            ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
1919                            ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
1920                            ;*    CLEAR THE INPUT DATA
1921                            ;*  BGNDO
1922                            ;*  : ENABLE THE READ PATH CLOCK
1923                            ;*  : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
1924                            ;*  : CLOCK THE FIFO'S
1925                            ;*  : ISSUE CLEAR ALL COMMAND
1926                            ;*  : WAIT
1927                            ;*  : STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
1928                            ;*  : CALL SUBROUTINE CLECC
1929                            ;*  : SET THE LOOP COUNT TO 7
1930                            ;*  : BGNDO
1931                            ;*  : :  GET THE INPUT DATA BYTE
1932                            ;*  : :  CALL SUBROUTINE ECC
1933                            ;*  : :  DECREMENT THE LOOP COUNT
1934                            ;*  : :  DO UNTIL THE LOOP COUNT=0
1935                            ;*  : ENDDO
1936                            ;*  : SET THE LOOP COUNT TO 2
1937                            ;*  : BGNDO
1938                            ;*  : :  FILL THE TRANSLATOR SUBROUTINE BUFFER WITH THE DATA BYTE
1939                            ;*  : :  CALL SUBROUTINE T4X5
1940                            ;*  : :  INIT THE LOOP COUNT TO 5
1941                            ;*  : :  BGNDO
1942                            ;*  : :  :  MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
1943                            ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1944                            ;*  : :  :  MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
1945                            ;*  : :  :  CLOCK THE DATA INTO THE FIFO
1946                            ;*  : :  :  DECREMENT THE LOOP COUNT
1947                            ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1948                            ;*  : :  :  DO UNTIL LOOP COUNT = 0
```

```
1949                                    ;*  : :  ENDDO
1950                                    ;*  : :  FILL THREE BYTES OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE DATA BYTE
1951                                    ;*  : :  FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
1952                                    ;*  : :  CALL SUBROUTINE T4X5
1953                                    ;*  : :  INIT THE LOOP COUNT TO 5
1954                                    ;*  : :  BGNDO
1955                                    ;*  : :  :  MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
1956                                    ;*  : :  :  MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
1957                                    ;*  : :  :  CLOCK THE DATA INTO THE FIFO
1958                                    ;*  : :  :  DECREMENT THE LOOP COUNT
1959                                    ;*  : :  :  DO UNTIL THE LOOP COUNT = 0
1960                                    ;*  : :  ENDDO
1961                                    ;*  : :  DECREMENT THE LOOP COUNT
1962                                    ;*  : :  DO UNTIL THE LOOP COUNT = 0
1963                                    ;*  : ENDDO
1964                                    ;*  : ISSUE DIAGNOSTIC READ COMMAND
1965                                    ;*  : SET UP WATCHDOG TIMER COUNT
1966                                    ;*  : BGNDO
1967                                    ;*  : :    SINGLE STEP THE READ PATH
1968                                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1969                                    ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1970                                    ;*  : ENDDO
1971                                    ;*  : IF WATCHDOG TIMER=0
1972                                    ;*  : :    THEN-ERROR
1973                                    ;*  : :    ELSE-CONTINUE
1974                                    ;*  : ENDIF
1975                                    ;*  : GET THE TIE BUS TABLE ADDRESS
1976                                    ;*  : SINGLE STEP THE READ PATH
1977                                    ;*  : INIT THE LOOP COUNT TO 8
1978                                    ;*  : BGNDO
1979                                    ;*  : :    LOAD THE TIE BUS REGISTER FROM THE TABLE
1980                                    ;*  : :    INCREMENT THE TIE BUS DATA TABLE POINTER
1981                                    ;*  : :    SINGLE STEP THE READ PATH
1982                                    ;*  : :    INPUT THE ACTUAL DATA
1983                                    ;*  : :    INPUT THE ACTUAL PARITY
1984                                    ;*  : :    CALL SUBROUTINE CK5X4
1985                                    ;*  : :    DECREMENT THE LOOP COUNT
1986                                    ;*  : :    DO UNTIL LOOP COUNT = 0
1987                                    ;*  : ENDDO
1988                                    ;*  : LOAD THE TIE BUS REGISTER FROM THE TABLE
1989                                    ;*  : INPUT THE ECC CHARACTER FROM THE READ CHANNELS
1990                                    ;*  : IF ACTUAL ECC CHARACTER=EXPECTED ECC CHARACTER
1991                                    ;*  : :    THEN-CONTINUE
1992                                    ;*  : :    ELSE-ERROR
1993                                    ;*  : ENDIF
1994                                    ;*  : SET UP WATCHDOG TIMER COUNT
1995                                    ;*  : BGNDO
1996                                    ;*  : :  SINGLE STEP THE READ PATH
1997                                    ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
1998                                    ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
1999                                    ;*  : ENDDO
2000                                    ;*  : IF WATCHDOG TIMER=0
2001                                    ;*  : :  THEN-ERROR-EXIT TEST
2002                                    ;*  : :  ELSE-CONTINUE
```

```
                                ;*  : ENDIF
                                ;*  : SINGLE STEP THE READ PATH
                                ;*  : CALL SUBROUTINE CKDONE
                                ;*  : INIT THE  LOOP COUNT TO 7
                                ;*  : BGNDO
                                ;*  : :  SINGLE STEP THE READ PATH
                                ;*  : :  COMPARE CORRECTED DATA WITH INPUT DATA
                                ;*  : :  IF NOT EQUAL
                                ;*  : :  :  THEN-ECC ERROR
                                ;*  : :  :  ELSE-CONTINUE
                                ;*  : :  ENDIF
                                ;*  : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
                                ;*  : :  IF NOT EQUAL
                                ;*  : :  :  THEN-ECC ERROR
                                ;*  : :  :  ELSE-CONTINUE
                                ;*  : :  ENDIF
                                ;*  : :  DECREMENT THE LOOP COUNT
                                ;*  : :  DO UNTIL THE LOOP COUNT = 0
                                ;*  : ENDDO
                                ;*  : SINGLE STEP THE READ PATH
                                ;*  : IF EXPECTED NOT EQUAL
                                ;*  : :  THEN-ECC ERROR
                                ;*  : :  ELSE-CONTINUE
                                ;*  : ENDIF
                                ;*  : INPUT THE ECC STATUS
                                ;*  : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
                                ;*  : IF THE RESULTING STATUS=0
                                ;*  : :  THEN-CONTINUE
                                ;*  : :  ELSE-ERROR
                                ;*  : ENDIF
                                ;*  : INCREMENT THE INPUT DATA
                                ;*  : DO UNTIL INPUT DATA = 0
                                ;*  ENDDO
                                ;*ENDTST
4561                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*ECC1 MICRO TEST 02
                                ;*ECC1 MICRO ERROR 12
                                ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
                                ;*M8951, M8950'S
                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
                                ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
                                ;*BYTE/SCLK COUNT NUMBER = LLL
                                ;*FATAL ERROR - MICRO TEST ABORTED
                                ;*
                                ;*ECC1 MICRO TEST 02
                                ;*ECC1 MICRO ERROR 13
                                ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
                                ;*M8951, M8950'S
                                ;*ECC CHARACTER INCORRECT AT OUTPUT OF M8950'S
                                ;*BYTE/SCLK COUNT NUMBER = LLL
                                ;*ACTUAL = NNNN
```

```
2054                            ;*EXPECTED = NNNN
2055                            ;*
2056                            ;*ECC1 MICRO TEST 02
2057                            ;*ECC1 MICRO ERROR 14
2058                            ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2059                            ;*M8951, M8950'S
2060                            ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
2061                            ;*SENT TO THE ECC
2062                            ;*BYTE/SCLK COUNT NUMBER = LLL
2063                            ;*FATAL ERROR - MICRO TEST ABORTED
2064                            ;*
2065                            ;*ECC1 MICRO TEST 02
2066                            ;*ECC1 MICRO ERROR 15
2067                            ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2068                            ;*M8951, M8950'S
2069                            ;*CORRECTED DATA FROM THE ECC-INCORRECT
2070                            ;*NO CORRECTION SHOULD HAVE TAKEN PLACE
2071                            ;*BYTE/SCLK COUNT NUMBER = LLL
2072                            ;*ACTUAL = NNNN
2073                            ;*EXPECTED = NNNN
2074                            ;*
2075                            ;*ECC1 MICRO TEST 02
2076                            ;*ECC1 MICRO ERROR 16
2077                            ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2078                            ;*M8951, M8950'S
2079                            ;*CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
2080                            ;*BYTE/SCLK COUNT NUMBER = LLL
2081                            ;*
2082                            ;*ECC1 MICRO TEST 02
2083                            ;*ECC1 MICRO ERROR 17
2084                            ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2085                            ;*M8951, M8950'S
2086                            ;*CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
2087                            ;*BYTE/SCLK COUNT NUMBER = LLL
2088                            ;*
2089                            ;*ECC1 MICRO TEST 02
2090                            ;*ECC1 MICRO ERROR 20
2091                            ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2092                            ;*M8951, M8950'S
2093                            ;*CORRECT ECC CHARACTER INCORRECT
2094                            ;*BYTE/SCLK COUNT NUMBER = LLL
2095                            ;*
2096                            ;*ECC1 MICRO TEST 02
2097                            ;*ECC1 MICRO ERROR 21
2098                            ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2099                            ;*M8951, M8950'S
2100                            ;*ECC STATUS INCORRECT AFTER A DATA GROUP
2101                            ;*BYTE/SCLK COUNT NUMBER = LLL
2102                            ;*ACTUAL = NNNN
2103                            ;*EXPECTED = NNNN
2104                            ;*
2105                            ;*ECC1 MICRO TEST 02
2106                            ;*ECC1 MICRO ERROR 22
2107                            ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
```

```
2108                    ;*M8951, M8950'S
2109                    ;*CHANNEL 0 FAILED
2110                    ;*BYTE/SCLK COUNT NUMBER = LLL
2111                    ;*ACTUAL = NNNN
2112                    ;*EXPECTED = NNNN
2113                    ;*
2114                    ;*ECC1 MICRO TEST 02
2115                    ;*ECC1 MICRO ERROR 23
2116                    ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2117                    ;*M8951, M8950'S
2118                    ;*CHANNEL 1 FAILED
2119                    ;*BYTE/SCLK COUNT NUMBER = LLL
2120                    ;*ACTUAL = NNNN
2121                    ;*EXPECTED = NNNN
2122                    ;*
2123                    ;*ECC1 MICRO TEST 02
2124                    ;*ECC1 MICRO ERROR 24
2125                    ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2126                    ;*M8951, M8950'S
2127                    ;*CHANNEL 2 FAILED
2128                    ;*BYTE/SCLK COUNT NUMBER = LLL
2129                    ;*ACTUAL = NNNN
2130                    ;*EXPECTED = NNNN
2131                    ;*
2132                    ;*ECC1 MICRO TEST 02
2133                    ;*ECC1 MICRO ERROR 25
2134                    ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2135                    ;*M8951, M8950'S
2136                    ;*CHANNEL 3 FAILED
2137                    ;*BYTE/SCLK COUNT NUMBER = LLL
2138                    ;*ACTUAL = NNNN
2139                    ;*EXPECTED = NNNN
2140                    ;*
2141                    ;*ECC1 MICRO TEST 02
2142                    ;*ECC1 MICRO ERROR 26
2143                    ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2144                    ;*M8951, M8950'S
2145                    ;*CHANNEL 4 FAILED
2146                    ;*BYTE/SCLK COUNT NUMBER = LLL
2147                    ;*ACTUAL = NNNN
2148                    ;*EXPECTED = NNNN
2149                    ;*
2150                    ;*ECC1 MICRO TEST 02
2151                    ;*ECC1 MICRO ERROR 27
2152                    ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
2153                    ;*M8951, M8950'S
2154                    ;*CHANNEL 5 FAILED
2155                    ;*BYTE/SCLK COUNT NUMBER = LLL
2156                    ;*ACTUAL = NNNN
2157                    ;*EXPECTED = NNNN
2158                    ;*
2159                    ;*ECC1 MICRO TEST 02
2160                    ;*ECC1 MICRO ERROR 30
2161                    ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
```

```
                                        ;*M8951, M8950'S
                                        ;*CHANNEL 6 FAILED
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*ECC1 MICRO TEST 02
                                        ;*ECC1 MICRO ERROR 31
                                        ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
                                        ;*M8951, M8950'S
                                        ;*CHANNEL 7 FAILED
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*ECC1 MICRO TEST 02
                                        ;*ECC1 MICRO ERROR 32
                                        ;*ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
                                        ;*M8951, M8950'S
                                        ;*CHANNEL P FAILED
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4561                                    S
 (1)                                    * ************************************************************************************

                                        TEST2:  TESTX   @2              ;INITIALIZE THE TEST
 (1)  4561  3E  02            7.0                MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4563  CD  03  28       18.0                CALL    TSET            ;SETUP THE TEST
                                        ;%ECC1-ECC CONTROLLER-SOFT POINTERS NO BAD DATA
                                        ;&M8951, M8950'S

4566  3E  10                 7.0  TST02X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
4568  D3  F0                10.0         OUT     CLKCTL
456A  3E  10                 7.0         MVI     A,W.GCR         ;SET GCR MODE
456C  D3  D3                10.0         OUT     WMCCTL
456E  CD  95  47            18.0         CALL    CLEAR           ;CLEAR ALL TU PORTS
4571  3A  27  4A             13.0         LDA     SUNIT
4574  D3  E0                10.0         OUT     MBSEL
4576  3E  04                 7.0         MVI     A,P.RPEN
4578  D3  4C                10.0         OUT     PENAB
457A  3E  60                 7.0         MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
457C  D3  48                10.0         OUT     PDIAG           ;LCS, LWR
457E  AF                     4.0         XRA     A
457F  D3  44                10.0         OUT     TAMT
4581  D3  D2                10.0         OUT     TRKENA          ;CLEAR ALL TRACKS FROM
4583  D3  D2                10.0         OUT     TRKENA          ;CLEAR ALL TRACKS FROM
4585  32  24  4A             13.0         STA     INDAT           ;CLEAR THE INPUT DATA BYTE
4588                                       ROUT    R05H
 (1)  4588  D3  8B            10.0                OUT     R05H            ;WRITE AC INTO R05H
 (1)  458A  7F                 4.0                MOV     A,A             ;RETRY LINK
```

```
2211  458B  3E  A8          7.0  TST02C: MVI   A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
2212                                                                   ;SET TIE BUS JAM AND PLO DISABLE
2213  458D  D3  09         10.0          OUT   RPCTL
2214  458F  D3  08         10.0          OUT   RFIFOL                 ;CLOCK THE FIFO'S
2215  4591      0D          7.0          MVI   A,RCLRT                ;ISSUE CLEAR ALL COMMAND
2216  4593      0B         10.0          OUT   RCMD
2217
2218  4595  00              4.0          NOP                          ;WAIT
2219  4596  00              4.0          NOP
2220  4597  00              4.0          NOP
2221  4598  00              4.0          NOP
2222
2223  4599  3E  A9          7.0          MVI   A,R.PLO1!R.STPC.R.PLOD.R.TBJN   ;STOP THE READ PATH
2224                                                                           ;SET TIE BUS JAM AND PLO DISABLE
2225  459B  D3  09         10.0          OUT   RPCTL
2226  459D  CD  50  48     18.0          CALL  CLECC                  ;CLEAR THE ECC CHARACTER
2227  45A0  16  07          7.0          MVI   D,@7                   ;SET UP THE LOOP COUNT
2228  45A2  3A  24  4A     13.0  TST02F: LDA   INDAT                  ;GET THE INPUT DATA
2229  45A5  CD  55  48     18.0          CALL  ECC                    ;CALCULATE ECC FOR
2230  45A8  15              4.0          DCR   D                      ;THE 7 INPUT BYTES
2231  45A9  C2  A2  45     10.0          JNZ   TST02F
2232
2233  45AC  0E  03          7.0          MVI   C,@3                   ;INIT THE LOOP COUNT
2234  45AE  3A  24  4A     13.0  TST02I: LDA   INDAT                  ;GET THE INPUT DATA
2235  45B1                        ROUT  RO5L
 (1)  45B1  D3  8A         10.0                OUT     RO5L                ;WRITE AC INTO RO5L
 (1)  45B3  7F              4.0                MOV     A,A                 ;RETRY LINK
2236  45B4  32  40  48     13.0          STA   TRNIN                  ;FILL THE TRANSLATOR
2237  45B7  32  41  48     13.0          STA   TRNIN+1                ;SUBROUTINE INPUT
2238  45BA  32  42  48     13.0          STA   TRNIN+2                ;BUFFER
2239  45BD  32  43  48     13.0          STA   TRNIN+3                ;
2240  45C0  CD  C0  47     18.0          CALL  T4X5                   ;TRANSLATE THE SUBGROUP
2241
2242  45C3  21  44  48     10.0          LXI   H,TRNOUT               ;GET POINTER TO TRANSLATE DATA TABLE
2243  45C6  06  05          7.0          MVI   B,@5                   ;SET UP LOOP COUNT
2244  45C8  7E              7.0  TST02G: MOV   A,M                    ;GET A DATA BYTE
2245  45C9  D3  40         10.0          OUT   TCMD                   ;STORE DATA IN COMMAND ADDRESS
2246  45CB  23              6.0          INX   H                      ;POINT TO DATA PARITY
2247  45CC  7E              7.0          MOV   A,M                    ;GET THE DATA PARITY
2248
2249  45CD  07              4.0          RLC                          ;POSITION FOR OUTPUT
2250  45CE  F6  60          7.0          ORI   P.LWR!P.LCS            ;OR IN CONTROL BITS
2251
2252  45D0  D3  48         10.0          OUT   PDIAG                  ;OUTPUT THE DATA PARITY
2253
2254  45D2  D3  08         10.0          OUT   RFIFOL                 ;CLOCK DATA INTO THE FIFO'S
2255
2256  45D4  23              6.0          INX   H                      ;UPDATE THE TABLE POINTER
2257  45D5  05              4.0          DCR   B                      ;DECREMENT LOOP COUNT
2258  45D6  C2  C8  45     10.0          JNZ   TST02G                 ;DO UNTIL LOOP COUNT = 0
2259
2260  45D9  3A  24  4A     13.0          LDA   INDAT                  ;GET THE INPUT DATA
2261  45DC  32  40  48     13.0          STA   TRNIN                  ;FILL THE FIRST THREE
2262  45DF  32  41  48     13.0          STA   TRNIN+1                ;BYTES OF THE TRANSLATOR
```

```
2263  45E2  32  42  48   13.0          STA   TRNIN+2          ;SUBROUTINE WITH THE INPUT DATA
2264  45E5  3A  78  48   13.0          LDA   ECCCHR           ;STORE THE CALCULATED
2265  45E8  32  43  48   13.0          STA   TRNIN+3          ;ECC CHARACTER AS THE LAST CHARACTER
2266
2267  45EB  CD  C0  47   18.0          CALL  T4X5             ;TRANSLATE THE SECOND SUBGROUP
2268
2269  45EE  21  44  48   10.0          LXI   H,TRNOUT         ;GET THE POINTER TO INPUT DATA TABLE
2270  45F1  06  05        7.0          MVI   B,@5             ;SET UP THE LOOP COUNT
2271  45F3  7E            7.0  TST02H: MOV   A,M             ;GET A DATA BYTE
2272  45F4  D3  40       10.0          OUT   TCMD             ;STORE DATA IN CMD ADDRESS
2273  45F6  23            6.0          INX   H                ;POINT TO DATA PARITY BIT
2274  45F7  7E            7.0          MOV   A,M              ;GET THE DATA PARITY
2275  45F8  07            4.0          RLC                    ;POSITION FOR OUTPUT
2276  45F9  F6  60        7.0          ORI   P.LWR!P.LCS      ;OR IN CONTROL BITS
2277  45FB  D3  48       10.0          OUT   PDIAG            ;OUTPUT THE DATA PARITY
2278  45FD  D3  08       10.0          OUT   RFIFOL           ;CLOCK DATA INTO THE FIFOS
2279  45FF  23            6.0          INX   H                ;UPDATE THE TABLE POINTER
2280  4600  05            4.0          DCR   B                ;DECREMENT THE LOOP COUNT
2281  4601  C2  F3  45   10.0          JNZ   TST02H           ;DO UNTIL LOOP COUNT=0
2282
2283  4604  0D            4.0          DCR   C                ;PUT THE GROUP IN
2284  4605  C2  AE  45   10.0          JNZ   TST02I           ;THE FIFO TWICE
2285
2286  4608  3E  0B        7.0          MVI   A,DIARD          ;LOAD THE DIAGNOSTIC READ
2287  460A  D3  0B       10.0          OUT   RCMD             ;COMMAND
2288  460C  11  01  00   10.0          LXI   D,1              ;SET WATCH DOG INCREMENT
2289  460F  21  A8  FD   10.0          LXI   H,-600           ;SET WATCH DOG COUNT TO 600
2290  4612  D3  0C       10.0  2$:     OUT   RINST            ;STEP THE READ PATH
2291  4614  DB  01       10.0          IN    RPCHI            ;DATA READY SET?
2292  4616  E6  10        7.0          ANI   R.DRDY
2293  4618  C2  2D  46   10.0          JNZ   TST02A           ;YES-GO PROCESS
2294  461B  19           10.0          DAD   D                ;WATCH DOG TIMEOUT?
2295  461C  D2  12  46   10.0          JNC   2$               ;NO-CONTINUE
2296  461F                             ERR   TST02C,DUMMU,1
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  461F  CD  09  28   18.0                CALL  ERLP             ;PROCESS ERROR - DO 2.3
 (1)       000A                              MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4622  0A                                .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  4623  01                                .BYTE 1
 (1)  4624  CD  15  28   18.0          DUMMU:: CALL  CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4627  DA  8B  45   10.0                JC    TST02C           ;LOOP ADDRESS IF LOOP SPECIFIED
2297                             ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
2298                             ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
2299                             ;<FATAL ERROR - MICRO TEST ABORTED
2300  462A  C3  92  47   10.0          JMP   EXIT
2301
2302  462D  00            4.0  TST02A: NOP
2303  462E  21  74  49   10.0          LXI   H,TIETBL
2304  4631  3A  24  4A   13.0          LDA   INDAT            ;GET THE INPUT DATA
2305  4634  E6  0F        7.0          ANI   @17              ;GET LO 4 BITS
2306  4636  CA  3F  46   10.0          JZ    2$               ;EXIT IF ZERO
2307  4639  23            6.0  1$:     INX   H                ;UPDATE TABLE POINTER
2308  463A  23            6.0          INX   H                ;
2309  463B  3D            4.0          DCR   A                ;DECREMENT LOOP COUNT
```

```
2310  463C  C2  39  46    10.0          JNZ   1$              ;CONTINUE UNTIL ZERO
2311  463F  7E              7.0  2$:     MOV   A,M             ;GET HIGH BYTE FROM TABLE
2312  4640  57              4.0          MOV   D,A             ;MOVE TO D
2313  4641  23              6.0          INX   H               ;GET LOW BYTE FROM TABLE
2314  4642  7E              7.0          MOV   A,M             ;
2315  4643  5F              4.0          MOV   E,A             ;MOVE TO E
2316  4644  EB              4.0          XCHG                  ;MOV D+E TO H+L
2317  4645  06  08          7.0          MVI   B,$8            ;INIT THE LOOP COUNT
2318  4647  00              4.0  TST02B: NOP
2319  4648  7E              7.0          MOV   A,M             ;GET THE TIE BUS POINTER
2320  4649  D3  0A         10.0          OUT   RTIEB           ;LOAD THE TIE BUS
2321  464B  23              6.0          INX   H               ;UPDATE THE TABLE POINTER
2322  464C  DB  17         10.0          IN    RDATA           ;INPUT THE ACTUAL DATA BYTE
2323  464E  32  2A  4A     13.0          STA   DATAA           ;STORE IT
2324  4651  DB  15         10.0          IN    RPSTA           ;INPUT THE ACTUAL PARITY BIT
2325  4653  E6  40          7.0          ANI   R.DATA          ;REMOVE JUNK BITS
2326  4655  07              4.0          RLC                   ;POSITION THE BIT FOR COMPARE
2327  4656  07              4.0          RLC
2328  4657  32  2B  4A     13.0          STA   DATAAP          ;STORE IT
2329  465A  3A  24  4A     13.0          LDA   INDAT           ;GET THE EXPECTED DATA BYTE
2330  465D  32  2C  4A     13.0          STA   DATAE           ;STORE IT
2331  4660  A7              4.0          ANA   A               ;SET CONDITION BITS
2332  4661  E2  69  46     10.0          JPO   1$              ;ODD # OF ONES - CLEAR PARITY
2333  4664  3E  01          7.0          MVI   A,@1           ;EVEN # OF ONES - SET PARITY
2334  4666  C3  6A  46     10.0          JMP   2$
2335  4669  AF              4.0  1$:     XRA   A
2336  466A  32  2D  4A     13.0  2$:     STA   DATAEP
2337  466D  CD  8B  48     18.0          CALL  CK5X4           ;GO CHECK FOR FAILING READ CHANNELS
2338  4670  DA  8B  45     10.0          JC    TST02C          ;IF LOOP ON ERROR REPEAT THE TEST
2339  4673  05              4.0          DCR   B               ;DECREMENT THE LOOP COUNT
2340  4674  D3  0C         10.0          OUT   RINST           ;SINGLE STEP THE READ PATH
2341  4676  C2  47  46     10.0          JNZ   TST02B          ;DO UNTIL LOOP COUNT = ZERO
2342  4679  7E              7.0          MOV   A,M             ;GET THE LAST POINTER
2343  467A  D3  0A         10.0          OUT   RTIEB           ;LOAD THE TIE BUS
2344  467C  DB  17         10.0          IN    RDATA           ;INPUT THE ACTUAL ECC CHARACTER
2345  467E                               ROUT  ADATA           ;STORE IN CAS
 (1)  467E  D3  94         10.0                OUT   ADATA             ;WRITE AC INTO ADATA
 (1)  4680  7F              4.0                MOV   A,A               ;RETRY LINK
2346  4681  47              4.0          MOV   B,A             ;SAVE IN REG. B
2347  4682  3A  78  48     13.0          LDA   ECCCHR          ;GET THE CORRECT ECC CHARACTER
2348  4685                               ROUT  EDATA           ;STORE IN CAS
 (1)  4685  D3  95         10.0                OUT   EDATA             ;WRITE AC INTO EDATA
 (1)  4687  7F              4.0                MOV   A,A               ;RETRY LINK
2349  4688  B8              4.0          CMP   B               ;ACTUAL=EXPECTED?
2350  4689  CA  91  46     10.0          JZ    ECCCK2          ;YES-CONTINUE
```

```
2352  468C                                    ERRB    TST02C,ECCCK2,1
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  468C  CD  12  28      18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000B                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  468F  0B                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4690  01                                        .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4691  CD  15  28      18.0              ECCCK2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4694  DA  8B  45      10.0                      JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2353                                    ;>ECC CHARACTER INCORRECT AT OUTPUT OF M8950'S
2354                                    ;<
2355  4697  11  01  00        10.0              LXI     D,1                     ;SET WATCHDOG TIMER INCREMENT
2356  469A  21  A8  FD        10.0              LXI     H,-600                  ;SET WATCHDOG COUNT TO 600
2357  469D  D3  0C             10.0  3$:        OUT     RINST                   ;SINGLE STEP THE READ PATH
2358  469F  DB  01             10.0              IN      RPCHI                   ;DATA READY SET?
2359  46A1  E6  10              7.0              ANI     R.DRDY
2360  46A3  C2  B8  46         10.0              JNZ     TST02K                  ;YES-GO PROCESS
2361  46A6  19                  10.0              DAD     D                       ;NO-WATCHDOG TIMEOUT?
2362  46A7  D2  9D  46          10.0              JNC     3$                      ;NO-CONTINUE
2363  46AA                                    ERR     TST02C,DUMMV,1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  46AA  CD  09  28      18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000C                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46AD  0C                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46AE  01                                        .BYTE   1
 (1)  46AF  CD  15  28      18.0              DUMMV:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  46B2  DA  8B  45      10.0                      JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2364                                    ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
2365                                    ;>SENT TO THE ECC
2366                                    ;<FATAL ERROR - MICRO TEST ABORTED
2367  46B5  C3  92  47          10.0              JMP     EXIT
2368
2369  46B8  16  01               7.0  TST02K:    MVI     D,@1                ;INIT THE LOOP COUNT
2370  46BA  AF                   4.0  TST02Y:    XRA     A
2371  46BB                                    ROUT    RO5H
 (1)  46BB  D3  8B             10.0                    OUT     RO5H            ;WRITE AC INTO RO5H
 (1)  46BD  7F                  4.0                    MOV     A,A             ;RETRY LINK
2372  46BE  3A  24  4A          13.0              LDA     INDAT               ;GET THE INPUT DATA
2373  46C1                                    ROUT    EDATA
 (1)  46C1  D3  95             10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  46C3  7F                  4.0                    MOV     A,A             ;RETRY LINK
2374  46C4  A7                   4.0              ANA     A                   ;SET CONDITION BITS
2375  46C5  3E  00               7.0              MVI     A,0                 ;CLEAR THE ACC.
2376  46C7  E2  CB  46          10.0              JPO     1$                  ;ODD # OF ONES - CONTINUE
2377  46CA  3C                   4.0              INR     A                   ;EVEN # OF ONES - SET PARITY
2378  46CB  32  25  4A          13.0  1$:        STA     INDATP              ;STORE THE PARITY BIT
2379  46CE  D3  0C              10.0  TST02L:    OUT     RINST               ;SINGLE STEP THE READ LOGIC
2380  46D0  3A  24  4A          13.0              LDA     INDAT
2381  46D3  47                   4.0              MOV     B,A
2382  46D4  DB  19              10.0              IN      ECCCOR
2383  46D6  B8                   4.0              CMP     B
2384  46D7  CA  E6  46          10.0              JZ      TST02M
2385  46DA                                    ROUT    ADATA
 (1)  46DA  D3  94             10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46DC  7F                  4.0                    MOV     A,A             ;RETRY LINK
```

```
2386  46DD  78                  4.0          MOV     A,B             ;GET THE BYTE COUNT
2387  46DE                                   ROUT    R05L            ;STORE IN CAS
 (1)  46DE  D3  8A              10.0                  OUT     R05L            ;WRITE AC INTO R05L
 (1)  46E0  7F                   4.0                  MOV     A,A             ;RETRY LINK
2388  46E1                                   ERRB    TST02C,TST02M,@1
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46E1  CD  12  28          18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      000D                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46E4  0D                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46E5  01                                      .BYTE   @1              ;PRINT ROUTINE NUMBER
 (1)  46E6  CD  15  28          18.0          TST02M::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46E9  DA  8B  45          10.0                  JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2389                                   ;>CORRECTED DATA FROM THE ECC-INCORRECT
2390                                   ;>NO CORRECTION SHOULD HAVE TAKEN PLACE
2391
2392  46EC  DB  15                10.0          IN      RPSTA           ;GET THE ECC PARITY BIT
2393  46EE  E6  80                 7.0          ANI     E.CDP           ;GET THE CORRECTED PARITY BIT
2394  46F0  07                     4.0          RLC                     ;POSITION BIT FOR COMPARE
2395  46F1  4F                     4.0          MOV     C,A             ;SAVE IN REG. C
2396  46F2  3A  25  4A            13.0          LDA     INDATP          ;GET THE EXPECTED PARITY BIT
2397  46F5  A7                     4.0          ANA     A               ;SET THE CONDITION BITS
2398  46F6  CA  11  47            10.0          JZ      TST02QO         ;GO CHECK FOR PARITY=0
2399  46F9  79                     4.0          MOV     A,C             ;ELSE CHECK FOR PARITY=1
2400  46FA  FE  01                 7.0          CPI     @1
2401  46FC  CA  08  47            10.0          JZ      TST02RG
2402  46FF  78                     4.0          MOV     A,B
2403  4700                                   ROUT    R05L
 (1)  4700  D3  8A              10.0                  OUT     R05L            ;WRITE AC INTO R05L
 (1)  4702  7F                   4.0                  MOV     A,A             ;RETRY LINK
2404  4703                                   ERR     TST02C,TST02RG,@1
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4703  CD  09  28          18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)      000E                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4706  0E                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4707  01                                      .BYTE   @1
 (1)  4708  CD  15  28          18.0          TST02RG::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  470B  DA  8B  45          10.0                  JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2405                                   ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
2406                                   ;<
2407
2408  470E  C3  25  47            10.0          JMP     TST02SZ         ;CONTINUE WITH TEST
2409  4711  79                     4.0  TST02QO: MOV     A,C             ;CHECK FOR PARITY = ZERO
2410  4712  A7                     4.0          ANA     A               ;SET CONDITION BITS
2411  4713  CA  1F  47            10.0          JZ      TST02TP         ;CONTINUE IF ZERO
2412  4716  78                     4.0          MOV     A,B
2413  4717                                   ROUT    R05L
 (1)  4717  D3  8A              10.0                  OUT     R05L            ;WRITE AC INTO R05L
 (1)  4719  7F                   4.0                  MOV     A,A             ;RETRY LINK
```

```
2415  471A                                    ERR     TST02C,TST02TP,@1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  471A  CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000F                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  471D  0F                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  471E  01                                        .BYTE   @1
 (1)  471F  CD  15  28        18.0            TST02TP::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4722  DA  8B  45        10.0                    JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2416                                    ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
2417                                    ;<
2418
2419  4725  7A                 4.0  TST02SZ: MOV     A,D             ;GET THE CHARACTER COUNT
2420  4726  3C                 4.0          INR     A               ;INCREMENT
2421  4727  FE  08             7.0          CPI     $8              ;DONE?
2422  4729  CA  32  47        10.0          JZ      TST02N          ;YES-GO CHECK ECC
2423  472C  57                 4.0          MOV     D,A             ;NO-CONTINUE
2424  472D  00                 4.0          NOP
2425  472E  00                 4.0          NOP
2426  472F  C3  BA  46        10.0          JMP     TST02Y
2427
2428  4732  D3  0C            10.0  TST02N: OUT     RINST           ;SINGLE STEP THE READ PATH
2429  4734  3A  78  48        13.0          LDA     ECCCHR          ;GET THE EXPECTED ECC CHARACTER
2430  4737                               ROUT    EDATA           ;SAVE IN CAS
 (1)  4737  D3  95            10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4739  7F                 4.0                    MOV     A,A             ;RETRY LINK
2431  473A  47                 4.0          MOV     B,A             ;SAVE IN REGISTER B
2432  473B  DB  19            10.0          IN      ECCCOR
2433  473D                               ROUT    ADATA
 (1)  473D  D3  94            10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  473F  7F                 4.0                    MOV     A,A             ;RETRY LINK
2434  4740  B8                 4.0          CMP     B
2435  4741  CA  4F  47        10.0          JZ      TST02U
2436  4744  3A  24  4A        13.0          LDA     INDAT           ;GET THE DATA/COUNT
2437  4747                               ROUT    R05L            ;SAVE IT
 (1)  4747  D3  8A            10.0                    OUT     R05L            ;WRITE AC INTO R05L
 (1)  4749  7F                 4.0                    MOV     A,A             ;RETRY LINK
2438  474A                               ERR     TST02C,TST02U,1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  474A  CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0010                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  474D  10                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  474E  01                                        .BYTE   1
 (1)  474F  CD  15  28        18.0            TST02U::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4752  DA  8B  45        10.0                    JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2439                                    ;>CORRECTED ECC CHARACTER INCORRECT
2440                                    ;<
2441  4755  DB  1A            10.0          IN      ECCSTA          ;GET THE ECC STATUS
2442  4757  E6  0F             7.0          ANI     @17             ;REMOVE UNWANTED BITS
2443  4759  CA  6E  47        10.0          JZ      TST02V
2444  475C                               ROUT    ADATA
 (1)  475C  D3  94            10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  475E  7F                 4.0                    MOV     A,A             ;RETRY LINK
2445  475F  AF                 4.0          XRA     A
2446  4760                               ROUT    EDATA
```

```
  (1)  4760  D3  95        10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  4762  7F             4.0                  MOV     A,A             ;RETRY LINK
2447   4763  3A  24  4A     13.0          LDA     INDAT           ;GET THE DATA/COUNT
2448   4766                               ROUT    RO5L            ;SAVE IT
  (1)  4766  D3  8A        10.0                  OUT     RO5L            ;WRITE AC INTO RO5L
  (1)  4768  7F             4.0                  MOV     A,A             ;RETRY LINK
2449   4769                               ERRB    TST02C,TST02V,1
  (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  4769  CD  12  28    18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0011                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  476C  11                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  476D  01                                        .BYTE   1               ;PRINT ROUTINE NUMBER
  (1)  476E  CD  15  28    18.0                  TST02V::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  4771  DA  8B  45    10.0                          JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2450                                 ;>ECC STATUS INCORRECT AFTER A DATA GROUP
2451                                 ;<
2452
2453   4774  3A  24  4A     13.0          LDA     INDAT
2454   4777  3C                4.0          INR     A
2455   4778  32  24  4A     13.0          STA     INDAT
2456   477B  C2  8B  45     10.0          JNZ     TST02C
2457   477E                               ENDTST  TST02X
  (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  477E                            REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  477E  CD  06  28    18.0                          CALL    REQST
  (2)  4781  00                                        .BYTE                   ;DATA PATTERN NUMBER
  (2)  4782  00  00                                    .WORD                   ;SYSTEM "" COUNT
  (2)  4784  00  00                                    .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  4786  00                                        .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  4787  07                                        .BYTE   7               ;REQUEST CODE
  (1)  4788  3A  9A  4F     13.0          LDA     ITERA           ;GET ITERATION COUNT
  (1)  478B  3D             4.0          DCR     A               ;DOWNCOUNT
  (1)  478C  32  9A  4F     13.0          STA     ITERA           ;SAVE COUNT
  (1)  478F  F2  66  45     10.0          JP      TST02X          ;DO TEST UNTIL TILL = 0
2458   4792  C3  18  28     10.0  EXIT:   JMP     TSTEND
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ; ************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*----------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ; ************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ; ************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ; ************************************************************************

4795  F5                  12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
4796  C5                  12.0          PUSH    B
4797  06  00               7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
4799  DB  E0              10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
479B  E6  80               7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
479D  B0                   4.0          ORA     B               ;ADD IN THE SELECTED PORT #
479E  D3  E0              10.0          OUT     MBSEL           ;RESET TO THIS PORT
47A0  3E  80               7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
47A2  D3  40              10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
47A4  AF                   4.0          XRA     A               ;CLEAR TU COMMAND A
47A5  D3  40              10.0          OUT     TCMD            ;
47A7  3E  81               7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
47A9  D3  40              10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
47AB  3E  00               7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
47AD  D3  40              10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
47AF  AF                   4.0          XRA     A
47B0  D3  44              10.0          OUT     TAMT            ;CLEAR AMTIE WORD
47B2  D3  48              10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
47B4  D3  4C              10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
47B6  04                   4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
47B7  78                   4.0          MOV     A,B
47B8  FE  04               7.0          CPI     4               ;DONE?
47BA  C2  99  47          10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
47BD  C1                  10.0          POP     B               ;RESET B & C
47BE  F1                  10.0          POP     PSW             ;ALL DONE
47BF  C9                  10.0          RET                     ;EXIT
```

```
                                .SBTTL  SUBROUTINE 4X5 TRANSLATE A SUBGROUP
                                SSUB
                                ; ***************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*4X5 TRANSLATE A SUBGROUP
                                SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO 4X5 TRANSLATE A 4 BYTE DATA SUBGROUP INTO
                                ;*A 5 BYTE SUBGROUP AS RECORDED ON TAPE OR READ FROM TAPE.
                                SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*    GET THE 4 (INPUT) DATA BYTES TO BE TRANSLATED
                                ;*    CLEAR THE TRANSLATED (OUTPUT) BUFFER (OUTPUT BUFFER IS 9X5)
                                ;*    SET LOOP COUNT TO 9
                                ;*    BGNDO
                                ;*    :    CALCULATE ODD PARITY FOR THE FOUR (INPUT) DATA BYTES (INPUT BUFFER IS NOW 9X4
                                ;*    :    ROTATE EACH (5) ENTRY OF THE TRANSLATED (OUTPUT) BUFFER RIGHT
                                ;*    :    GET THE LEAST SIGNIFICANT BIT OF EACH (INPUT) DATA BYTE AND
                                ;*    :        FORM A TABLE INDEX (MS BIT OF INDEX = LS BIT OF THE FIRST
                                ;*    :        BYTE OF THE INPUT DATA, ETC.)
                                ;*    :    REMOVE THE LS BIT OF EACH INPUT DATA BYTE (THE ONE JUST USED
                                ;*    :        TO BUILD THE INDEX)
                                ;*    :    USE THE 4 BIT INDEX TO GET THE TRANSLATION VALUE
                                ;*    :    PUT ONE BIT OF THE TRANSLATION VALUE, AS THE MOST SIGNIFICANT
                                ;*    :        BIT EACH OF THE 5 TRANSLATED DATA BYTES (LS BIT OF THE
                                ;*    :        TRANSLATION VALUE REPLACES MS BIT OF THE FIRST TRANSLATED
                                ;*    :        DATA BYTE.
                                ;*    :    DECREMENT THE LOOP COUNT
                                ;*    :    DO UNTIL THE LOOP COUNT = 0
                                ;*    ENDDO
                                ;*ENDSUB
                                ;*
                                S
                                ; ***************************************************************************

47C0                    12.0    T4X5:   PUSH    PSW             ;SAVE REGISTER A + PSW
47C1                    12.0            PUSH    B               ;SAVE REGISTER B + C
47C2                    12.0            PUSH    D               ;SAVE REGISTER D + E
47C3                    12.0            PUSH    H               ;SAVE REGISTER H + L
                                ;FOUR DATA BYTES ARE NOW IS THE BUFFER TABLE
                                ;CLEAR THE TRANSLATED DATA TABLE.
47C4  06  0A              7.0           MVI     B,10            ;SET UP LOOP COUNT
47C6  11  44  48         10.0           LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
47C9  AF                  4.0           XRA     A               ;CLEAR A
47CA  12                  7.0    D4X5:  STAX    D               ;STORE IN THE TABLE
47CB  13                  6.0           INX     D               ;UPDATE TABLE POINTER
47CC  05                  4.0           DCR     B               ;DECREMENT LOOP COUNT
47CD  C2  CA  47         10.0           JNZ     D4X5            ;DO UNTIL LOOP COUNT=0
```

```
2549  47D0  0E  09        7.0          MVI   C,9              ;SET UP TRACK COUNT
2550  47D2  06  04        7.0  B4X5:   MVI   B,4              ;SET UP BIT COUNT
2551  47D4  11  40  48    10.0         LXI   D,TRNIN
2552  47D7  AF             4.0          XRA   A              ;CLEAR THE GROUP POSITION COUNT
2553  47D8  32  4E  48    13.0         STA   GP4X5
2554  47DB  1A             7.0  C4X5:   LDAX  D              ;GET A DATA BYTE
2555  47DC  A7             4.0          ANA   A              ;SET CONDITION BITS
2556  47DD  E2  E1  47     10.0         JPO   PO4X5          ;ODD PARITY LEAVE CARRY CLEAR
2557  47E0  37             4.0          STC                  ;EVEN PARITY SET CARRY
2558  47E1  1F             4.0  PO4X5:  RAR                  ;SHIFT OUT DESIRED BIT
2559  47E2  12             7.0          STAX  D              ;STORE RESULT BACK IN TEMP TABLE
2560  47E3  3A  4E  48     13.0         LDA   GP4X5          ;GET THE GROUP POSITION BYTE
2561  47E6  17             4.0          RAL                  ;PUT IN THIS DATA BIT
2562  47E7  32  4E  48     13.0         STA   GP4X5          ;SAVE THE UPDATED GROUP POSITION BYTE
2563  47EA  13             6.0          INX   D              ;UPDATE THE TABLE POINTER
2564  47EB  05             4.0          DCR   B              ;DECREMENT THE BIT COUNT
2565  47EC  C2  DB  47     10.0         JNZ   C4X5           ;DO UNTIL ALL 4 BITS ARE RECEIVED
2566
2567                                ;AT THIS POINT GP4X5 CONTAINS 4 BITS - ONE FROM EACH OF THE FOUR BYTES
2568
2569                                ;PERFORM A DOUBLE PRECISION SHIFT ON THE TRANSLATED DATA
2570  47EF  06  05        7.0          MVI   B,5              ;SET UP LOOP COUNT
2571  47F1  11  44  48    10.0         LXI   D,TRNOUT         ;GET POINTER TO TRANSLATED DATA TABLE
2572  47F4  13             6.0  E4X5:   INX   D              ;POINT TO PARITY BIT
2573  47F5  1A             7.0          LDAX  D              ;GET PARITY BIT
2574  47F6  A7             4.0          ANA   A              ;CLEAR THE CARRY BIT
2575  47F7  1F             4.0          RAR                  ;SHIFT TO THE CARRY BIT
2576  47F8  12             7.0          STAX  D              ;STORE IT BACK
2577  47F9  1B             6.0          DCX   D              ;DECREMENT TO DATA BITS
2578  47FA  1A             7.0          LDAX  D              ;GET THE DATA BITS
2579  47FB  1F             4.0          RAR                  ;SHIFT IN CARRY BITS
2580  47FC  12             7.0          STAX  D              ;STORE BACK IN TABLE
2581  47FD  13             6.0          INX   D              ;POINT TO NEXT TABLE ENTRY
2582  47FE  13             6.0          INX   D              ;
2583  47FF  05             4.0          DCR   B              ;DECREMENT LOOP COUNT
2584  4800  C2  F4  47     10.0         JNZ   E4X5           ;DO UNTIL LOOP COUNT 0
2585  4803  11  30  48     10.0         LXI   D,TAB4X5       ;LOAD ADDRESS OF TRANSLATION TABLE
2586  4806  26  00         7.0          MVI   H,0            ;CLEAR REGISTER H
2587  4808  3A  4E  48     13.0         LDA   GP4X5          ;GET GROUP POSITION COUNT
2588  480B  6F             4.0          MOV   L,A
2589  480C  19            10.0          DAD   D              ;ADD GROUP POSITION COUNT TO TABLE
2590  480D  7E             7.0          MOV   A,M            ;GET TRANSLATED DATA
2591  480E  32  4F  48     13.0         STA   TRNTMP
2592                                ;"TRNTMP" = THE TRANSLATED DATA
2593
2594  4811  11  44  48     10.0         LXI   D,TRNOUT         ;GET POINTER TO TRANSLATED DATA TABLE
2595  4814  06  05         7.0          MVI   B,5              ;SET UP LOOP COUNT
2596  4816  3A  4F  48     13.0  F4X5:   LDA   TRNTMP          ;GET TRANSLATED DATA
2597  4819  13             6.0          INX   D              ;POINT TO PARITY POSITION
2598  481A  A7             4.0          ANA   A              ;CLEAR CARRY BIT
2599  481B  1F             4.0          RAR                  ;SHIFT OUT A BIT
2600  481C  32  4F  48     13.0         STA   TRNTMP         ;STORE TRANSLATED DATA
2601  481F  1A             7.0          LDAX  D              ;GET THE PARITY BIT (ALWAYS ZERO)
2602  4820  17             4.0          RAL                  ;ROLL IN THE CARRY BIT
```

```
2603  4821  12                7.0          STAX    D              ;STORE AS THE PARITY BIT
2604  4822  13                6.0          INX     D              ;UPDATE TABLE POINTER
2605  4823  05                4.0          DCR     B              ;DECREMENT THE LOOP COUNT
2606  4824  C2  16  48        10.0          JNZ     F4X5           ;DO UNTIL LOOP COUNT=0
2607
2608                                  ;NOW THE TRANSLATED DATA TABLE HAS BEEN UPDATED
2609
2610  4827  0D                4.0          DCR     C              ;DECREMENT THE TRACK COUNT
2611  4828  C2  D2  47        10.0          JNZ     B4X5           ;DO UNTIL ALL TRACKS TRANSLATED
2612  482B  E1                10.0          POP     H              ;RESTORE REGISTER H + L
2613  482C  D1                10.0          POP     D              ;RESTORE REGISTER D + E
2614  482D  C1                10.0          POP     B              ;RESTORE REGISTER B + C
2615  482E  F1                10.0          POP     PSW            ;RESTORE REGISTER A + PSW
2616  482F  C9                10.0          RET                    ;RETURN TO USER
2617                                  .SBTTL  TABLE 4 X 5 TRANSLATION
2618
2619                                  ;THIS TABLE TRANSLATED DATA FOR AS FOLLOWS
2620                                  ;  INPUT GROUP POSITIONS              OUTPUT GROUP POSITIONS
2621                                  ;   1 2 3 4 / 5 6 7 8                 10 9 8 7 6 / 5 4 3 2 1
2622                                  ;
2623                                  ;       0000                              10011
2624                                  ;       0001                              11011
2625                                  ;       0010                              01001
2626                                  ;       0011                              11001
2627                                  ;       0100                              10111
2628                                  ;       0101                              10101
2629                                  ;       0110                              01101
2630                                  ;       0111                              11101
2631                                  ;       1000                              01011
2632                                  ;       1001                              10010
2633                                  ;       1010                              01010
2634                                  ;       1011                              11010
2635                                  ;       1100                              01111
2636                                  ;       1101                              10110
2637                                  ;       1110                              01110
2638                                  ;       1111                              11110
2639
2640  4830  13                      TAB4X5: .BYTE   @23
2641  4831  1B                              .BYTE   @33
2642  4832  09                              .BYTE   @11
2643  4833  19                              .BYTE   @31
2644  4834  17                              .BYTE   @27
2645  4835  15                              .BYTE   @25
2646  4836  0D                              .BYTE   @15
2647  4837  1D                              .BYTE   @35
2648  4838  0B                              .BYTE   @13
2649  4839  12                              .BYTE   @22
2650  483A  0A                              .BYTE   @12
2651  483B  1A                              .BYTE   @32
2652  483C  0F                              .BYTE   @17
2653  483D  16                              .BYTE   @26
2654  483E  0E                              .BYTE   @16
2655  483F  1E                              .BYTE   @36
2656
```

```
2658                                    .SBTTL  SUBROUTINE VARIABLES
2659
2660  4840 0004                         TRNIN:  .BLKB   4               ;TABLE CONTAINING THE 4 DATA BYTES TO
2661                                                                    ;BE TRANSLATED
2662  4844 000A                         TRNOUT: .BLKB   10              ;TRANSLATED DATA TABLE DATA AFTER 4X5
2663                                                                    ;TRANSLATION
2664  484E   00                         GP4X5:  .BYTE   0               ;ACCUMULATOR FOR THE 4 BIT SUBGROUP TO
2665                                                                    ;BE TRANSLATED
2666  484F   00                         TRNTMP: .BYTE   0               ;TEMP. STORAGE FOR THE TRANSLATED SUBGROUP.
2667
2668                                    .SBTTL  SUBROUTINE CLEAR ECC
2669  4850                              $
 (1)                                    ; ************************************************************************
2670                                    ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION "ECCCHR" -
2671                                    ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
2672                                    ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
2673  4850                              $
 (1)                                    ; ************************************************************************
2674
2675  4850  AF                    4.0   CLECC:  XRA     A                 ;CLEAR THE ACCUMULATOR
2676  4851  32  78  48           13.0          STA     ECCCHR            ;CLEAR THE ECC CHARACTER
2677  4854  C9                    10.0          RET                       ;RETURN TO USER
2678
```

```
2680                                    .SBTTL  SUBROUTINE CALCUALTE ECC CHARACTER
2681  4855                              S
 (1)                                    ; ************************************************************************
2682                                    ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
2683                                    ;THE CONTENTS OF "ECCCHR" TO UPDATE THE CONTENTS OF "ECCCHR" ACCORDING
2684                                    ;TO THE ANSI STANDARD ECC POLYNOMIAL.
2685  4855                              S
 (1)                                    ; ************************************************************************
2686
2687  4855  21  78  48      10.0  ECC:    LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
2688  4858  AE                  7.0          XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
2689  4859  5F                  4.0          MOV     E,A             ;SAVE XOR RESULT IN E
2690  485A  E6  10               7.0          ANI     $10             ;IS BIT #4 OF RESULT SET
2691  485C  7B                  4.0          MOV     A,E             ;RESTORE XOR RESULT FROM B
2692  485D  CA  62  48          10.0          JZ      ECC1            ;CONTINUE IF BIT #4 RESET
2693  4860  EE  23               7.0          XRI     $23             ;ELSE-XOR WITH 23
2694  4862  5F                  4.0  ECC1:   MOV     E,A             ;STORE THE ECC RESULT IN E
2695                                                                  ;
2696  4863  AF                  4.0          XRA     A               ;CLEAR A
2697  4864  4F                  4.0          MOV     C,A             ;CLEAR THE TRANSLATE RESULT
2698  4865  21  70  48          10.0          LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
2699  4868  CD  79  48          18.0          CALL    TRANS           ;TRANSLATE THE BITS
2700  486B  79                  4.0          MOV     A,C             ;GET THE TRANSLATED RESULT
2701  486C  32  78  48          13.0          STA     ECCCHR          ;STORE RESULT
2702  486F  C9                  10.0          RET
2703
2704  4870  08                        ECCTBL: $08                     ;BIT 0 = POSITION 3
2705  4871  20                                $20                     ;BIT 1 = POSITION 5
2706  4872  02                                $02                     ;BIT 2 = POSITION 1
2707  4873  40                                $40                     ;BIT 3 = POSITION 6
2708  4874  80                                $80                     ;BIT 4 = POSITION 7
2709  4875  01                                $01                     ;BIT 5 = POSITION 0
2710  4876  10                                $10                     ;BIT 6 = POSITION 4
2711  4877  04                                $04                     ;BIT 7 - POSITION 2
2712
2713  4878  00                        ECCCHR: .BYTE   0
2714
```

```
                                        .SBTTL  SUBROUTINE POLYNOMIAL BIT TRANSLATION

                                        ;
                                        ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
                                        ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
                                        ;

4879  06  01          7.0  TRANS:  MVI   B,1          ;INIT 'B' TO BIT POSITION 0
487B  7B                4.0  TRANS1: MOV   A,E        ;GET CHAR TO BE TRANSLATED
487C  A0                4.0          ANA   B          ;SEE IF BIT POSITION IN 'B' IS SET
487D  CA  83  48       10.0          JZ    TRANS2     ;DO NEXT BIT POSITION IF NOT SET
4880  79                4.0          MOV   A,C        ;GET PREVIOUS RESULT OF 'OR'
4881  B6                7.0          ORA   M          ;'OR' IN NEW POSITION
4882  4F                4.0          MOV   C,A        ;SAVE RESULT

4883  78                4.0  TRANS2: MOV   A,B
4884  07                4.0          RLC              ;POSITION MASK TO NEXT BIT
4885  47                4.0          MOV   B,A
4886  D8               12.0          RC               ;EXIT WHEN ALL POSITIONS DONE
4887  23                6.0          INX   H          ;POINT TO NEXT TABLE ENTRY
4888  C3  7B  48       10.0          JMP   TRANS1     ;PROCESS NEXT BIT
```

```
                                .SBTTL  SUBROUTINE CHECK 5X4
 488B                           SSUB
  (1)                           ;*************************************************************************
  (1)                           ;*SUBROUTINE TITLE
  (1)                           ;*---------- -----
                                ;*CHECK 5X4 SUBROUTINE
 488B                           SD
  (1)                           ;*************************************************************************
  (1)                           ;*DESCRIPTION
  (1)                           ;*-----------
                                ;*THIS SUBROUTINE IS USED TO ISOLATE A BIT FAILURE IN THE READ CHANNEL DATA
                                ;*OR AMTIE BYTES TO A SPECIFIC READ CHANNEL BY COMPARING THE ACTUAL AND
                                ;*EXPECTED INPUTS TO THE SUBROUTINE ON A BIT BY BIT BASIS.
 488B                           SP
  (1)                           ;*************************************************************************
  (1)                           ;*PROCEDURE
  (1)                           ;*---------
                                ;*BGNSUB
                                ;*    IF CHANNEL 0 ACTUAL INFORMATION NOT = CHANNEL 0 EXPECTED
                                ;*      THEN-ERROR CHANNEL 0
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 1 ACTUAL INFORMATION NOT = CHANNEL 1 EXPECTED
                                ;*      THEN-ERROR CHANNEL 1
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 2 ACTUAL INFORMATION NOT = CHANNEL 2 EXPECTED
                                ;*      THEN-ERROR CHANNEL 2
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 3 ACTUAL INFORMATION NOT = CHANNEL 3 EXPECTED
                                ;*      THEN-ERROR CHANNEL 3
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 4 ACTUAL INFORMATION NOT = CHANNEL 4 EXPECTED
                                ;*      THEN-ERROR CHANNEL 4
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 5 ACTUAL INFORMATION NOT = CHANNEL 5 EXPECTED
                                ;*      THEN-ERROR CHANNEL 5
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 6 ACTUAL INFORMATION NOT = CHANNEL 6 EXPECTED
                                ;*      THEN-ERROR CHANNEL 6
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL 7 ACTUAL INFORMATION NOT = CHANNEL 7 EXPECTED
                                ;*      THEN-ERROR CHANNEL 7
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
                                ;*    IF CHANNEL P ACTUAL INFORMATION NOT = CHANNEL P EXPECTED
                                ;*      THEN-ERROR CHANNEL P
                                ;*      ELSE-CONTINUE
                                ;*    ENDIF
```

```
2783                                        ;*ENDSUB
2784  488B                                  S
 (1)                                        ; ************************************************************************
2785  488B  E5                   12.0  CK5X4:  PUSH    H                 ;SAVE H&L
2786  488C  C5                   12.0          PUSH    B                 ;SAVE B&C
2787  488D  21  2C  4A           10.0          LXI     H,DATAE           ;LOAD EXPECTED DATA ADDRESS
2788  4890  3A  2A  4A           13.0          LDA     DATAA             ;GET THE ACTUAL DATA
2789  4893                                    ROUT    ADATA             ;WRITE THE ACTUAL DATA TO THE CAS
 (1)  4893  D3  94                10.0                  OUT     ADATA             ;WRITE AC INTO ADATA
 (1)  4895  7F                     4.0                   MOV     A,A               ;RETRY LINK
2790  4896  E6  01                 7.0          ANI     $01               ;REMOVE UNWANTED BITS
2791  4898  47                     4.0          MOV     B,A               ;SAVE IN B
2792  489B  7E                     7.0          MOV     A,M               ;GET EXPECTED DATA
2793  489A                                    ROUT    EDATA             ;WRITE THE EXPECTED DATA TO THE CAS
 (1)  489A  D3  95                10.0                  OUT     EDATA             ;WRITE AC INTO EDATA
 (1)  489C  7F                     4.0                   MOV     A,A               ;RETRY LINK
2794  489D  E6  01                 7.0          ANI     $01               ;REMOVE UNWANTED BITS
2795  489F  B8                     4.0          CMP     B                 ;COMPARE
2796  48A0  CA  A8  48            10.0          JZ      CKAM0             ;CONTINUE IF EQUAL
2797  48A3                                    ERRB    OUT,CKAM0,1
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  48A3  CD  12  28            18.0                  CALL    ERLPB             ;PROCESS ERROR - DO 2.3
 (1)        0012                                        MSGN    =       MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48A6  12                                          .BYTE   MSGN              ;MESSAGE NUMBER ID
 (1)  48A7  01                                          .BYTE   1                 ;PRINT ROUTINE NUMBER
 (1)  48A8  CD  15  28            18.0          CKAM0:: CALL    CKLOP             ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48AB  DA  71  49            10.0                  JC      OUT               ;LOOP ADDRESS IF LOOP SPECIFIED
2798                                        ;>CHANNEL 0 FAILED
2799  48AE  3A  2A  4A           13.0          LDA     DATAA             ;GET THE ACTUAL DATA
2800  48B1  E6  02                 7.0          ANI     $02               ;REMOVE UNWANTED BITS
2801  48B3  47                     4.0          MOV     B,A               ;SAVE IN B
2802  48B4  7E                     7.0          MOV     A,M               ;GET EXPECTED DATA
2803  48B5  E6  02                 7.0          ANI     $02               ;REMOVE UNWANTED BITS
2804  48B7  B8                     4.0          CMP     B                 ;COMPARE
2805  48B8  CA  C0  48            10.0          JZ      CKAM1             ;CONTINUE IF EQUAL
2806  48BB                                    ERRB    OUT,CKAM1,1
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  48BB  CD  12  28            18.0                  CALL    ERLPB             ;PROCESS ERROR - DO 2.3
 (1)        0013                                        MSGN    =       MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48BE  13                                          .BYTE   MSGN              ;MESSAGE NUMBER ID
 (1)  48BF  01                                          .BYTE   1                 ;PRINT ROUTINE NUMBER
 (1)  48C0  CD  15  28            18.0          CKAM1:: CALL    CKLOP             ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48C3  DA  71  49            10.0                  JC      OUT               ;LOOP ADDRESS IF LOOP SPECIFIED
2807                                        ;>CHANNEL 1 FAILED
2808  48C6  3A  2A  4A           13.0          LDA     DATAA             ;GET THE ACTUAL DATA
2809  48C9  E6  04                 7.0          ANI     $04               ;REMOVE UNWANTED BITS
2810  48CB  47                     4.0          MOV     B,A               ;SAVE IN B
2811  48CC  7E                     7.0          MOV     A,M               ;GET EXPECTED DATA
2812  48CD  E6  04                 7.0          ANI     $04               ;REMOVE UNWANTED BITS
2813  48CF  B8                     4.0          CMP     B                 ;COMPARE
2814  48D0  CA  D8  48            10.0          JZ      CKAM2             ;CONTINUE IF EQUAL
```

```
2816  48D3                                        ERRB    OUT,CKAM2,1
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  48D3  CD  12  28      18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0014                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48D6  14                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48D7  01                                        .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  48D8  CD  15  28      18.0              CKAM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48DB  DA  71  49      10.0                      JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2817                                    ;>CHANNEL 2 FAILED
2818  48DE  3A  2A  4A        13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2819  48E1  E6  08              7.0          ANI     $08             ;REMOVE UNWANTED BITS
2820  48E3  47                  4.0          MOV     B,A             ;SAVE IN B
2821  48E4  7E                  7.0          MOV     A,M             ;GET EXPECTED DATA
2822  48E5  E6  08              7.0          ANI     $08             ;REMOVE UNWANTED BITS
2823  48E7  B8                  4.0          CMP     B               ;COMPARE
2824  48E8  CA  F0  48         10.0          JZ      CKAM3           ;CONTINUE IF EQUAL
2825  48EB                                    ERRB    OUT,CKAM3,1
 (1)                                            ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  48EB  CD  12  28      18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0015                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  48EE  15                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  48EF  01                                        .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  48F0  CD  15  28      18.0              CKAM3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  48F3  DA  71  49      10.0                      JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2826                                    ;>CHANNEL 3 FAILED
2827  48F6  3A  2A  4A        13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2828  48F9  E6  10              7.0          ANI     $10             ;REMOVE UNWANTED BITS
2829  48FB  47                  4.0          MOV     B,A             ;SAVE IN B
2830  48FC  7E                  7.0          MOV     A,M             ;GET EXPECTED DATA
2831  48FD  E6  10              7.0          ANI     $10             ;REMOVE UNWANTED BITS
2832  48FF  B8                  4.0          CMP     B               ;COMPARE
2833  4900  CA  08  49         10.0          JZ      CKAM4           ;CONTINUE IF EQUAL
2834  4903                                    ERRB    OUT,CKAM4,1
 (1)                                            ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4903  CD  12  28      18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0016                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4906  16                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4907  01                                        .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4908  CD  15  28      18.0              CKAM4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  490B  DA  71  49      10.0                      JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2835                                    ;>CHANNEL 4 FAILED
2836
2837  490E  3A  2A  4A        13.0          LDA     DATAA           ;GET THE ACTUAL DATA
2838  4911  E6  20              7.0          ANI     $20             ;REMOVE UNWANTED BITS
2839  4913  47                  4.0          MOV     B,A             ;SAVE IN B
2840  4914  7E                  7.0          MOV     A,M             ;GET EXPECTED DATA
2841  4915  E6  20              7.0          ANI     $20             ;REMOVE UNWANTED BITS
2842  4917  B8                  4.0          CMP     B               ;COMPARE
2843  4918  CA  20  49         10.0          JZ      CKAM5           ;CONTINUE IF EQUAL
```

```
2845  491B                                        ERRB    OUT,CKAM5,1
 (1)
 (1)  491B   CD   12   28        18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0017                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  491E   17                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  491F   01                                          .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4920   CD   15   28        18.0            CKAM5:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4923   DA   71   49        10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2846                                    ;>CHANNEL 5 FAILED
2847
2848  4926   3A   2A   4A        13.0            LDA     DATAA           ;GET THE ACTUAL DATA
2849  4929   E6   40              7.0            ANI     $40             ;REMOVE UNWANTED BITS
2850  492B   47                   4.0            MOV     B,A             ;SAVE IN B
2851  492C   7E                   7.0            MOV     A,M             ;GET EXPECTED DATA
2852  492D   E6   40              7.0            ANI     $40             ;REMOVE UNWANTED BITS
2853  492F   B8                   4.0            CMP     B               ;COMPARE
2854  4930   CA   38   49        10.0            JZ      CKAM6           ;CONTINUE IF EQUAL
2855  4933                                        ERRB    OUT,CKAM6,1
 (1)
 (1)  4933   CD   12   28        18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0018                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4936   18                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4937   01                                          .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4938   CD   15   28        18.0            CKAM6:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  493B   DA   71   49        10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2856                                    ;>CHANNEL 6 FAILED
2857
2858  493E   3A   2A   4A        13.0            LDA     DATAA           ;GET THE ACTUAL DATA
2859  4941   E6   80              7.0            ANI     $80             ;REMOVE UNWANTED BITS
2860  4943   47                   4.0            MOV     B,A             ;SAVE IN B
2861  4944   7E                   7.0            MOV     A,M             ;GET EXPECTED DATA
2862  4945   E6   80              7.0            ANI     $80             ;REMOVE UNWANTED BITS
2863  4947   B8                   4.0            CMP     B               ;COMPARE
2864  4948   CA   50   49        10.0            JZ      CKAM7           ;CONTINUE IF EQUAL
2865  494B                                        ERRB    OUT,CKAM7,1
 (1)
 (1)  494B   CD   12   28        18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0019                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  494E   19                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  494F   01                                          .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4950   CD   15   28        18.0            CKAM7:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4953   DA   71   49        10.0                    JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2866                                    ;>CHANNEL 7 FAILED
2867
2868  4956   21   2D   4A        10.0            LXI     H,DATAEP        ;GET THE EXPECTED DATA PARITY ADDRESS
2869  4959   3A   2B   4A        13.0            LDA     DATAAP          ;GET THE ACTUAL DATA
2870  495C   E6   01              7.0            ANI     $01             ;REMOVE UNWANTED BITS
2871  495E   47                   4.0            MOV     B,A             ;SAVE IN B
2872  495F   7E                   7.0            MOV     A,M             ;GET EXPECTED DATA
2873  4960   E6   01              7.0            ANI     $01             ;REMOVE UNWANTED BITS
2874  4962   B8                   4.0            CMP     B               ;COMPARE
2875  4963   CA   6B   49        10.0            JZ      CKAMP           ;CONTINUE IF EQUAL
```

```
2877  4966                                        ERRB    OUT,CKAMP,1
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4966  CD   12   28      18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        001A                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4969  1A                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  496A  01                                          .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  496B  CD   15   28      18.0              CKAMP:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  496E  DA   71   49      10.0                      JC      OUT             ;LOOP ADDRESS IF LOOP SPECIFIED
2878                                    ;>CHANNEL P FAILED
2879
2880  4971  C1                 10.0  OUT:    POP     B               ;RESTORE B&C
2881  4972  E1                 10.0          POP     H               ;RESTORE H&L
2882  4973  C9                 10.0          RET
2883
```

```
2885                                    .SBTTL  TABLE TIE BUS VECTOR TABLE
2886
2887  4974   49   94                    TIETBL: .WORD   TI0000
2888  4976   49   9D                            .WORD   TI0001
2889  4978   49   A6                            .WORD   TI0010
2890  497A   49   AF                            .WORD   TI0011
2891  497C   49   B8                            .WORD   TI0100
2892  497E   49   C1                            .WORD   TI0101
2893  4980   49   CA                            .WORD   TI0110
2894  4982   49   D3                            .WORD   TI0111
2895  4984   49   DC                            .WORD   TI1000
2896  4986   49   E5                            .WORD   TI1001
2897  4988   49   EE                            .WORD   TI1010
2898  498A   49   F7                            .WORD   TI1011
2899  498C   4A   00                            .WORD   TI1100
2900  498E   4A   09                            .WORD   TI1101
2901  4990   4A   12                            .WORD   TI1110
2902  4992   4A   1B                            .WORD   TI1111
2903
```

```
                                .SBTTL  TABLE TIE BUS DATA VALUES

                        TI0000: .BYTE   $01             ;TIE BUS FOR CLOCK 0
                                .BYTE   $04             ;TIE BUS FOR CLOCK 1
                                .BYTE   $00             ;TIE BUS FOR CLOCK 2
                                .BYTE   $00             ;TIE BUS FOR CLOCK 3
                                .BYTE   $00             ;TIE BUS FOR CLOCK 4
                                .BYTE   $00             ;TIE BUS FOR CLOCK 5
                                .BYTE   $00             ;TIE BUS FOR CLOCK 6
                                .BYTE   $00             ;TIE BUS FOR CLOCK 7
                                .BYTE   $00             ;TIE BUS FOR CLOCK 8

                        TI0001: .BYTE   $00             ;TIE BUS FOR CLOCK 0
                                .BYTE   $00             ;TIE BUS FOR CLOCK 1
                                .BYTE   $01             ;TIE BUS FOR CLOCK 2
                                .BYTE   $04             ;TIE BUS FOR CLOCK 3
                                .BYTE   $00             ;TIE BUS FOR CLOCK 4
                                .BYTE   $00             ;TIE BUS FOR CLOCK 5
                                .BYTE   $00             ;TIE BUS FOR CLOCK 6
                                .BYTE   $00             ;TIE BUS FOR CLOCK 7
                                .BYTE   $00             ;TIE BUS FOR CLOCK 8

                        TI0010: .BYTE   $00             ;TIE BUS FOR CLOCK 0
                                .BYTE   $00             ;TIE BUS FOR CLOCK 1
                                .BYTE   $00             ;TIE BUS FOR CLOCK 2
                                .BYTE   $00             ;TIE BUS FOR CLOCK 3
                                .BYTE   $01             ;TIE BUS FOR CLOCK 4
                                .BYTE   $04             ;TIE BUS FOR CLOCK 5
                                .BYTE   $00             ;TIE BUS FOR CLOCK 6
                                .BYTE   $00             ;TIE BUS FOR CLOCK 7
                                .BYTE   $00             ;TIE BUS FOR CLOCK 8

                        TI0011: .BYTE   $00             ;TIE BUS FOR CLOCK 0
                                .BYTE   $00             ;TIE BUS FOR CLOCK 1
                                .BYTE   $00             ;TIE BUS FOR CLOCK 2
                                .BYTE   $00             ;TIE BUS FOR CLOCK 3
                                .BYTE   $00             ;TIE BUS FOR CLOCK 4
                                .BYTE   $00             ;TIE BUS FOR CLOCK 5
                                .BYTE   $01             ;TIE BUS FOR CLOCK 6
                                .BYTE   $04             ;TIE BUS FOR CLOCK 7
                                .BYTE   $00             ;TIE BUS FOR CLOCK 8

                        TI0100: .BYTE   $00             ;TIE BUS FOR CLOCK 0
                                .BYTE   $00             ;TIE BUS FOR CLOCK 1
                                .BYTE   $00             ;TIE BUS FOR CLOCK 2
                                .BYTE   $00             ;TIE BUS FOR CLOCK 3
                                .BYTE   $00             ;TIE BUS FOR CLOCK 4
                                .BYTE   $00             ;TIE BUS FOR CLOCK 5
                                .BYTE   $00             ;TIE BUS FOR CLOCK 6
                                .BYTE   $01             ;TIE BUS FOR CLOCK 7
                                .BYTE   $04             ;TIE BUS FOR CLOCK 8

                        TI0101: .BYTE   $02             ;TIE BUS FOR CLOCK 0
                                .BYTE   $01             ;TIE BUS FOR CLOCK 1
```

```
2959  49C3  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 2
2960  49C4  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 3
2961  49C5  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 4
2962  49C6  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 5
2963  49C7  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 6
2964  49C8  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 7
2965  49C9  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 8
2966
2967  49CA  00                TI0110:   .BYTE   $00          ;TIE BUS FOR CLOCK 0
2968  49CB  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 1
2969  49CC  02                          .BYTE   $02          ;TIE BUS FOR CLOCK 2
2970  49CD  04                          .BYTE   $04          ;TIE BUS FOR CLOCK 3
2971  49CE  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 4
2972  49CF  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 5
2973  49D0  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 6
2974  49D1  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 7
2975  49D2  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 8
2976
2977  49D3  00                TI0111:   .BYTE   $00          ;TIE BUS FOR CLOCK 0
2978  49D4  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 1
2979  49D5  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 2
2980  49D6  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 3
2981  49D7  02                          .BYTE   $02          ;TIE BUS FOR CLOCK 4
2982  49D8  01                          .BYTE   $01          ;TIE BUS FOR CLOCK 5
2983  49D9  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 6
2984  49DA  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 7
2985  49DB  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 8
2986
2987  49DC  00                TI1000:   .BYTE   $00          ;TIE BUS FOR CLOCK 0
2988  49DD  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 1
2989  49DE  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 2
2990  49DF  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 3
2991  49E0  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 4
2992  49E1  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 5
2993  49E2  02                          .BYTE   $02          ;TIE BUS FOR CLOCK 6
2994  49E3  02                          .BYTE   $02          ;TIE BUS FOR CLOCK 7
2995  49E4  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 8
2996
2997  49E5  00                TI1001:   .BYTE   $00          ;TIE BUS FOR CLOCK 0
2998  49E6  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 1
2999  49E7  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 2
3000  49E8  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 3
3001  49E9  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 4
3002  49EA  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 5
3003  49EB  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 6
3004  49FC  02                          .BYTE   $02          ;TIE BUS FOR CLOCK 7
3005  49ED  02                          .BYTE   $02          ;TIE BUS FOR CLOCK 8
3006
3007  49EE  01                TI1010:   .BYTE   $01          ;TIE BUS FOR CLOCK 0
3008  49EF  02                          .BYTE   $02          ;TIE BUS FOR CLOCK 1
3009  49F0  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 2
3010  49F1  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 3
3011  49F2  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 4
3012  49F3  00                          .BYTE   $00          ;TIE BUS FOR CLOCK 5
```

```
3013  49F4  00                    .BYTE   $00            ;TIE BUS FOR CLOCK 6
3014  49F5  01                    .BYTE   $01            ;TIE BUS FOR CLOCK 7
3015  49F6  04                    .BYTE   $04            ;TIE BUS FOR CLOCK 8
3016
3017  49F7  00              TI1011: .BYTE   $00            ;TIE BUS FOR CLOCK 0
3018  49F8  01                    .BYTE   $01            ;TIE BUS FOR CLOCK 1
3019  49F9  02                    .BYTE   $02            ;TIE BUS FOR CLOCK 2
3020  49FA  00                    .BYTE   $00            ;TIE BUS FOR CLOCK 3
3021  49FB  00                    .BYTE   $00            ;TIE BUS FOR CLOCK 4
3022  49FC  00                    .BYTE   $00            ;TIE BUS FOR CLOCK 5
3023  49FD  04                    .BYTE   $04            ;TIE BUS FOR CLOCK 6
3024  49FE  02                    .BYTE   $02            ;TIE BUS FOR CLOCK 7
3025  49FF  00                    .BYTE   $00            ;TIE BUS FOR CLOCK 8
3026
3027  4A00  01              TI1100: .BYTE   $01            ;TIE BUS FOR CLOCK 0
3028  4A01  02                    .BYTE   $02            ;TIE BUS FOR CLOCK 1
3029  4A02  04                    .BYTE   $04            ;TIE BUS FOR CLOCK 2
3030  4A03  01                    .BYTE   $01            ;TIE BUS FOR CLOCK 3
3031  4A04  02                    .BYTE   $02            ;TIE BUS FOR CLOCK 4
3032  4A05  04                    .BYTE   $04            ;TIE BUS FOR CLOCK 5
3033  4A06  01                    .BYTE   $01            ;TIE BUS FOR CLOCK 6
3034  4A07  02                    .BYTE   $02            ;TIE BUS FOR CLOCK 7
3035  4A08  04                    .BYTE   $04            ;TIE BUS FOR CLOCK 8
3036
3037  4A09  04              TI1101: .BYTE   $04            ;TIE BUS FOR CLOCK 0
3038  4A0A  01                    .BYTE   $01            ;TIE BUS FOR CLOCK 1
3039  4A0B  02                    .BYTE   $02            ;TIE BUS FOR CLOCK 2
3040  4A0C  04                    .BYTE   $04            ;TIE BUS FOR CLOCK 3
3041  4A0D  01                    .BYTE   $01            ;TIE BUS FOR CLOCK 4
3042  4A0E  02                    .BYTE   $02            ;TIE BUS FOR CLOCK 5
3043  4A0F  04                    .BYTE   $04            ;TIE BUS FOR CLOCK 6
3044  4A10  01                    .BYTE   $01            ;TIE BUS FOR CLOCK 7
3045  4A11  02                    .BYTE   $02            ;TIE BUS FOR CLOCK 8
3046
3047  4A12  02              TI1110: .BYTE   $02            ;TIE BUS FOR CLOCK 0
3048  4A13  04                    .BYTE   $04            ;TIE BUS FOR CLOCK 1
3049  4A14  01                    .BYTE   $01            ;TIE BUS FOR CLOCK 2
3050  4A15  02                    .BYTE   $02            ;TIE BUS FOR CLOCK 3
3051  4A16  04                    .BYTE   $04            ;TIE BUS FOR CLOCK 4
3052  4A17  01                    .BYTE   $01            ;TIE BUS FOR CLOCK 5
3053  4A18  02                    .BYTE   $02            ;TIE BUS FOR CLOCK 6
3054  4A19  04                    .BYTE   $04            ;TIE BUS FOR CLOCK 7
3055  4A1A  01                    .BYTE   $01            ;TIE BUS FOR CLOCK 8
3056
3057  4A1B  01              TI1111: .BYTE   $01            ;TIE BUS FOR CLOCK 0
3058  4A1C  02                    .BYTE   $02            ;TIE BUS FOR CLOCK 1
3059  4A1D  04                    .BYTE   $04            ;TIE BUS FOR CLOCK 2
3060  4A1E  01                    .BYTE   $01            ;TIE BUS FOR CLOCK 3
3061  4A1F  02                    .BYTE   $02            ;TIE BUS FOR CLOCK 4
3062  4A20  01                    .BYTE   $01            ;TIE BUS FOR CLOCK 5
3063  4A21  04                    .BYTE   $04            ;TIE BUS FOR CLOCK 6
3064  4A22  02                    .BYTE   $02            ;TIE BUS FOR CLOCK 7
3065  4A23  01                    .BYTE   $01            ;TIE BUS FOR CLOCK 8
3066                      .SBTTL  PROGRAM VARIABLES
```

```
3067
3068  4A24  00                  INDAT:   .BYTE   0          ;DATA BYTE TO ECC
3069  4A25  00                  INDATP:  .BYTE   0          ;DATA BYTE PARITY
3070  4A26  00                  UNITMP:  .BYTE   0          ;UNIT MAP
3071  4A27  00                  SUNIT:   .BYTE   0          ;MB/UNIT SELECT BYTE
3072  4A28  00  00              AMTMSK:  .WORD   0          ,AMTIE MASK WORD
3073  4A2A  00                  DATAA:   .BYTE   0          ;ACTUAL DATA AFTER TRANSLATION
3074  4A2B  00                  DATAAP:  .BYTE   0          ;ACTUAL DATA PARITY AFTER TRANSLATION
3075  4A2C  00                  DATAE:   .BYTE   0          ;EXPECTED DATA AFTER TRANSLATION
3076  4A2D  00                  DATAEP:  .BYTE   0          ;EXPECTED DATA PARITY AFTER TRANSLATION
3077        0000                       .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
AMTMSK   4A28          ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM    4F91          B      =%0000          BADST  = 0090          BIT0   = 0001
BIT1   = 0002          BIT15  = 8000          BIT2   = 0004          BIT3   = 0008
BIT4   = 0010          BIT5   = 0020          BIT6   = 0040          BIT7   = 0080
BIT8   = 0100          BIT9   = 0200          BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60           BRKXCT= 4F00          BSAVE   4F9C          BYTCNT= 00D4
BYTEH    4F24          BYTEL    4F23          B4X5    47D2          C      =%0001
CASCT    4F21          CASCTL= 00A0          CASSTA= 00A0          CATTH  = 0089
CATTL  = 0088          CBUSST= 00A1          CBYTH  = 008B          CBYTL  = 008A
CDG1H  = 0087          CDG1L  = 0086          CDG2H  = 0093          CDG2L  = 0092
CDG3H  = 0095          CDG3L  = 0094          CDVTH  = 008D          CDVTL  = 008C
CHPTIE= 0028           CHOTIE= 0020          CH1TIE= 0021          CH2TIE= 0022
CH3TIE= 0023           CH4TIE= 0024          CH5TIE= 0025          CH6TIE= 0026
CH7TIE= 0027           CKAMP    496B  G      CKAM0    48A8  G      CKAM1    48C0  G
CKAM2    48D8  G       CKAM3    48F0  G      CKAM4    4908  G      CKAM5    4920  G
CKAM6    4938  G       CKAM7    4950  G      CKLOP  = 2815          CK5X4    488B
CLEAR    4795          CLECC    4850          CLKCTL= 00E0          CLOCK    4F26
CLRLP    4799          CMC0H  = 0099          CMC0L  = 0098          CMC1H  = 009B
CMC1L  = 009A          CMC2H  = 009D          CMC2L  = 009C          CMC3H  = 009F
CMC3L  = 009E          CMINH  = 0097          CMINL  = 0096          CNTCTL= 00D7
CRCWRD= 0018           CSAVE    4F9D          CSRLH  = 0091          CSRLL  = 0090
CTCH   = 0085          CTCL   = 0084          CTSTH  = 008F          CTSTL  = 008E
CXCTH  = 0081          CXCTL  = 0080          CXINH  = 0083          CXINL  = 0082
C.     = 0001          C.AVAI= 0080          C.DP   = 0008          C.DSE  = 0010
C.DTU  = 0003          C.DVA  = 0008          C.FAIL= 00FC          C.FMT  = 0070
C.FNCT= 003E           C.GO   = 0001          C.INTC= 00FE          C.MAIN= 0020
C.NSA  = 0080          C.RCT  = 00FC          C.SER  = 0080          C.SHR  = 0040
C.SKPC= 000F           C.TAPE= 0040          C.WCS  = 0002          C4X5     47DB
D      =%0002          DATAA    4A2A          DATAAP   4A2B          DATACT= 00D0
DATAE    4A2C          DATAEP   4A2D          DBUS     4F28          DBUSCT= 00C0
DBUSST= 00C0           DDRA   = 00D8          DDRAIN= 0010          DDRB   = 00D9
DDRBIN= 0002           DDRC   = 00DA          DDRCIN= 0001          DDRCO  = 0088
DDRCTL= 00DB           DIAFLG   4F22          DIAGPG= 4300          DIAGRM= 4F90
DIARD  = 000B          DONE1  = 0045          DONINT= 0010          DSAVE    4F9E
DSE    = 0006          DUMMU    4624  G       DUMMV    46AF  G      DUMMW    4477  G
DUMMX    440B  G       DUMMY    431E  G       D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010           D.LAGC= 0020          D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008           D.WR4  = 0080          D4X5     47CA          E      =%0003
ECC      4855          ECCBAD= 0042          ECCCHR   4878          ECCCK1   4459  G
ECCCK2   4691  G       ECCCOR= 0019          ECCOK  = 0041          ECCSTA= 001A
ECCTBL   4870          ECCTST= 000E          ECC1     4862          EDATA  = 0095
EOTCLR= 0003           ERFLG    4F93          ERLP   = 2809          ERLPA  = 280F
ERLPB  = 2812          ERLPE  = 280C          ERNUM    4F90          ERRCNT= 00D6
ESAVE    4F9F          EXIT     4792          E.ACRC= 0010          E.AMT  = 0020
E.CDP  = 0080          E.CRC  = 0080          E.PNTR= 0008          E.RPE  = 0040
E.STEC= 0001           E.TTEC= 0002          E.UNC  = 0004          E4X5     47F4
FIFORD= 006A           FORMAT   4F25          FOUND    4344          FWDTST= 0061
F4X5     4816          GCRID  = 0089          GCRSET= 0002          GOODTM= 0092
GP4X5    484E          H      =%0004          HLSAVE   4FA0          IE     = 0008
INDAT    4A24          INDATP   4A25          INTSTA= 00E0          ITERA    4F9A
I.PWR  = 0020          I.RMPE= 0040          I5.5   = 0010          I6.5   = 0020
I7.5   = 0040          KCALL  = 005F          KCLR   = 007B          KDEP   = 003F
KENAB  = 0078          KEXAM  = 003E          KEYBRD= 00C8          KEY1   = 0078
```

```
KEY10 = 006D    KEY11 = 006E    KEY12 = 006F    KEY13 = 005C
KEY14 = 005D    KEY15 = 005E    KEY16 = 005F    KEY17 = 003C
KEY18 = 003D    KEY19 = 003E    KEY2  = 0079    KEY20 = 003F
KEY3  = 007A    KEY4  = 007B    KEY5  = 0074    KEY6  = 0075
KEY7  = 0076    KEY8  = 0077    KEY9  = 006C    KINTA = 006F
KLDAD = 003D    KN0   = 003C    KN1   = 005C    KN2   = 005D
KN3   = 005E    KN4   = 006C    KN5   = 006D    KN6   = 006E
KN7   = 0074    KN8   = 0075    KN9   = 0076    KU2   = 0079
KU3   = 007A    KU8   = 0077    L     =%0005    LBLANK= 000F
LCE   = 000B    LCH   = 000C    LCL   = 000D    LCP   = 000E
LC0   = 0000    LC1   = 0001    LC2   = 0002    LC3   = 0003
LC4   = 0004    LC5   = 0005    LC6   = 0006    LC7   = 0007
LC8   = 0008    LC9   = 0009    LDLEDA= 00CA    LDLEDB= 00CB
LDLEDC= 00CC    LDLEDD= 00CD    LDLEDE= 00CE    LDLEDF= 00CF
LKDIAG= 2800    LKKBD = 004C    LKKEY = 0049    LKLWMG= 0058
LKLWMP= 0055    LKLWPG= 0052    LKLWPP= 004F    LKMOD7= 0046
LKOPR = 0046    LPFLG   4F94    LPNUM   4F92    M     =%0006
MBSEL = 00E0    MB.A  = 0008    MB.B  = 0004    MEMTOP= 4FFF
MINUS = 000A    MM    = 8000    MSE   = 0008    MSGN  = 001A
MTACLR= 00C0    MT.ARA= 0020    MT.CPE= 0080    MT.DSE= 0001
MT.FWD= 0040    MT.INH= 0008    MT.LWR= 0004    MT.MOT= 0002
MT.NWT= 0080    MT.PEC= 0040    MT.PSB= 0004    MT.PS0= 0001
MT.PS1= 0002    MT.REV= 0020    MT.WRT= 0010    MT.Z  = 0008
M.ATA = 0080    M.CAPE= 0020    M.CONT= 0080    M.DEM = 0020
M.EBL = 0010    M.EXC = 0008    M.FAIL= 0008    M.ILR = 0010
M.INIT= 0010    M.OCC = 0020    M.ONLI= 0001    M.PE  = 0040
M.PORT= 0080    M.RDEN= 0002    M.RDPE= 0008    M.RUN = 0004
M.SCLK= 0001    M.TRA = 0040    M.UNIT= 0007    M.WCLK= 0040
M.WCLN= 0080    M.WREN= 0080    M5.5  = 0001    M6.5  = 0002
M7.5  = 0004    NOTCAP= 0088    OKAY  = 00FF    OPRRAM= 4300
OPSTRT= 0058    OPVER = 0040    OUT     4971    PADCNT= 00D5
PADCRC= 0080    PDIAG = 0048    PEID  = 008A    PENAB = 004C
PESEI = 0001    PL    = C0B1    PO4X5   47E1    PRDD  = 004C
PRENF = 009C    PS    = 00B2    PSTAT = 0048    PSW   =%0009
P.AMTP= 0001    P.BCTC= 0040    P.CMDP= 0020    P.INTE= 0080
P.LCS = 0040    P.LWR = 0020    P.RDP = 0002    P.RPEN= 0004
P.RPST= 0002    P.RPOE= 0020    P.RP1E= 0010    P.RP2E= 0020
P.RP3E= 0010    P.SING= 0080    P.STAT= 0002    P.STPE= 0080
P.TACH= 0008    P.TUPR= 0010    P.WCSP= 0004    P.WDS = 0040
P.WFLP= 0001    P.WPEN= 0010    P.WPOE= 0008    P.WP1E= 0004
P.WP2E= 0008    P.WP3E= 0004    P.5VOK= 0002    QUE   = 281B
QUEM  = 281E    RAMT  = 0010    RARA  = 0006    RARAI = 0004
RCHBD0= 0048    RCHBD1= 0047    RCHOK = 0046    RCHTST= 000C
RCLRT = 000D    RCMD  = 000B    RCMLP = 0003    RCONT = 0080
RDATA = 0017    RDCLK = 0010    RDON  = 0011    READG = 0007
REND  = 0014    REQST = 2806    RESCHR= 00D1    REVTST= 0064
REWIND= 0004    RFIFOL= 0008    RGCLK = 0002    RGCRI = 0003
RIBG  = 0001    RILL  = 0012    RINST = 00CC    RMCTST= 0008
RMK2  = 0013    RNOP  = 0000    RPATH = 0001    RPBAD = 0044
RPCHI = 0001    RPCLK = 0003    RPCTL = 0009    RPEI  = 0002
RPFAIL= 0000    RPF1  = 009D    RPF2  = 009E    RPOK  = 0043
RPOSTN= 0016    RPSTA = 0015    RRCMT = 000A    RSTAT = 0002
RTIEB = 000A    RTIER = 0030    RTM   = 0005    RUNKI = 0009
RUPTST= 005E    RWDUNL= 0005    R.AMT = 0001    R.BOP   0008
```

```
R.DATA= 0040    R.DON = 0002    R.DRDY= 0010    R.END = 0010
R.ILL = 0004    R.JVOK= 0004    R.MK2 = 0008    R.PLOD= 0008
R.PLO0= 0010    R.PLO1= 0020    R.POST= 0020    R.STNM= 0002
R.STOP= 0004    R.STPC= 0001    R.TBJN= 0080    R.TSTD= 0040
R.VOK = 0080    R00H  = 0081    R00L  = 0080    R01H  = 0083
R01L  = 0082    R02H  = 0085    R02L  = 0084    R03H  = 0087
R03L  = 0086    R04H  = 0089    R04L  = 0088    R05H  = 008B
R05L  = 008A    R06H  = 008D    R06L  = 008C    R07H  = 008F
R07L  = 008E    R10H  = 0091    R10L  = 0090    R11H  = 0093
R11L  = 0092    R12H  = 0095    R12L  = 0094    R13H  = 0097
R13L  = 0096    R14H  = 0099    R14L  = 0098    R15H  = 009B
R15L  = 009A    R16H  = 009D    R16L  = 009C    R17H  = 009F
R17L  = 009E    R7.5  = 0010    SELCLR= 0000    SETATA= 00A1
SID   = 0080    SOD   = 0080    SOE   = 0040    SP    =%0008
SSCLK = 0040    SSTEP = 0005    STACK = 4FFF    STATRM= 4F20
STPCT   4F20    STRSP = 5000    SUNIT   4A27    TAB4X5  4830
TADR00= 0080    TADR01= 0081    TADR02= 0082    TADR03= 0083
TADR04= 0084    TADR05= 0085    TADR06= 0086    TADR07= 0087
TADR10= 0088    TADR11= 0089    TADR12= 008A    TADR13= 008B
TAMT  = 0044    TASEL = 0080    TCMD  = 0040    TC.FWD= 0040
TC.INH= 0008    TC.LWR= 0004    TC.REV= 0020    TC.WRT= 0010
TEMP    4F99    TEST1   4300    TEST2   4561    TIETBL  4974
TI0000  4994    TI0001  499D    TI0010  49A6    TI0011  49AF
TI0100  49B8    TI0101  49C1    TI0110  49CA    TI0111  49D3
TI1000  49DC    TI1001  49E5    TI1010  49EE    TI1011  49F7
TI1100  4A00    TI1101  4A09    TI1110  4A12    TI1111  4A1B
TMF   = 0099    TMRDY = 0040    TRANS   4879    TRANS1  487B
TRANS2  4883    TRKENA= 00D2    TRNIN   4840    TRNOUT  4844
TRNTMP  484F    TSET  = 2803    TSTEND= 2818    TSTS  = 0040
TST01A  4414    TST01B  4416    TST01C  436E    TST01F  4389
TST01G  43AF    TST01H  43DA    TST01I  4395    TST01K  4480
TST01L  4496    TST01M  44AE  G TST01N  44FA    TST01Q  44D9
TST01R  44D0  G TST01S  44ED    TST01T  44E7  G TST01U  451B  G
TST01V  453A  G TST01X  4355    TST01Y  4482    TST02A  462D
TST02B  4647    TST02C  458B    TST02F  45A2    TST02G  45C8
TST02H  45F3    TST02I  45AE    TST02K  46B8    TST02L  46CE
TST02M  46E6  G TST02N  4732    TST02Q  4711    TST02R  4708  G
TST02S  4725    TST02T  471F  G TST02U  474F  G TST02V  476E  G
TST02X  4566    TST02Y  46BA    TUSEL0= 00D1    TUSEL1= 00D2
TU78  = 0010    T.ATH0= 0001    T.ATH1= 0002    T.BOT = 0004
T.EOT = 0002    T.FPT = 0001    T.NTHR= 0004    T.ONL = 0020
T.PES = 0008    T.PSBJ= 0020    T.PSOJ= 0008    T.PS1J= 0010
T.RDY = 0080    T.RDYO= 0040    T.RWD = 0010    T.SCLK= 0002
T4X5    47C0    UIBG  = 00A1    UNITMP  4A26    UNITSL  4327
VALFC   4F98    VALTB   4F95    VELTST= 005B    WDR.P = 0010
WMCCTL= 00D3    WMCERR= 00DA    WMCSTA= 00D0    WRTCLK= 0000
WRTDAT= 00D3    W.ACRC= 0004    W.CRC = 0008    W.DIAG= 0002
W.DONN= 0040    W.ECC = 0010    W.ENAB= 0080    W.ERR = 0020
W.FMT = 0070    W.GCR = 0010    W.LEFT= 0004    W.ONES= 0020
W.RESI= 0002    W.REV = 0004    W.ROME= 0010    W.RST = 0001
W.SKIP= 000F    W.WRIT= 0008    W.XFER= 0020    X     =%000A
X.DONN= 0080    X.ENAB= 0040    X.PEPE= 0002    X.ROME= 0001
X.WCLK= 0001    Y     =%000B    .     = 4A2E
```

ERRORS DETECTED: 0

*ECC1.A78/PTP,ECC1=NLIST,PARAM,MACRO,LIST,ECC1
RUN-TIME: 5 8 0 SECONDS
CORE USED: 10K

3 "MEMORY MAP" EXTENDED ADDRESS BIT
19 READ MICRO CONTROLLER CONTROL & STATUS WORDS
139 TRANSPORT/PORT COMMAND + STATUS WORDS
380 COMMON ADDRESS SPACE REGISTERS
444 MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485 MASSBUS CONTROL WORD DEFINITIONS
557 WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709 KEYBOARD/DISPLAY CONTROL + STATUS WORD
797 MISCELLANEOUS VARIABLE DEFINITIONS
801 RAM ADDRESS DEFINITIONS
819 LINKAGE TABLE ADDRESSES
842 BIT DEFINITIONS
856 RIM AND SIM INSTRUCTION BIT DEFINITIONS
877 INTERRUPTABLE INSTRUCTION MACROS
924 DIAGNOSTIC VARIABLES
945 COMMON VARIABLE STORAGE
959 MACRO CALL DEFINITIONS
1047 TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101 8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1333 TEST 1 - ECC BAD DATA ONE TRACK NO POINTER
1812 TEST 2 - ECC BAD DATA ONE TRACK GOOD POINTER
2279 SUBROUTINE CLEAR ALL TU PORTS
2323 SUBROUTINE 4X5 TRANSLATE A SUBGROUP
2467 SUBROUTINE CALCULATE POINTER
2507 TABLE 4 X 5 TRANSLATION
2548 SUBROUTINE VARIABLES
2562 SUBROUTINE CLEAR ECC
2574 SUBROUTINE CALCUALTE ECC CHARACTER
2610 SUBROUTINE POLYNOMIAL BIT TRANSLATION
2633 SUBROUTINE TRK
2695 PROGRAM VARIABLES

```
                ;*************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;                "BYTE/SCLK COUNT NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                MM = DATA FORMAT FROM CAS REGISTER 2
                ;                NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;                BYTE-SCLK COUNT = LLL
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                LLL = AS ABOVE
                ;                MM  = AS ABOVE
                ;                NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;                TRANSITION COUNT = LLL
                ;                WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;                EXPECTED 18 BITS =      E EEEEEE
                ;                ACTUAL 18 BITS =        A AAAAAA
                ;
                ;                WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;                BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;                OF CAS REG 15 LOW BYTE.
                ;
                ;                ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;                AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;                17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;                TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;                AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;                "SUBGROUP NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       10 -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;*************************************************************************
```

```
1016                    ;*************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ;*************************************************************************
1035                    ;*************************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ;*************************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  ECC2 - ECC CONTROLLER PART #2
                                .SBTTL  TEST 1 - ECC BAD DATA ONE TRACK NO POINTER
                                ;ID     ECC2-ERROR CORRECTION CONTROLLER PART #2
4300                            ST
                                ;*********************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*ECC BAD  TA ONE TRACK NO POINTER
4300                            SD
                                ;*********************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE ECC CONTROLLER TO PERFORM SINGLE
                                ;*TRACK ERROR CORRECTION WITH THE DATA IN A SINGLE TRACK INCORRECT, AND
                                ;*NO POINTERS PROVIDED.  THE PROGRAM CHECKS THAT THE ECC OUTPUT DATA
                                ;*IS CORRECTED, AND THAT THE SINGLE TRACK ERROR CORRECT STATUS BIT IS SET.
                                ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
                                ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
                                ;*MESSAGE, AND TRACK SIMULATED IN ERROR CAN BE DETERMINED FROM THE
                                ;*FOLLOWING TABLE.
                                ;*
                                ;*      INPUT/EXPECTED          DATA BIT POSITION       TRACK WITH ECC
                                ;*          DATA                WITH ECC CORRECTION       CORRECTION
                                ;*      -------------           -------------------     -------------
                                ;*
                                ;*          XXXX0000                    0                   2
                                ;*          XXXX0001                    1                   8
                                ;*          XXXX0010                    2                   1
                                ;*          XXXX0011                    3                   9
                                ;*          XXXX0100                    4                   3
                                ;*          XXXX0101                    5                   5
                                ;*          XXXX0110                    6                   6
                                ;*          XXXX0111                    7                   7
                                ;*          XXXX1000                    P                   4
                                ;*          XXXX1001                    1                   8
                                ;*          XXXX1010                    2                   1
                                ;*          XXXX1011                    3                   9
                                ;*          XXXX1100                    4                   3
                                ;*          XXXX1101                    5                   5
                                ;*          XXXX1110                    6                   6
                                ;*          XXXX1111                    7                   7
                                ;*
                                ;*      X = DON'T CARE BITS
4300                            SP
                                ;*********************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
                                ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
                                ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
                                ;*    CLEAR THE TIE BUS
```

```
1377                    ;*    CLEAR THE INPUT DATA
1378                    ;*    CLEAR BAD TRANSLATION POINTERS
1379                    ;*  BGNDO
1380                    ;*  : ENABLE THE READ PATH CLOCK
1381                    ;*  : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
1382                    ;*  : CLOCK THE FIFO'S
1383                    ;*  : ISSUE CLEAR ALL COMMAND
1384                    ;*  : WAIT
1385                    ;*  : STOP THE READ PATH, SET TIE BUS JAM AND SET PLO DISABLE
1386                    ;*  : CALL SUBROUTINE CLECC
1387                    ;*  : SET THE LOOP COUNT TO 7
1388                    ;*  : BGNDO
1389                    ;*  : :  GET THE INPUT DATA BYTE
1390                    ;*  : :  CALL SUBROUTINE ECC
1391                    ;*  : :  DECREMENT THE LOOP COUNT
1392                    ;*  : :  DO UNTIL THE LOOP COUNT=0
1393                    ;*  : ENDDO
1394                    ;*  : SET THE LOOP COUNT TO 3
1395                    ;*  : BGNDO
1396                    ;*  : :  FILL THE TRANSLATOR SUBROUTINE BUFFER WITH THE DATA BYTE
1397                    ;*  : :  CALCULATE INCORRECT TRANSLATION TRACK
1398                    ;*  : :  STORE POINTER FOR SUBROUTINE
1399                    ;*  : :  CALL SUBROUTINE TRK
1400                    ;*  : :  CALL SUBROUTINE BADT4X5
1401                    ;*  : :  INIT THE LOOP COUNT TO 5
1402                    ;*  : :  BGNDO
1403                    ;*  : :  :  MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
1404                    ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1405                    ;*  : :  :  MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
1406                    ;*  : :  :  CLOCK THE DATA INTO THE FIFO
1407                    ;*  : :  :  DECREMENT THE LOOP COUNT
1408                    ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1409                    ;*  : :  :  DO UNTIL LOOP COUNT = 0
1410                    ;*  : :  ENDDO
1411                    ;*  : :  FILL THREE BYTES OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE DATA BYTE
1412                    ;*  : :  FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
1413                    ;*  : :  CALCULATE INCORRECT TRANSLATION TRACK
1414                    ;*  : :  STORE POINTER FOR SUBROUTINE
1415                    ;*  : :  CALL SUBROUTINE TRK
1416                    ;*  : :  CALL SUBROUTINE BADT4X5
1417                    ;*  : :  INIT THE LOOP COUNT TO 5
1418                    ;*  : :  BGNDO
1419                    ;*  : :  :  MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
1420                    ;*  : :  :  MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
1421                    ;*  : :  :  CLOCK THE DATA INTO THE FIFO
1422                    ;*  : :  :  DECREMENT THE LOOP COUNT
1423                    ;*  : :  :  DO UNTIL THE LOOP COUNT = 0
1424                    ;*  : :  ENDDO
1425                    ;*  : :  DECREMENT THE LOOP COUNT
1426                    ;*  : :  DO UNTIL THE LOOP COUNT = 0
1427                    ;*  : ENDDO
1428                    ;*  : ISSUE DIAGNOSTIC READ COMMAND
1429                    ;*  : SET UP WATCHDOG TIMER COUNT
1430                    ;*  : BGNDO
```

```
1431                    ;*  : :    SINGLE STEP THE READ PATH
1432                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1433                    ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1434                    ;*  : ENDDO
1435                    ;*  : IF WATCHDOG TIMER=0
1436                    ;*  : :    THEN-ERROR
1437                    ;*  : :    ELSE-CONTINUE
1438                    ;*  : ENDIF
1439                    ;*  : INIT THE LOOP COUNT TO 8
1440                    ;*  : BGNDO
1441                    ;*  : :    SINGLE STEP THE READ PATH
1442                    ;*  : :    DECREMENT THE LOOP COUNT
1443                    ;*  : :    DO UNTIL LOOP COUNT = 0
1444                    ;*  : ENDDO
1445                    ;*  : SET UP WATCHDOG TIMER COUNT
1446                    ;*  : BGNDO
1447                    ;*  : :  SINGLE STEP THE READ PATH
1448                    ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
1449                    ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
1450                    ;*  : ENDDO
1451                    ;*  : IF WATCHDOG TIMER=0
1452                    ;*  : :  THEN-ERROR-EXIT TEST
1453                    ;*  : :  ELSE-CONTINUE
1454                    ;*  : ENDIF
1455                    ;*  : CALL SUBROUTINE CKDONE
1456                    ;*  : SINGLE STEP THE READ PATH
1457                    ;*  : INIT THE  LOOP COUNT TO 7
1458                    ;*  : BGNDO
1459                    ;*  : :  SINGLE STEP THE READ PATH
1460                    ;*  : :  COMPARE CORRECTED DATA WITH INPUT DATA
1461                    ;*  : :  IF NOT EQUAL
1462                    ;*  : :  :  THEN-ECC ERROR
1463                    ;*  : :  :  ELSE-CONTINUE
1464                    ;*  : :  ENDIF
1465                    ;*  : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
1466                    ;*  : :  IF NOT EQUAL
1467                    ;*  : :  :  THEN-ECC ERROR
1468                    ;*  : :     ELSE-CONTINUE
1469                    ;*  : :  ENDIF
1470                    ;*  : :  DECREMENT THE LOOP COUNT
1471                    ;*  : :  DO UNTIL THE LOOP COUNT = 0
1472                    ;*  : ENDDO
1473                    ;*  : SINGLE STEP THE READ PATH
1474                    ;*  : IF EXPECTED NOT EQUAL
1475                    ;*  : :  THEN-ECC ERROR
1476                    ;*  : :  ELSE-CONTINUE
1477                    ;*  : ENDIF
1478                    ;*  : INPUT THE ECC STATUS
1479                    ;*  : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
1480                    ;*  : IF THE RESULTING STATUS=SINGLE TRACK ERROR CORRECT + POINTER ERROR
1481                    ;*  : :  THEN-CONTINUE
1482                    ;*  : :  ELSE-ERROR
1483                    ;*  : ENDIF
1484                    ;*  : INCREMENT THE INPUT DATA
```

```
                                ;*  : DO UNTIL INPUT DATA = 0.
                                ;*  ENDDO
                                ;*ENDTST
 4300                           SE
  (1)                           ; ***************************************************************************
  (1)                           ;*ERRORS
  (1)                           ;*------
                                ;*ECC2 MICRO TEST 01
                                ;*ECC2 MICRO ERROR 01
                                ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
                                ;*M8951, M8950'S
                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*ECC2 MICRO TEST 01
                                ;*ECC2 MICRO ERROR 02
                                ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
                                ;*M8951, M8950'S
                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
                                ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
                                ;*BYTE/SCLK COUNT NUMBER = LLL
                                ;*FATAL ERROR - MICRO TEST ABORTED
                                ;*
                                ;*ECC2 MICRO TEST 01
                                ;*ECC2 MICRO ERROR 03
                                ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
                                ;*M8951, M8950'S
                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
                                ;*SENT TO THE ECC
                                ;*BYTE/SCLK COUNT NUMBER = LLL
                                ;*FATAL ERROR - MICRO TEST ABORTED
                                ;*
                                ;*ECC2 MICRO TEST 01
                                ;*ECC2 MICRO ERROR 04
                                ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
                                ;*M8951, M8950'S
                                ;*CORRECTED DATA FROM THE ECC-INCORRECT
                                ;*BYTE/SCLK COUNT NUMBER = LLL
                                ;*ACTUAL = NNNN
                                ;*EXPECTED = NNNN
                                ;*
                                ;*ECC2 MICRO TEST 01
                                ;*ECC2 MICRO ERROR 05
                                ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
                                ;*M8951, M8950'S
                                ;*CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
                                ;*BYTE/SCLK COUNT NUMBER = LLL
                                ;*
                                ;*ECC2 MICRO TEST 01
                                ;*ECC2 MICRO ERROR 06
                                ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
                                ;*M8951, M8950'S
                                ;*CORRECTED DATA PARITY BIT = ONE SHOULD - ZERO
                                ;*BYTE/SCLK COUNT NUMBER = LLL
```

```
1536                                    ;*
1537                                    ;*ECC2 MICRO TEST 01
1538                                    ;*ECC2 MICRO ERROR 07
1539                                    ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
1540                                    ;*M8951, M8950'S
1541                                    ;*CORRECTED ECC CHARACTER INCORRECT
1542                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1543                                    ;*
1544                                    ;*ECC2 MICRO TEST 01
1545                                    ;*ECC2 MICRO ERROR 10
1546                                    ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
1547                                    ;*M8951, M8950'S
1548                                    ;*ECC STATUS INCORRECT AFTER A DATA GROUP
1549                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1550                                    ;*ACTUAL = NNNN
1551                                    ;*EXPECTED = NNNN
1552  4300                              ;
 (1)                                    ;*****************************************************************************
1553
1554  4300                              TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1)  4300  3E  01            7.0                 MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
1555                                    ;%ECC2-ECC CONTROLLER-BAD DATA ONE TRACK NO POINTER
1556                                    ;&M8951, M8950'S
1557
1558  4305                                      REQ     @7,0,0,0,0
 (1)  4305  CD  06  28       18.0                 CALL    REQST
 (1)  4308  00                                    .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4309  00  00                                .WORD   0               ;SYSTEM "0" COUNT
 (1)  430B  00  00                                .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                    .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1)  430E  07                                    .BYTE   @7              ;REQUEST CODE
1559  430F                                RIN     R12L
 (1)  430F  DB  94            10.0                IN      R12L            ;READ R12L INTO AC
 (1)  4311  7F                 4.0                MOV     A,A             ;RETRY LINK
1560  4312  32  B2  48        13.0          STA     UNITMP
1561  4315  A7                 4.0          ANA     A
1562  4316  C2  27  43        10.0          JNZ     UNITSL
1563  4319                                  ERR     EXIT,DUMMY
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28       18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                 .BYTE
 (1)  431E  CD  15  28       18.0          DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  36  47       10.0                 JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1564                                    ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1565                                    ;>FATAL ERROR - TEST ABORTED
1566  4324  C3  36  47        10.0          JMP     EXIT
1567
1568  4327  06  00              7.0    UNITSL: MVI     B,0
1569  4329  3A  B2  48        13.0          LDA     UNITMP          ;DID THE USER SPECIFY TU
1570  432C  E6  01              7.0          ANI     @01             ;PORT 0?
1571  432E  C2  44  43        10.0          JNZ     FOUND           ;YES-GO USE PORT #0
```

```
1572  4331  04              4.0          INR   B              ;NO-UPDATE POINTER TO PORT #1
1573  4332  3A  B2  48     13.0          LDA   UNITMP         ;DID THE USER SPECIFY TU
1574  4335  E6  02           7.0          ANI   @02            ;PORT 1?
1575  4337  C2  44  43     10.0          JNZ   FOUND          ;YES-GO USE PORT #1
1576  433A  04              4.0          INR   B              ;NO-UPDATE POINTER TO PORT #2
1577  433B  3A  B2  48     13.0          LDA   UNITMP         ;DID THE USER SPECIFY TU
1578  433E  E6  04           7.0          ANI   @04            ;PORT 2?
1579  4340  C2  44  43     10.0          JNZ   FOUND          ;YES-GO USE PORT #2
1580  4343  04              4.0          INR   B              ;NO-ASSUME PORT #3
1581  4344  CD  39  47     18.0  FOUND:  CALL  CLEAR
1582  4347  DB  E0         10.0          IN    INTSTA
1583  4349  E6  80           7.0          ANI   BIT7
1584  434B  B0              4.0          ORA   B
1585  434C  D3  E0         10.0          OUT   MBSEL
1586  434E  32  B3  48     13.0          STA   SUNIT
1587
1588  4351  3E  10           7.0  TST01X: MVI   A,RDCLK        ;SET NORMAL READ PATH CLOCKS
1589  4353  D3  F0         10.0          OUT   CLKCTL
1590  4355  3E  10           7.0          MVI   A,W.GCR
1591  4357  D3  D3         10.0          OUT   WMCCTL
1592
1593  4359  3E  04           7.0          MVI   A,P.RPEN
1594  435B  D3  4C         10.0          OUT   PENAB
1595
1596  435D  3E  60           7.0          MVI   A,P.LWR!P.LCS  ;SET THE PORT CONTROL TO
1597  435F  D3  48         10.0          OUT   PDIAG          ;LCS, LWR AND DPEN
1598  4361  AF              4.0          XRA   A
1599  4362  D3  44         10.0          OUT   TAMT
1600  4364  D3  D2         10.0          OUT   TRKENA         ;CLEAR ALL TRACKS FROM
1601  4366  D3  D2         10.0          OUT   TRKENA         ;THE TRANSLATOR
1602  4368  D3  0A         10.0          OUT   RTIEB          ;CLEAR THE TIE BUS
1603  436A  32  B0  48     13.0          STA   INDAT
1604  436D  32  4B  48     13.0          STA   BAD1
1605  4370  32  4C  48     13.0          STA   BAD2
1606  4373  32  4D  48     13.0          STA   BAD3
1607  4376                               ROUT  R05H
 (1)  4376  D3  8B         10.0                OUT   R05H           ;WRITE AC INTO R05H
 (1)  4378  7F              4.0                MOV   A,A            ;RETRY LINK
1608
1609  4379  3E  A8           7.0  TST01C: MVI   A,R.PL01!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1610                                                          ;SET TIE BUS JAM AND PLO DISABLE
1611  437B  D3  09         10.0          OUT   RPCTL
1612  437D  D3  08         10.0          OUT   RFIFOL         ;CLOCK THE FIFO'S
1613  437F  3E  0D           7.0          MVI   A,RCLRT        ;ISSUE CLEAR ALL COMMAND
1614  4381  D3  0B         10.0          OUT   RCMD
1615
1616  4383  00              4.0          NOP
1617  4384  00              4.0          NOP                  ;WAIT
1618  4385  00              4.0          NOP
1619  4386  00              4.0          NOP
1620  4387  00              4.0          NOP
1621
1622  4388  3E  A9           7.0          MVI   A,R.PL01!R.STPC!R.PLOD!R.TBJN  ;STOP THE READ PATH
1623                                                          ;SET TIE BUS JAM AND PLO DISABLE
```

```
1624  438A  D3  09      10.0          OUT     RPCTL
1625  438C  CD  4E  48   18.0          CALL    CLECC           ;CLEAR THE ECC CHARACTER
1626  438F  16  07       7.0          MVI     D,@7            ;SET UP THE LOOP COUNT
1627  4391  3A  B0  48   13.0  TST01F: LDA     INDAT           ;GET THE INPUT DATA
1628  4394  CD  53  48   18.0          CALL    ECC             ;CALCULATE ECC FOR
1629  4397  15           4.0          DCR     D               ;THE 7 INPUT BYTES
1630  4398  C2  91  43   10.0          JNZ     TST01F
1631
1632  439B  0E  03       7.0          MVI     C,@3            ;INIT THE LOOP COUNT
1633  439D  3A  B0  48   13.0  TST01I: LDA     INDAT           ;GET THE INPUT DATA
1634  43A0                             ROUT    RO5L
 (1)  43A0  D3  8A      10.0                  OUT     RO5L            ;WRITE AC INTO RO5L
 (1)  43A2  7F           4.0                  MOV     A,A             ;RETRY LINK
1635  43A3  32  3B  48   13.0          STA     TRNIN           ;FILL THE TRANSLATOR
1636  43A6  32  3C  48   13.0          STA     TRNIN+1         ;SUBROUTINE INPUT
1637  43A9  32  3D  48   13.0          STA     TRNIN+2         ;BUFFER
1638  43AC  32  3E  48   13.0          STA     TRNIN+3         ;
1639  43AF  CD  89  48   18.0          CALL    TRK
1640  43B2  CD  64  47   18.0          CALL    BADT4X5         ;TRANSLATE THE SUBGROUP
1641
1642  43B5  21  3F  48   10.0          LXI     H,TRNOUT        ;GET POINTER TO TRANSLATE DATA TABLE
1643  43B8  06  05       7.0          MVI     B,@5            ;SET UP LOOP COUNT
1644  43BA  7E           7.0  TST01G: MOV     A,M             ;GET A DATA BYTE
1645  43BB  D3  40      10.0          OUT     TCMD            ;STORE DATA IN COMMAND ADDRESS
1646  43BD  23           6.0          INX     H               ;POINT TO DATA PARITY
1647  43BE  7E           7.0          MOV     A,M             ;GET THE DATA PARITY
1648
1649  43BF  07           4.0          RLC                     ;POSITION FOR OUTPUT
1650  43C0  F6  60       7.0          ORI     P.LWR!P.LCS     ;OR IN CONTROL BITS
1651  43C2  D3  48      10.0          OUT     PDIAG           ;OUTPUT THE DATA PARITY
1652
1653  43C4  D3  08      10.0          OUT     RFIFOL          ;CLOCK DATA INTO THE FIFO'S
1654
1655  43C6  23           6.0          INX     H               ;UPDATE THE TABLE POINTER
1656  43C7  05           4.0          DCR     B               ;DECREMENT LOOP COUNT
1657  43C8  C2  BA  43   10.0          JNZ     TST01G          ;DO UNTIL LOOP COUNT = 0
1658
1659  43CB  3A  B0  48   13.0          LDA     INDAT           ;GET THE INPUT DATA
1660  43CE  32  3B  48   13.0          STA     TRNIN           ;FILL THE FIRST THREE
1661  43D1  32  3C  48   13.0          STA     TRNIN+1         ;BYTES OF THE TRANSLATOR
1662  43D4  32  3D  48   13.0          STA     TRNIN+2         ;SUBROUTINE WITH THE INPUT DATA
1663  43D7  3A  76  48   13.0          LDA     ECCCHR          ;STORE THE CALCULATED
1664  43DA  32  3E  48   13.0          STA     TRNIN+3         ;ECC CHARACTER AS THE LAST CHARACTER
1665  43DD  CD  89  48   18.0          CALL    TRK
1666
1667  43E0  CD  64  47   18.0          CALL    BADT4X5         ;TRANSLATE THE SECOND SUBGROUP
1668
1669  43E3  21  3F  48   10.0          LXI     H,TRNOUT        ;GET THE POINTER TO INPUT DATA TABLE
1670  43E6  06  05       7.0          MVI     B,@5            ;SET UP THE LOOP COUNT
1671  43E8  7E           7.0  TST01H: MOV     A,M             ;GET A DATA BYTE
1672  43E9  D3  40      10.0          OUT     TCMD            ;STORE DATA IN CMD ADDRESS
1673  43EB  23           6.0          INX     H               ;POINT TO DATA PARITY BIT
1674  43EC  7E           7.0          MOV     A,M             ;GET THE DATA PARITY
1675  43ED  07           4.0          RLC                     ;POSITION FOR OUTPUT
```

```
1676  43EE  F6  60        7.0          ORI   P.LWR.P.LCS        ;OR IN CONTROL BITS
1677  43F0  D3  48       10.0          OUT   PDIAG              ;OUTPUT THE DATA PARITY
1678  43F2  D3  08       10.0          OUT   RFIFOL             ;CLOCK DATA INTO THE FIFOS
1679  43F4  23            6.0          INX   H                  ;UPDATE THE TABLE POINTER
1680  43F5  05            4.0          DCR   B                  ;DECREMENT THE LOOP COUNT
1681  43F6  C2  E8  43   10.0          JNZ   TST01H             ;DO UNTIL LOOP COUNT=0
1682
1683  43F9  0D            4.0          DCR   C                  ;PUT THE GROUP IN
1684  43FA  C2  9D  43   10.0          JNZ   TST011             ;THE FIFO TWICE
1685
1686  43FD  3E  0B        7.0          MVI   A,DIARD            ;LOAD THE DIAGNOSTIC READ
1687  43FF  D3  0B       10.0          OUT   RCMD               ;COMMAND
1688  4401  11  01  00   10.0          LXI   D,1                ;SET WATCH DOG INCREMENT
1689  4404  21  A8  FD   10.0          LXI   H,-600             ;SET WATCH DOG COUNT TO 600
1690  4407  D3  0C       10.0  2$:     OUT   RINST              ;STEP THE READ PATH
1691  4409  DB  01       10.0          IN    RPCHI              ;DATA READY SET?
1692  440B  E6  10        7.0          ANI   R.DRDY
1693  440D  C2  22  44   10.0          JNZ   TST01A             ;YES-GO PROCESS
1694  4410  19           10.0          DAD   D                  ;WATCH DOG TIMEOUT?
1695  4411  D2  07  44   10.0          JNC   2$                 ;NO-CONTINUE
1696  4414                             ERR   TST01C,DUMMX,1
 (1)                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4414  CD  09  28   18.0                CALL  ERLP               ;PROCESS ERROR - DO 2.3
 (1)       0002                     MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4417  02                      .BYTE MSGN               ;MESSAGE NUMBER ID
 (1)  4418  01                      .BYTE 1
 (1)  4419  CD  15  28   18.0  DUMMX:: CALL  CKLOP              ;CHECK LOOP FUNCTION - DO 2.3
 (1)  441C  DA  79  43   10.0          JC    TST01C             ;LOOP ADDRESS IF LOOP SPECIFIED
1697                             ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
1698                             ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1699                             ;<FATAL ERROR - MICRO TEST ABORTED
1700  441F  C3  36  47   10.0          JMP   EXIT
1701
1702  4422  D3  0C       10.0  TST01A: OUT   RINST              ;SINGLE STEP THE CLOCK
1703  4424  D3  0C       10.0          OUT   RINST              ;SINGLE STEP THE CLOCK
1704  4426  D3  0C       10.0          OUT   RINST              ;SINGLE STEP THE CLOCK
1705  4428  D3  0C       10.0          OUT   RINST              ;SINGLE STEP THE CLOCK
1706  442A  D3  0C       10.0          OUT   RINST              ;SINGLE STEP THE CLOCK
1707  442C  D3  0C       10.0          OUT   RINST              ;SINGLE STEP THE CLOCK
1708  442E  D3  0C       10.0          OUT   RINST              ;SINGLE STEP THE CLOCK
1709  4430  D3  0C       10.0          OUT   RINST              ;SINGLE STEP THE CLOCK
1710  4432  D3  0C       10.0          OUT   RINST              ;SINGLE STEP THE CLOCK
1711  4434  11  01  00   10.0          LXI   D,1                ;SET WATCHDOG TIMER INCREMENT
1712  4437  21  A8  FD   10.0          LXI   H,-600             ;SET WATCHDOG COUNT TO 600
1713  443A  D3  0C       10.0  3$:     OUT   RINST              ;SINGLE STEP THE READ PATH
1714  443C  DB  01       10.0          IN    RPCHI              ;DATA READY SET?
1715  443E  E6  10        7.0          ANI   R.DRDY
1716  4440  C2  55  44   10.0          JNZ   TST01K             ;YES-GO PROCESS
1717  4443  19           10.0          DAD   D                  ;NO-WATCHDOG TIMEOUT?
1718  4444  D2  3A  44   10.0          JNC   3$                 ;NO-CONTINUE
```

```
1720  4447                                        ERR     TST01C,DUMMW,1
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4447  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0003                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  444A  03                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  444B  01                                           .BYTE   1
 (1)  444C  CD  15  28      18.0                DUMMW:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  444F  DA  79  43      10.0                        JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1721                                    ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
1722                                    ;>SENT TO THE ECC
1723                                    ;<FATAL ERROR - MICRO TEST ABORTED
1724  4452  C3  36  47      10.0                JMP     EXIT
1725
1726  4455  16  01            7.0   TST01K: MVI     D,@1            ;INIT THE LOOP COUNT
1727  4457  AF                 4.0   TST01Y: XRA     A
1728  4458                               ROUT    RO5H
 (1)  4458  D3  8B           10.0                OUT     RO5H            ;WRITE AC INTO RO5H
 (1)  445A  7F                4.0                MOV     A,A             ;RETRY LINK
1729  445B  3A  B0  48        13.0                LDA     INDAT           ;GET THE INPUT DATA
1730  445E                               ROUT    EDATA
 (1)  445E  D3  95           10.0                OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4460  7F                4.0                MOV     A,A             ;RETRY LINK
1731  4461  A7                 4.0                ANA     A               ;SET CONDITION BITS
1732  4462  3E  00            7.0                MVI     A,0             ;CLEAR THE ACC.
1733  4464  E2  68  44        10.0                JPO     1$              ;ODD # OF ONES - CONTINUE
1734  4467  3C                 4.0                INR     A               ;EVEN # OF ONES - SET PARITY
1735  4468  32  B1  48        13.0   1$:         STA     INDATP          ;STORE THE PARITY BIT
1736  446B  D3  0C            10.0   TST01L: OUT     RINST           ;SINGLE STEP THE READ LOGIC
1737  446D  3A  B0  48        13.0                LDA     INDAT
1738  4470  47                 4.0                MOV     B,A
1739  4471  DB  19            10.0                IN      ECCCOR
1740  4473  B8                 4.0                CMP     B
1741  4474  CA  83  44        10.0                JZ      TST01M
1742  4477                               ROUT    ADATA
 (1)  4477  D3  94           10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4479  7F                4.0                MOV     A,A             ;RETRY LINK
1743  447A  78                 4.0                MOV     A,B             ;GET THE BYTE COUNT
1744  447B                               ROUT    RO5L                    ;STORE IN CAS
 (1)  447B  D3  8A           10.0                OUT     RO5L            ;WRITE AC INTO RO5L
 (1)  447D  7F                4.0                MOV     A,A             ;RETRY LINK
1745  447E                               ERRB    TST01C,TST01M,@1
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  447E  CD  12  28      18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0004                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4481  04                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4482  01                                           .BYTE   @1              ;PRINT ROUTINE NUMBER
 (1)  4483  CD  15  28      18.0                TST01M::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4486  DA  79  43      10.0                        JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1746                                    ;>CORRECTED DATA FROM THE ECC-INCORRECT
1747  4489  DB  15            10.0                IN      RPSTA           ;GET THE ECC PARITY BIT
1748  448B  E6  80             7.0                ANI     E.CDP           ;GET THE CORRECTED PARITY BIT
1749  448D  07                 4.0                RLC                     ;POSITION BIT FOR COMPARE
1750  448E  4F                 4.0                MOV     C,A             ;SAVE IN REG. C
1751  448F  3A  B1  48        13.0                LDA     INDATP          ;GET THE EXPECTED PARITY BIT
```

```
1752  4492  A7                    4.0          ANA     A                ;SET THE CONDITION BITS
1753  4493  CA  AE  44           10.0          JZ      TST01QO          ;GO CHECK FOR PARITY=0
1754  4496  79                    4.0          MOV     A,C              ;ELSE CHECK FOR PARITY=1
1755  4497  FE  01                7.0          CPI     @1
1756  4499  CA  A5  44           10.0          JZ      TST01RG
1757  449C  78                    4.0          MOV     A,B
1758  449D                                      ROUT    R05L
 (1)  449D  D3  8A                10.0                  OUT     R05L             ;WRITE AC INTO R05L
 (1)  449F  7F                    4.0                  MOV     A,A              ;RETRY LINK
1759  44A0                                      ERR     TST01C,TST01RG,@1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44A0  CD  09  28            18.0                  CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0005                                        MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44A3  05                                          .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  44A4  01                                          .BYTE   @1
 (1)  44A5  CD  15  28            18.0          TST01RG::       CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44A8  DA  79  43            10.0                  JC      TST01C           ;LOOP ADDRESS IF LOOP SPECIFIED
1760                                      ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
1761                                      ;<
1762  44AB  C3  C2  44           10.0          JMP     TST01SZ          ;CONTINUE WITH TEST
1763  44AE  79                    4.0  TST01QO: MOV     A,C              ;CHECK FOR PARITY = ZERO
1764  44AF  A7                    4.0          ANA     A                ;SET CONDITION BITS
1765  44B0  CA  BC  44           10.0          JZ      TST01TP          ;CONTINUE IF ZERO
1766  44B3  78                    4.0          MOV     A,B
1767  44B4                                      ROUT    R05L
 (1)  44B4  D3  8A                10.0                  OUT     R05L             ;WRITE AC INTO R05L
 (1)  44B6  7F                    4.0                  MOV     A,A              ;RETRY LINK
1768  44B7                                      ERR     TST01C,TST01TP,@1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44B7  CD  09  28            18.0                  CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0006                                        MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44BA  06                                          .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  44BB  01                                          .BYTE   @1
 (1)  44BC  CD  15  28            18.0          TST01TP::       CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44BF  DA  79  43            10.0                  JC      TST01C           ;LOOP ADDRESS IF LOOP SPECIFIED
1769                                      ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
1770                                      ;<
1771
1772  44C2  7A                    4.0  TST01SZ: MOV     A,D              ;GET THE CHARACTER COUNT
1773  44C3  3C                    4.0          INR     A                ;INCREMENT
1774  44C4  FE  08                7.0          CPI     $8               ;DONE?
1775  44C6  CA  CD  44           10.0          JZ      TST01N           ;YES-GO CHECK ECC
1776  44C9  57                    4.0          MOV     D,A              ;NO-CONTINUE
1777  44CA  C3  57  44           10.0          JMP     TST01Y
1778
1779  44CD  D3  0C                10.0  TST01N: OUT     RINST            ;SINGLE STEP THE READ PATH
1780  44CF  3A  76  48            13.0          LDA     ECCCHR           ;GET THE EXPECTED ECC CHARACTER
1781  44D2                                      ROUT    EDATA            ;SAVE IN CAS
 (1)  44D2  D3  95                10.0                  OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  44D4  7F                    4.0                  MOV     A,A              ;RETRY LINK
1782  44D5  47                    4.0          MOV     B,A              ;SAVE IN REGISTER B
1783  44D6  DB  19                10.0          IN      ECCCOR
1784  44D8                                      ROUT    ADATA
 (1)  44D8  D3  94                10.0                  OUT     ADATA            ;WRITE AC INTO ADATA
```

```
 (1)  44DA  7F                4.0                    MOV     A,A             ;RETRY LINK
1785  44DB  B8                4.0            CMP     B
1786  44DC  CA  EA  44         10.0           JZ      TST01U
1787  44DF  3A  B0  48         13.0           LDA     INDAT           ;GET THE DATA/COUNT
1788  44E2                                     ROUT    RO5L            ;WRITE TO THE CAS
 (1)  44E2  D3  8A             10.0                   OUT     RO5L            ;WRITE AC INTO RO5L
 (1)  44E4  7F                4.0                    MOV     A,A             ;RETRY LINK
1789  44E5                                     ERR     TST01C,TST01U,1
 (1)                                             ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44E5  CD  09  28         18.0                   CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44E8  07                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44E9  01                                       .BYTE   1
 (1)  44EA  CD  15  28         18.0           TST01U::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44ED  DA  79  43         10.0                   JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1790                                     ;>CORRECTED ECC CHARACTER INCORRECT
1791                                     ;<
1792  44F0  DB  1A               10.0           IN      ECCSTA          ;GET THE ECC STATUS
1793  44F2  E6  0F               7.0            ANI     @17             ;REMOVE UNWANTED BITS
1794  44F4  FE  09               7.0            CPI     E.STEC!E.PNTR
1795  44F6  CA  0C  45           10.0           JZ      TST01V
1796  44F9                                     ROUT    ADATA
 (1)  44F9  D3  94               10.0                   OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44FB  7F                   4.0                    MOV     A,A             ;RETRY LINK
1797  44FC  3E  09               7.0            MVI     A,E.STEC!E.PNTR
1798  44FE                                     ROUT    EDATA
 (1)  44FE  D3  95               10.0                   OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4500  7F                   4.0                    MOV     A,A             ;RETRY LINK
1799  4501  3A  B0  48           13.0           LDA     INDAT           ;GET THE DATA/COUNT
1800  4504                                     ROUT    RO5L            ;WRITE TO THE CAS
 (1)  4504  D3  8A               10.0                   OUT     RO5L            ;WRITE AC INTO RO5L
 (1)  4506  7F                   4.0                    MOV     A,A             ;RETRY LINK
1801  4507                                     ERRB    TST01C,TST01V,1
 (1)                                             ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4507  CD  12  28           18.0                   CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0008                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  450A  08                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  450B  01                                       .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  450C  CD  15  28           18.0           TST01V::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  450F  DA  79  43           10.0                   JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1802                                     ;>ECC STATUS INCORRECT AFTER A DATA GROUP
1803                                     ;<
1804  4512  3A  B0  48           13.0           LDA     INDAT
1805  4515  3C                   4.0            INR     A
1806  4516  32  B0  48           13.0           STA     INDAT
1807  4519  C2  79  43           10.0           JNZ     TST01C
1808  451C                                     ENDTST  TST01X
 (1)                                             ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  451C                                       REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  451C  CD  06  28           18.0                   CALL    REQST
 (2)  451F  00                                       .BYTE                   ;DATA PATTERN NUMBER
 (2)  4520  00  00                                   .WORD                   ;SYSTEM "" COUNT
 (2)  4522  00  00                                   .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4524  00                                       .BYTE                   ;DATA COMPARE FLAG IF -1
```

```
   (2)  4525  07                                        .BYTE   7               ;REQUEST CODE
   (1)  4526  3A   9A   4F      13.0          LDA      ITERA                   ;GET ITERATION COUNT
   (1)  4529  3D                 4.0          DCR      A                       ;DOWNCOUNT
   (1)  452A  32   9A   4F      13.0          STA      ITERA                   ;SAVE COUNT
   (1)  452D  F2   51   43      10.0          JP       TST01X                  ;DO TEST UNTIL TILL = 0
1809    4530  CD   39   47      18.0          CALL     CLEAR                   ;CLEAR ALL TU PORTS
```

```
1812                            .SBTTL  TEST 2 - ECC BAD DATA ONE TRACK GOOD POINTER
1813  4533                      ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1814                            ;*ECC BAD DATA ONE TRACK GOOD POINTER
1815  4533                      SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1816                            ;*THIS TEST CHECKS THE ABILITY OF THE ECC CONTROLLER TO PERFORM SINGLE
1817                            ;*TRACK ERROR CORRECTION WITH INCORRECT DATA IN A SINGLE TRACK, AND A
1818                            ;*CORRECT POINTER ALSO PROVIDED.  THE PROGRAM CHECKS THAT THE DATA IS
1819                            ;*CORRECTED, AND THAT THE SINGLE TRACK ERROR CORRECT STATUS BIT IS SET.
1820                            ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
1821                            ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
1822                            ;*MESSAGE, AND TRACK SIMULATED IN ERROR CAN BE DETERMINED FROM THE
1823                            ;*FOLLOWING TABLE.
1824                            ;*
1825                            ;*      INPUT/EXPECTED          DATA BIT POSITION       TRACK WITH ECC
1826                            ;*          DATA                WITH ECC CORRECTION       CORRECTION
1827                            ;*      -------------           -------------------     -------------
1828                            ;*
1829                            ;*          XXXX0000                    0                   2
1830                            ;*          XXXX0001                    1                   8
1831                            ;*          XXXX0010                    2                   1
1832                            ;*          XXXX0011                    3                   9
1833                            ;*          XXXX0100                    4                   3
1834                            ;*          XXXX0101                    5                   5
1835                            ;*          XXXX0110                    6                   6
1836                            ;*          XXXX0111                    7                   7
1837                            ;*          XXXX1000                    P                   4
1838                            ;*          XXXX1001                    1                   8
1839                            ;*          XXXX1010                    2                   1
1840                            ;*          XXXX1011                    3                   9
1841                            ;*          XXXX1100                    4                   3
1842                            ;*          XXXX1101                    5                   5
1843                            ;*          XXXX1110                    6                   6
1844                            ;*          XXXX1111                    7                   7
1845                            ;*
1846                            ;*      X = DON'T CARE BITS
1847                            ;*
1848                            ;*THE POINTERS PROVIDED FOR THE TRACK IN ERROR ARE ALSO DERIVED FROM THE
1849                            ;*INPUT/EXPECTED DATA AS PER THE FOLLOWING TABLE.
1850                            ;*
1851                            ;*      INPUT/EXPECTED
1852                            ;*          DATA                        POINTER GENERATED
1853                            ;*      -------------                   -----------------
1854                            ;*
1855                            ;*          XX00XXXX                    HISTORY
1856                            ;*          XX01XXXX                    PHASE TIE
1857                            ;*          XX10XXXX                    AMTIE
1858                            ;*          XX11XXXX                    ILLEGAL 5X4
1859                            ;*
```

```
1860                                ;*      X = DON'T CARE BITS
1861  4533                          SP
 (1)                                : ************************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
1862                                ;*BGNTST
1863                                ;*    SET NORMAL READ PATH CLOCK
1864                                ;*    CALL SUBROUTINE CLEAR
1865                                ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
1866                                ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
1867                                ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
1868                                ;*    CLEAR THE INPUT DATA
1869                                ;*    CLEAR BAD TRANSLATION POINTERS
1870                                ;*  BGNDO
1871                                ;*  : ENABLE THE READ PATH CLOCK
1872                                ;*  : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
1873                                ;*  : CLOCK THE FIFO'S
1874                                ;*  : ISSUE CLEAR ALL COMMAND
1875                                ;*  : WAIT
1876                                ;*  : STOP THE READ PATH, SET TIE BUS JAM AND SET PLO DISABLE
1877                                ;*  : CALL SUBROUTINE CLECC
1878                                ;*  : SET THE LOOP COUNT TO 7
1879                                ;*  : BGNDO
1880                                ;*  : :  GET THE INPUT DATA BYTE
1881                                ;*  : :  CALL SUBROUTINE ECC
1882                                ;*  : :  DECREMENT THE LOOP COUNT
1883                                ;*  : :  DO UNTIL THE LOOP COUNT=0
1884                                ;*  : ENDDO
1885                                ;*  : SET THE LOOP COUNT TO 3
1886                                ;*  : BGNDO
1887                                ;*  : :  FILL THE TRANSLATOR SUBROUTINE BUFFER WITH THE DATA BYTE
1888                                ;*  : :  CALL SUBROUTINE TRK
1889                                ;*  : :  CALL SUBROUTINE BADT4X5
1890                                ;*  : :  INIT THE LOOP COUNT TO 5
1891                                ;*  : :  BGNDO
1892                                ;*  : :  :  MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
1893                                ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1894                                ;*  : :  :  MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
1895                                ;*  : :  :  CLOCK THE DATA INTO THE FIFO
1896                                ;*  : :  :  DECREMENT THE LOOP COUNT
1897                                ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1898                                ;*  : :  :  DO UNTIL LOOP COUNT = 0
1899                                ;*  : :  ENDDO
1900                                ;*  : :  FILL THREE BYTES OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE DATA BYTE
1901                                ;*  : :  FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
1902                                ;*  : :  CALL SUBROUTINE TRK
1903                                ;*  : :  CALL SUBROUTINE BADT4X5
1904                                ;*  : :  INIT THE LOOP COUNT TO 5
1905                                ;*  : :  BGNDO
1906                                ;*  : :  :  MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
1907                                ;*  : :  :  MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
1908                                ;*  : :  :  CLOCK THE DATA INTO THE FIFO
1909                                ;*  : :  :  DECREMENT THE LOOP COUNT
1910                                ;*  : :  :  DO UNTIL THE LOOP COUNT = 0
```

```
1911                    ;*  : :  ENDDO
1912                    ;*  : :  DECREMENT THE LOOP COUNT
1913                    ;*  : :  DO UNTIL THE LOOP COUNT = 0
1914                    ;*  : ENDDO
1915                    ;*  : ISSUE DIAGNOSTIC READ COMMAND
1916                    ;*  : SET UP WATCHDOG TIMER COUNT
1917                    ;*  : BGNDO
1918                    ;*  : :    SINGLE STEP THE READ PATH
1919                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1920                    ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1921                    ;*  : ENDDO
1922                    ;*  : IF WATCHDOG TIMER=0
1923                    ;*  : :    THEN-ERROR
1924                    ;*  : :    E SE-CONTINUE
1925                    ;*  : ENDIF
1926                    ;*  : GET THE TIE BUS TABLE ADDRESS
1927                    ;*  : SINGLE STEP THE READ PATH
1928                    ;*  : INIT THE LOOP COUNT TO 0
1929                    ;*  : BGNDO
1930                    ;*  : :    IF LOOP COUNT=TIE BUS JAM COUNT
1931                    ;*  : :    :  THEN-CALL SUBROUTINE POINTER
1932                    ;*  : :    :  ELSE-CLEAR TIE BUS POINTER
1933                    ;*  : :    ENDIF
1934                    ;*  : :    LOAD THE TIE BUS REGISTER
1935                    ;*  : :    SINGLE STEP THE READ PATH
1936                    ;*  : :    DECREMENT THE LOOP COUNT
1937                    ;*  : :    DO UNTIL LOOP COUNT = 9
1938                    ;*  : ENDDO
1939                    ;*  : SET UP WATCHDOG TIMER COUNT
1940                    ;*  : BGNDO
1941                    ;*  : :  SINGLE STEP THE READ PATH
1942                    ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
1943                    ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
1944                    ;*  : ENDDO
1945                    ;*  : IF WATCHDOG TIMER=0
1946                    ;*  : :  THEN-ERROR-EXIT TEST
1947                    ;*  : :  ELSE-CONTINUE
1948                    ;*  : ENDIF
1949                    ;*  : SINGLE STEP THE READ PATH
1950                    ;*  : INIT THE  LOOP COUNT TO 7
1951                    ;*  : BGNDO
1952                    ;*  : :  SINGLE STEP THE READ PATH
1953                    ;*  : :  COMPARE CORRECTED DATA WITH INPUT DATA
1954                    ;*  : :  IF NOT EQUAL
1955                    ;*  : :  :  THEN-ECC ERROR
1956                    ;*  : :  :  ELSE-CONTINUE
1957                    ;*  : :  ENDIF
1958                    ;*  : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
1959                    ;*  : :  IF NOT EQUAL
1960                    ;*  : :  :  THEN-ECC ERROR
1961                    ;*  : :  :  ELSE-CONTINUE
1962                    ;*  : :  ENDIF
1963                    ;*  : :  DECREMENT THE LOOP COUNT
1964                    ;*  : :  DO UNTIL THE LOOP COUNT = 0
```

```
1965                                ;*  : ENDDO
1966                                ;*  : SINGLE STEP THE READ PATH
1967                                ;*  : IF EXPECTED NOT EQUAL
1968                                ;*  : :  THEN-ECC ERROR
1969                                ;*  : :  ELSE-CONTINUE
1970                                ;*  : ENDIF
1971                                ;*  : INPUT THE ECC STATUS
1972                                ;*  : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
1973                                ;*  : IF THE RESULTING STATUS=SINGLE TRACK ERROR CORRECT
1974                                ;*  : :  THEN-CONTINUE
1975                                ;*  : :  ELSE-ERROR
1976                                ;*  : ENDIF
1977                                ;*  : INCREMENT THE INPUT DATA
1978                                ;*  : DO UNTIL INPUT DATA = 0
1979                                ;*  ENDDO
1980                                ;*ENDTST
1981  4533                          SE
 (1)                                ; ***************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1982                                ;*ECC2 MICRO TEST 02
1983                                ;*ECC2 MICRO ERROR 11
1984                                ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK-GOOD POINTER
1985                                ;*M8951, M8950'S
1986                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
1987                                ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1988                                ;*FATAL ERROR - MICRO TEST ABORTED
1989                                ;*
1990                                ;*ECC2 MICRO TEST 02
1991                                ;*ECC2 MICRO ERROR 12
1992                                ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK-GOOD POINTER
1993                                ;*M8951, M8950'S
1994                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
1995                                ;*SENT TO THE ECC
1996                                ;*FATAL ERROR - MICRO TEST ABORTED
1997                                ;*
1998                                ;*ECC2 MICRO TEST 02
1999                                ;*ECC2 MICRO ERROR 13
2000                                ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK-GOOD POINTER
2001                                ;*M8951, M8950'S
2002                                ;*CORRECTED DATA FROM THE ECC-INCORRECT
2003                                ;*BYTE/SCLK COUNT NUMBER - LLL
2004                                ;*ACTUAL = NNNN
2005                                ;*EXPECTED = NNNN
2006                                ;*
2007                                ;*ECC2 MICRO TEST 02
2008                                ;*ECC2 MICRO ERROR 14
2009                                ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK-GOOD POINTER
2010                                ;*M8951, M8950'S
2011                                ;*CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
2012                                ;*BYTE/SCLK COUNT NUMBER = LLL
2013                                ;*
2014                                ;*ECC2 MICRO TEST 02
2015                                ;*ECC2 MICRO ERROR 15
```

```
2016                                    ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK-GOOD POINTER
2017                                    ;*M8951, M8950'S
2018                                    ;*CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
2019                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2020                                    ;*
2021                                    ;*ECC2 MICRO TEST 02
2022                                    ;*ECC2 MICRO ERROR 16
2023                                    ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK-GOOD POINTER
2024                                    ;*M8951, M8950'S
2025                                    ;*CORRECTED ECC CHARACTER INCORRECT
2026                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2027                                    ;*
2028                                    ;*ECC2 MICRO TEST 02
2029                                    ;*ECC2 MICRO ERROR 17
2030                                    ;*ECC2-ECC CONTROLLER-BAD DATA ONE TRACK-GOOD POINTER
2031                                    ;*M8951, M8950'S
2032                                    ;*ECC STATUS INCORRECT AFTER A DATA GROUP
2033                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2034                                    ;*ACTUAL = NNNN
2035                                    ;*EXPECTED = NNNN
2036  4533                              S
 (1)                                    ; ************************************************************************************
2037
2038  4533                              TEST2:  TESTX   @2              ;INITIALIZE THE TEST
 (1)  4533  3E  02          7.0                         MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4535  CD  03  28      18.0                        CALL    TSET            ;SETUP THE TEST
2039                                    ;%ECC2-ECC CONTROLLER-BAD DATA ONE TRACK-GOOD POINTER
2040                                    ;&M8951, M8950'S
2041
2042  4538                                      REQ     @7,0,0,0,0
 (1)  4538  CD  06  28      18.0                        CALL    REQST
 (1)  453B  00                                          .BYTE   0               ;DATA PATTERN NUMBER
 (1)  453C  00  00                                      .WORD   0               ;SYSTEM "0" COUNT
 (1)  453F  00  00                                      .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  4540  00                                          .BYTE   0               ;DATA COMPARE FLAG IF -1
 (1)  4541  07                                          .BYTE   @7              ;REQUEST CODE
2043
2044
2045  4542  3E  10           7.0  TST02X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
2046  4544  D3  F0          10.0         OUT     CLKCTL
2047  4546  3E  10           7.0         MVI     A,W.GCR
2048  4548  D3  D3          10.0         OUT     WMCCTL
2049
2050  454A  CD  39  47      18.0         CALL    CLEAR           ;CLEAR ALL TU PORTS
2051  454D  3A  B3  48      13.0         LDA     SUNIT
2052  4550  D3  E0          10.0         OUT     MBSEL
2053
2054  4552  3E  04           7.0         MVI     A,P.RPEN        ;ENABLE THE READ PATH
2055  4554  D3  4C          10.0         OUT     PENAB
2056
2057  4556  3E  60           7.0         MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
2058  4558  D3  48          10.0         OUT     PDIAG           ;LCS, LWR
2059  455A  AF                4.0         XRA     A
2060  455B  D3  44          10.0         OUT     TAMT
```

```
2061  455D  D3  D2        10.0          OUT     TRKENA            ;CLEAR ALL TRACKS FROM
2062  455F  D3  0A        10.0          OUT     RTIEB             ;CLEAR THE TIE BUS REGISTER
2063  4561  D3  D2        10.0          OUT     TRKENA
2064  4563  32  4B  48    13.0          STA     BAD1              ;CLEAR THE INVALID
2065  4566  32  4C  48    13.0          STA     BAD2              ;TRACK TRANSLATIONS
2066  4569  32  4D  48    13.0          STA     BAD3              ;POINTERS
2067  456C  32  B0  48    13.0          STA     INDAT             ;CLEAR THE INPUT DATA BYTE
2068
2069  456F  3E  A8         7.0  TST02C: MVI     A,R.PLO1.R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
2070                                                            ;SET TIE BUS JAM AND PLO DISABLE
2071  4571  D3  09        10.0          OUT     RPCTL
2072  4573  D3  08        10.0          OUT     RFIFOL            ;CLOCK THE FIFO'S
2073  4575  3E  0D         7.0          MVI     A,RCLRT           ;ISSUE CLEAR ALL COMMAND
2074  4577  D3  0B        10.0          OUT     RCMD
2075
2076  4579  00            4.0          NOP                       ;WAIT
2077  457A  00            4.0          NOP
2078  457B  00            4.0          NOP
2079  457C  00            4.0          NOP
2080  457D  00            4.0          NOP
2081
2082  457E  3E  A9         7.0          MVI     A,R.PLO1!R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
2083                                                            ;SET TIE BUS JAM AND PLO DISABLE
2084  4580  D3  09        10.0          OUT     RPCTL
2085  4582  CD  4E  48    18.0          CALL    CLECC             ;CLEAR THE ECC CHARACTER
2086  4585  16  07         7.0          MVI     D,@7              ;SET UP THE LOOP COUNT
2087  4587  3A  B0  48    13.0  TST02F: LDA     INDAT             ;GET THE INPUT DATA
2088  458A  CD  53  48    18.0          CALL    ECC               ;CALCULATE ECC FOR
2089  458D  15            4.0          DCR     D                 ;THE 7 INPUT BYTES
2090  458E  C2  87  45    10.0          JNZ     TST02F
2091
2092  4591  0E  03         7.0          MVI     C,@3              ;INIT THE LOOP COUNT
2093  4593  3A  B0  48    13.0  TST02I: LDA     INDAT             ;GET THE INPUT DATA
2094  4596  32  3B  48    13.0          STA     TRNIN             ;FILL THE TRANSLATOR
2095  4599  32  3C  48    13.0          STA     TRNIN+1           ;SUBROUTINE INPUT
2096  459C  32  3D  48    13.0          STA     TRNIN+2           ;BUFFER
2097  459F  32  3E  48    13.0          STA     TRNIN+3           ;
2098  45A2  CD  89  48    18.0          CALL    TRK
2099  45A5  CD  64  47    18.0          CALL    BADT4X5           ;TRANSLATE THE SUBGROUP
2100
2101  45A8  21  3F  48    10.0          LXI     H,TRNOUT          ;GET POINTER TO TRANSLATE DATA TABLE
2102  45AB  06  05         7.0          MVI     B,@5              ;SET UP LOOP COUNT
2103  45AD  7E             7.0  TST02G: MOV     A,M               ;GET A DATA BYTE
2104  45AE  D3  40        10.0          OUT     TCMD              ;STORE DATA IN COMMAND ADDRESS
2105  45B0  23            6.0          INX     H                 ;POINT TO DATA PARITY
2106  45B1  7E             7.0          MOV     A,M               ;GET THE DATA PARITY
2107
2108  45B2  07            4.0          RLC                       ;POSITION FOR OUTPUT
2109  45B3  F6  60         7.0          ORI     P.LWR.P.LCS       ;OR IN CONTROL BITS
2110  45B5  D3  48        10.0          OUT     PDIAG             ;OUTPUT THE DATA PARITY
2111
2112  45B7  D3  08        10.0          OUT     RFIFOL            ;CLOCK DATA INTO THE FIFO'S
2113
2114  45B9  23            6.0          INX     H                 ;UPDATE THE TABLE POINTER
```

```
2115  45BA  05              4.0          DCR     B               ;DECREMENT LOOP COUNT
2116  45BB  C2  AD  45      10.0         JNZ     TST02G          ;DO UNTIL LOOP COUNT = 0
2117
2118  45BE  3A  B0  48      13.0         LDA     INDAT           ;GET THE INPUT DATA
2119  45C1  32  3B  48      13.0         STA     TRNIN           ;FILL THE FIRST THREE
2120  45C4  32  3C  48      13.0         STA     TRNIN+1         ;BYTES OF THE TRANSLATOR
2121  45C7  32  3D  48      13.0         STA     TRNIN+2         ;SUBROUTINE WITH THE INPUT DATA
2122  45CA  3A  76  48      13.0         LDA     ECCCHR          ;STORE THE CALCULATED
2123  45CD  32  3E  48      13.0         STA     TRNIN+3         ;ECC CHARACTER AS THE LAST CHARACTER
2124  45D0  CD  89  48      18.0         CALL    TRK
2125
2126  45D3  CD  64  47      18.0         CALL    BADT4X5         ;TRANSLATE THE SECOND SUBGROUP
2127
2128  45D6  21  3F  48      10.0         LXI     H,TRNOUT        ;GET THE POINTER TO INPUT DATA TABLE
2129  45D9  06  05           7.0         MVI     B,@5            ;SET UP THE LOOP COUNT
2130  45DB  7E               7.0  TST02H: MOV     A,M             ;GET A DATA BYTE
2131  45DC  D3  40          10.0         OUT     TCMD            ;STORE DATA IN CMD ADDRESS
2132  45DE  23               6.0         INX     H               ;POINT TO DATA PARITY BIT
2133  45DF  7E               7.0         MOV     A,M             ;GET THE DATA PARITY
2134  45E0  07               4.0         RLC                     ;POSITION FOR OUTPUT
2135  45E1  F6  60           7.0         ORI     P.LWR!P.LCS     ;OR IN CONTROL BITS
2136  45E3  D3  48          10.0         OUT     PDIAG           ;OUTPUT THE DATA PARITY
2137  45E5  D3  08          10.0         OUT     RFIFOL          ;CLOCK DATA INTO THE FIFOS
2138  45E7  23               6.0         INX     H               ;UPDATE THE TABLE POINTER
2139  45E8  05               4.0         DCR     B               ;DECREMENT THE LOOP COUNT
2140  45E9  C2  DB  45      10.0         JNZ     TST02H          ;DO UNTIL LOOP COUNT=0
2141
2142  45EC  0D               4.0         DCR     C               ;PUT THE GROUP IN
2143  45ED  C2  93  45      10.0         JNZ     TST021          ;THE FIFO TWICE
2144
2145  45F0  3E  0B           7.0         MVI     A,DIARD         ;LOAD THE DIAGNOSTIC READ
2146  45F2  D3  0B          10.0         OUT     RCMD            ;COMMAND
2147  45F4  11  01  00      10.0         LXI     D,1             ;SET WATCH DOG INCREMENT
2148  45F7  21  A8  FD      10.0         LXI     H,-600          ;SET WATCH DOG COUNT TO 600
2149  45FA  D3  0C          10.0  2$:    OUT     RINST           ;STEP THE READ PATH
2150  45FC  DB  01          10.0         IN      RPCHI           ;DATA READY SET?
2151  45FE  E6  10           7.0         ANI     R.DRDY
2152  4600  C2  15  46      10.0         JNZ     TST02A          ;YES-GO PROCESS
2153  4603  19              10.0         DAD     D               ;WATCH DOG TIMEOUT?
2154  4604  D2  FA  45      10.0         JNC     2$              ;NO-CONTINUE
2155  4607                               ERR     TST02C,DUMMU
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4607  CD  09  28      18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0009                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  460A  09                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  460B  00                          .BYTE
 (1)  460C  CD  15  28      18.0  DUMMU:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  460F  DA  6F  45      10.0         JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2156                                ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
2157                                ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
2158                                ;<FATAL ERROR - MICRO TEST ABORTED
2159  4612  C3  36  47      10.0         JMP     EXIT
2160
2161  4615  D3  0C          10.0  TST02A: OUT     RINST
```

```
2162  4617  06  00          7.0          MVI   B,$0          ;INIT THE BIT TIME COUNT (LOOP COUNT)
2163  4619  00              4.0  TST02B: NOP
2164  461A  3A  AF  48      13.0          LDA   TIETIM        ;GET THE TIME FOR TIE BUS JAMING
2165  461D  B8              4.0          CMP   B             ;IS IT TIME TO JAM THE TIE BUS?
2166  461E  C2  27  46      10.0          JNZ   4$            ;NO - CONTINUE
2167  4621  CD  0F  48      18.0          CALL  POINTER       ;YES - GO GET A REAL POINTER
2168  4624  C3  2A  46      10.0          JMP   5$            ;
2169  4627  AF              4.0  4$:      XRA   A             ;IF NOT THE BAD TRACK, PROVIDE A GOOD POINTER
2170  4628  D3  0A          10.0          OUT   RTIEB         ;LOAD THE POINTER
2171  462A  D3  0C          10.0  5$:      OUT   RINST         ;STEP CLOCK
2172  462C  04              4.0          INR   B             ;DECREMENT THE LOOP COUNT
2173  462D  78              4.0          MOV   A,B
2174  462E  FE  09          7.0          CPI   $9            ;DONE
2175  4630  C2  19  46      10.0          JNZ   TST02B        ;DO UNTIL LOOP COUNT = ZERO
2176  4633  AF              4.0          XRA   A
2177  4634  D3  0A          10.0          OUT   RTIEB         ;CLEAR THE TIE BUS REGISTER
2178  4636  D3  0C          10.0          OUT   RINST         ;CLOCK THE READ PATH
2179  4638  D3  0C          10.0          OUT   RINST         ;CLOCK THE READ PATH
2180  463A  11  01  00      10.0          LXI   D,1           ;SET WATCHDOG TIMER INCREMENT
2181  463D  21  A8  FD      10.0          LXI   H,-600        ;SET WATCHDOG COUNT TO 600
2182  4640  D3  0C          10.0  3$:      OUT   RINST         ;SINGLE STEP THE READ PATH
2183  4642  DB  01          10.0          IN    RPCHI         ;DATA READY SET?
2184  4644  E6  10          7.0           ANI   R.DRDY
2185  4646  C2  5B  46      10.0          JNZ   TST02K        ;YES-GO PROCESS
2186  4649  19              10.0          DAD   D             ;NO-WATCHDOG TIMEOUT?
2187  464A  D2  40  46      10.0          JNC   3$            ;NO-CONTINUE
2188  464D                                ERR   TST02C,DUMMV
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  464D  CD  09  28      18.0                CALL  ERLP          ;PROCESS ERROR - DO 2.3
 (1)       000A                           MSGN  =     MSGN+1        ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4650  0A                           .BYTE MSGN          ;MESSAGE NUMBER ID
 (1)  4651  00                           .BYTE
 (1)  4652  CD  15  28      18.0  DUMMV:: CALL  CKLOP         ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4655  DA  6F  45      10.0          JC    TST02C        ;LOOP ADDRESS IF LOOP SPECIFIED
2189                        ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
2190                        ;>SENT TO THE ECC
2191                        ;<FATAL ERROR - MICRO TEST ABORTED
2192  4658  C3  36  47      10.0          JMP   EXIT
2193
2194  465B  16  01          7.0  TST02K: MVI   D,@1          ;INIT THE LOOP COUNT
2195  465D  AF              4.0  TST02Y: XRA   A
2196  465E                            ROUT  R05H
 (1)  465E  D3  8B          10.0          OUT   R05H          ;WRITE AC INTO R05H
 (1)  4660  7F              4.0          MOV   A,A           ;RETRY LINK
2197  4661  3A  B0  48      13.0          LDA   INDAT         ;GET THE INPUT DATA
2198  4664                            ROUT  EDATA
 (1)  4664  D3  95          10.0          OUT   EDATA         ;WRITE AC INTO EDATA
 (1)  4666  7F              4.0          MOV   A,A           ;RETRY LINK
2199  4667  A7              4.0          ANA   A             ;SET CONDITION BITS
2200  4668  3E  00          7.0          MVI   A,0           ;CLEAR THE ACC.
2201  466A  E2  6E  46      10.0          JPO   1$            ;ODD # OF ONES - CONTINUE
2202  466D  3C              4.0          INR   A             ;EVEN # OF ONES - SET PARITY
2203  466E  32  B1  48      13.0  1$:     STA   INDATP        ;STORE THE PARITY BIT
2204  4671  D3  0C          10.0  TST02L: OUT   RINST         ;SINGLE STEP THE READ LOGIC
```

```
2205  4673  3A B0 48     13.0          LDA     INDAT
2206  4676  47            4.0          MOV     B,A
2207  4677  DB 19        10.0          IN      ECCCOR
2208  4679  B8            4.0          CMP     B
2209  467A  CA 89 46     10.0          JZ      TST02M
2210  467D                              ROUT    ADATA
 (1)  467D  D3 94        10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  467F  7F            4.0                  MOV     A,A             ;RETRY LINK
2211  4680  78            4.0          MOV     A,B             ;GET THE BYTE COUNT
2212  4681                              ROUT    R05L            ;STORE IN CAS
 (1)  4681  D3 8A        10.0                  OUT     R05L            ;WRITE AC INTO R05L
 (1)  4683  7F            4.0                  MOV     A,A             ;RETRY LINK
2213  4684                              ERRB    TST02C,TST02M,@1
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4684  CD 12 28     18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000B                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4687  0B                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4688  01                                 .BYTE   @1              ;PRINT ROUTINE NUMBER
 (1)  4689  CD 15 28     18.0          TST02M::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  468C  DA 6F 45     10.0                  JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2214                              ;>CORRECTED DATA FROM THE ECC-INCORRECT
2215  468F  DB 15        10.0          IN      RPSTA           ;GET THE ECC PARITY BIT
2216  4691  E6 80         7.0          ANI     E.CDP           ;GET THE CORRECTED PARITY BIT
2217  4693  07            4.0          RLC                     ;POSITION BIT FOR COMPARE
2218  4694  4F            4.0          MOV     C,A             ;SAVE IN REG. C
2219  4695  3A B1 48     13.0          LDA     INDATP          ;GET THE EXPECTED PARITY BIT
2220  4698  A7            4.0          ANA     A               ;SET THE CONDITION BITS
2221  4699  CA B4 46     10.0          JZ      TST02Q0         ;GO CHECK FOR PARITY=0
2222  469C  79            4.0          MOV     A,C             ;ELSE CHECK FOR PARITY=1
2223  469D  FE 01         7.0          CPI     @1
2224  469F  CA AB 46     10.0          JZ      TST02RG
2225  46A2  78            4.0          MOV     A,B
2226  46A3                              ROUT    R05L
 (1)  46A3  D3 8A        10.0                  OUT     R05L            ;WRITE AC INTO R05L
 (1)  46A5  7F            4.0                  MOV     A,A             ;RETRY LINK
2227  46A6                              ERR     TST02C,TST02RG,@1
 (1)                                  ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  46A6  CD 09 28     18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000C                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46A9  0C                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46AA  01                                 .BYTE   @1
 (1)  46AB  CD 15 28     18.0          TST02RG::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  46AE  DA 6F 45     10.0                  JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2228                              ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
2229                              ;<
2230  46B1  C3 C8 46     10.0          JMP     TST02SZ         ;CONTINUE WITH TEST
2231  46B4  79            4.0  TST02Q0: MOV    A,C             ;CHECK FOR PARITY = ZERO
2232  46B5  A7            4.0          ANA     A               ;SET CONDITION BITS
2233  46B6  CA C2 46     10.0          JZ      TST02TP         ;CONTINUE IF ZERO
2234  46B9  78            4.0          MOV     A,B
2235  46BA                              ROUT    R05L
 (1)  46BA  D3 8A        10.0                  OUT     R05L            ;WRITE AC INTO R05L
 (1)  46BC  7F            4.0                  MOV     A,A             ;RETRY LINK
```

```
2237  46BD                                    ERR     TST02C,TST02TP,@1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  46BD  CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000D                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46C0  0D                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46C1  01                                        .BYTE   @1
 (1)  46C2  CD  15  28        18.0            TST02TP::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  46C5  DA  6F  45        10.0                    JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2238                                    ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
2239                                    ;<
2240  46C8  7A                   4.0    TST02SZ: MOV    A,D             ;GET THE CHARACTER COUNT
2241  46C9  3C                   4.0             INR     A               ;INCREMENT
2242  46CA  FE  08               7.0             CPI     $8              ;DONE?
2243  46CC  CA  D3  46          10.0             JZ      TST02N          ;YES-GO CHECK ECC
2244  46CF  57                   4.0             MOV     D,A             ;NO-CONTINUE
2245  46D0  C3  5D  46          10.0             JMP     TST02Y
2246
2247  46D3  D3  0C               10.0    TST02N: OUT     RINST           ;SINGLE STEP THE READ PATH
2248  46D5  3A  76  48          13.0             LDA     ECCCHR          ;GET THE EXPECTED ECC CHARACTER
2249  46D8                                     ROUT    EDATA           ;SAVE IN CAS
 (1)  46D8  D3  95               10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  46DA  7F                    4.0                    MOV     A,A             ;RETRY LINK
2250  46DB  47                   4.0             MOV     B,A             ;SAVE IN REGISTER B
2251  46DC  DB  19               10.0             IN      ECCCOR
2252  46DE                                     ROUT    ADATA
 (1)  46DE  D3  94               10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  46E0  7F                    4.0                    MOV     A,A             ;RETRY LINK
2253  46E1  B8                   4.0             CMP     B
2254  46E2  CA  F0  46          10.0             JZ      TST02U
2255  46E5  3A  B0  48          13.0             LDA     INDAT           ;GET THE DATA/COUNT
2256  46E8                                     ROUT    R05L            ;WRITE TO THE CAS
 (1)  46E8  D3  8A               10.0                    OUT     R05L            ;WRITE AC INTO R05L
 (1)  46EA  7F                    4.0                    MOV     A,A             ;RETRY LINK
2257  46EB                                     ERR     TST02C,TST02U,1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  46EB  CD  09  28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000E                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46EE  0E                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46EF  01                                        .BYTE   1
 (1)  46F0  CD  15  28        18.0            TST02U::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  46F3  DA  6F  45        10.0                    JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
2258                                    ;>CORRECTED ECC CHARACTER INCORRECT
2259                                    ;<
2260  46F6  DB  1A               10.0             IN      ECCSTA          ;GET THE ECC STATUS
2261  46F8  E6  0F                7.0             ANI     @17             ;REMOVE UNWANTED BITS
2262  46FA  FE  01                7.0             CPI     E.STEC
2263  46FC  CA  12  47          10.0             JZ      TST02V
2264  46FF                                     ROUT    ADATA
 (1)  46FF  D3  94               10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4701  7F                    4.0                    MOV     A,A             ;RETRY LINK
2265  4702  3E  01                7.0             MVI     A,E.STEC
2266  4704                                     ROUT    EDATA
 (1)  4704  D3  95               10.0                    OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4706  7F                    4.0                    MOV     A,A             ;RETRY LINK
```

```
 2267  4707  3A  B0  48      13.0          LDA     INDAT           ;GET THE DATA/COUNT
 2268  470A                                ROUT    RO5L            ;WRITE TO THE CAS
  (1)  470A  D3  8A          10.0                  OUT     RO5L            ;WRITE AC INTO RO5L
  (1)  470C  7F               4.0                  MOV     A,A             ;RETRY LINK
 2269  470D                                ERRB    TST02C,TST02V,1
  (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  470D  CD  12  28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       000F                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4710  0F                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4711  01                                     .BYTE   1               ;PRINT ROUTINE NUMBER
  (1)  4712  CD  15  28      18.0          TST02V::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  4715  DA  6F  45      10.0                  JC      TST02C          ;LOOP ADDRESS IF LOOP SPECIFIED
 2270                                ;>ECC STATUS INCORRECT AFTER A DATA GROUP
 2271                                ;<
 2272  4718  3A  B0  48      13.0          LDA     INDAT
 2273  471B  3C               4.0          INR     A
 2274  471C  32  B0  48      13.0          STA     INDAT
 2275  471F  C2  6F  45      10.0          JNZ     TST02C
 2276  4722                                ENDTST  TST02X
  (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  4722                                REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  4722  CD  06  28      18.0                  CALL    REQST
  (2)  4725  00                                     .BYTE                   ;DATA PATTERN NUMBER
  (2)  4726  00  00                                 .WORD                   ;SYSTEM "" COUNT
  (2)  4728  00  00                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  472A  00                                     .BYTE                   ;DATA COMPARE FLAG IF -1
  (2)  472B  07                                     .BYTE   7               ;REQUEST CODE
  (1)  472C  3A  9A  4F      13.0          LDA     ITERA           ;GET ITERATION COUNT
  (1)  472F  3D               4.0          DCR     A               ;DOWNCOUNT
  (1)  4730  32  9A  4F      13.0          STA     ITERA           ;SAVE COUNT
  (1)  4733  F2  42  45      10.0          JP      TST02X          ;DO TEST UNTIL TILL = 0
 2277  4736  C3  18  28      10.0  EXIT:   JMP     TSTEND
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ; ************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ; ************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ; ************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*    CLEAR PORT SELECT FOR TRANSPORT
                                        ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*    CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ; ************************************************************************
4739  F5                    12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
473A  C5                    12.0          PUSH    B
473B  06  00                 7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
473D  DB  E0                10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
473F  E6  80                 7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
4741  B0                     4.0          ORA     B               ;ADD IN THE SELECTED PORT #
4742  D3  E0                10.0          OUT     MBSEL           ;RESET TO THIS PORT
4744  3E  80                 7.0          MVI     A,@200          ;LOAD MTA RFGISTER #0 SELECT CODE
4746  D3  40                10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
4748  AF                     4.0          XRA     A               ;CLEAR TU COMMAND A
4749  D3  40                10.0          OUT     TCMD            ;
474B  3E  81                 7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
474D  D3  40                10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
474F  3E  00                 7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
4751  D3  40                10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
4753  AF                     4.0          XRA     A
4754  D3  44                10.0          OUT     TAMT            ;CLEAR AMTIE WORD
4756  D3  48                10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
4758  D3  4C                10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
475A  04                     4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
475B  78                     4.0          MOV     A,B
475C  FE  04                 7.0          CPI     4               ;DONE?
475E  C2  3D  47            10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
4761  C1                    10.0          POP     B               ;RESET B & C
4762  F1                    10.0          POP     PSW             ;ALL DONE
4763  C9                    10.0          RET                     ;EXIT
```

```
                                        .SBTTL  SUBROUTINE 4X5 TRANSLATE A SUBGROUP
4764                                    SSUB
 (1)                                    ; ************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
                                        ;*4X5 TRANSLATE \ SUBGROUP
4764                                    SD
 (1)                                    ; ************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO 4X5 TRANSLATE A 4 BYTE DATA SUBGROUP INTO
                                        ;*A 5 BYTE SUBGROUP AS RECORDED ON TAPE OR READ FROM TAPE, AT THE ENTRY
                                        ;*T4X5.  AT ENTRY BADT4X5 THE VARIABLES BAD 1, 2, 3 CAN CONTAIN NUMBERS
                                        ;*FROM 1-9 TO INDICATE TRACKS THAT WILL BE TRANSLATED TO A VALID VALUE,
                                        ;*BUT NOT CORRECT FOR THE INPUT DATA.
4764                                    SP
 (1)                                    ; ************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNSUB
                                        ;*    CLEAR THE BAD TRANSLATION POINTERS
                                        ;*    GET THE 4 (INPUT) DATA BYTES TO BE TRANSLATED
                                        ;*    CLEAR THE TRANSLATED (OUTPUT) BUFFER (OUTPUT BUFFER IS 9X5)
                                        ;*    SET LOOP COUNT TO 1
                                        ;*    BGNDO
                                        ;*    :    CALCULATE ODD PARITY FOR THE FOUR (INPUT) DATA BYTES (INPUT BUFFER IS NOW 9X4
                                        ;*    :    ROTATE EACH (5) ENTRY OF THE TRANSLATED (OUTPUT) BUFFER RIGHT
                                        ;*    :    GET THE LEAST SIGNIFICANT BIT OF EACH (INPUT) DATA BYTE AND
                                        ;*    :        FORM A TABLE INDEX (MS BIT OF INDEX = LS BIT OF THE FIRST
                                        ;*    :        BYTE OF THE INPUT DATA, ETC.)
                                        ;*    :    REMOVE THE LS BIT OF EACH INPUT DATA BYTE (THE ONE JUST USED
                                        ;*    :        TO BUILD THE INDEX)
                                        ;*    :    USE THE 4 BIT INDEX TO GET THE TRANSLATION VALUE
                                        ;*    :    PUT ONE BIT OF THE TRANSLATION VALUE, AS THE MOST SIGNIFICANT
                                        ;*    :        BIT EACH OF THE 5 TRANSLATED DATA BYTES (LS BIT OF THE
                                        ;*    :        TRANSLATION VALUE REPLACES MS BIT OF THE FIRST TRANSLATED
                                        ;*    :        DATA BYTE.
                                        ;*    :    INCREMENT THE LOOP COUNT
                                        ;*    :    DO UNTIL THE LOOP COUNT = 10
                                        ;*    ENDDO
                                        ;*ENDSUB
4764                                    S
 (1)                                    ; ************************************************************************

4764  F5              12.0  BADT4X5: PUSH   PSW             ;SAVE REGISTER A + PSW
4765  C3  73  47       10.0         JMP    BAD
4768  F5              12.0  T4X5:    PUSH   PSW             ;SAVE REGISTER A + PSW
4769  AF               4.0         XRA    A               ;CLEAR A
476A  32  4B  48       13.0         STA    BAD1            ;CLEAR BAD TRANSLATION 1
476D  32  4C  48       13.0         STA    BAD2            ;CLEAR BAD TRANSLATION 2
4770  32  4D  48       13.0         STA    BAD3            ;CLEAR BAD TRANSLATION 3
4773  C5              12.0  BAD:     PUSH   B               ;SAVE REGISTER B + C
4774  D5              12.0         PUSH   D               ;SAVE REGISTER D + E
4775  E5              12.0         PUSH   H               ;SAVE REGISTER H + L
```

```
2367                                        ;FOUR DATA BYTES ARE NOW IS THE BUFFER TABLE
2368                                        ;CLEAR THE TRANSLATED DATA TABLE.
2369  4776  06  0A          7.0           MVI   B,10            ;SET UP LOOP COUNT
2370  4778  11  3F  48      10.0           LXI   D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
2371  477B  AF                4.0           XRA   A             ;CLEAR A
2372  477C  12                7.0  D4X5:    STAX  D             ;STORE IN THE TABLE
2373  477D  13                6.0           INX   D             ;UPDATE TABLE POINTER
2374  477E  05                4.0           DCR   B             ;DECREMENT LOOP COUNT
2375  477F  C2  7C  47        10.0           JNZ   D4X5          ;DO UNTIL LOOP COUNT=0
2376  4782  0E  01            7.0           MVI   C,1           ;SET UP TRACK COUNT
2377  4784  06  04            7.0  B4X5:    MVI   B,4           ;SET UP BIT COUNT
2378  4786  11  3B  48      10.0           LXI   D,TRNIN
2379  4789  AF                4.0           XRA   A             ;CLEAR THE GROUP POSITION COUNT
2380  478A  32  49  48      13.0           STA   GP4X5
2381  478D  1A                7.0  C4X5:    LDAX  D             ;GET A DATA BYTE
2382  478E  A7                4.0           ANA   A             ;SET CONDITION BITS
2383  478F  E2  93  47        10.0           JPO   PO4X5         ;ODD PARITY LEAVE CARRY CLEAR
2384  4792  37                4.0           STC                 ;EVEN PARITY SET CARRY
2385  4793  1F                4.0  PO4X5:   RAR                 ;SHIFT OUT DESIRED BIT
2386  4794  12                7.0           STAX  D             ;STORE RESULT BACK IN TEMP TABLE
2387  4795  3A  49  48      13.0           LDA   GP4X5         ;GET THE GROUP POSITION BYTE
2388  4798  17                4.0           RAL                 ;PUT IN THIS DATA BIT
2389  4799  32  49  48      13.0           STA   GP4X5         ;SAVE THE UPDATED GROUP POSITION BYTE
2390  479C  13                6.0           INX   D             :UPDATE THE TABLE POINTER
2391  479D  05                4.0           DCR   B             ;DECREMENT THE BIT COUNT
2392  479E  C2  8D  47        10.0           JNZ   C4X5          ;DO UNTIL ALL 4 BITS ARE RECEIVED
2393
2394                                        ;AT THIS POINT GP4X5 CONTAINS 4 BITS - ONE FROM EACH OF THE FOUR BYTES
2395
2396                                        ;PERFORM A DOUBLE PRECISION SHIFT RIGHT ON THE TRANSLATED DATA
2397  47A1  06  05              7.0           MVI   B,5           ;SET UP LOOP COUNT
2398  47A3  11  3F  48      10.0           LXI   D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
2399  47A6  13                6.0  E4X5:    INX   D             ;POINT TO PARITY BIT
2400  47A7  1A                7.0           LDAX  D             ;GET PARITY BIT
2401  47A8  A7                4.0           ANA   A             ;CLEAR THE CARRY BIT
2402  47A9  1F                4.0           RAR                 ;SHIFT TO THE CARRY BIT
2403  47AA  12                7.0           STAX  D             ;STORE IT BACK
2404  47AB  1B                6.0           DCX   D             ;DECREMENT TO DATA BITS
2405  47AC  1A                7.0           LDAX  D             ;GET THE DATA BITS
2406  47AD  1F                4.0           RAR                 ;SHIFT IN CARRY BITS
2407  47AE  12                7.0           STAX  D             ;STORE BACK IN TABLE
2408  47AF  13                6.0           INX   D             ;POINT TO NEXT TABLE ENTRY
2409  47B0  13                6.0           INX   D             ;
2410  47B1  05                4.0           DCR   B             ;DECREMENT LOOP COUNT
2411  47B2  C2  A6  47        10.0           JNZ   E4X5          ;DO UNTIL LOOP COUNT=0
2412  47B5  3A  4B  48      13.0           LDA   BAD1          ;IS A BAD POINTER SPECIFIED?
2413  47B8  A7                4.0           ANA   A             ;SET CONDITION BITS
2414  47B9  CA  C0  47        10.0           JZ    BAD1C         ;NO-CONTINUE
2415  47BC  B9                4.0           CMP   C             ;YES-IS IT THIS DATA POSITION?
2416  47BD  CA  D6  47        10.0           JZ    INV4X5        ;YES-GO PERFORM INCORRECT TRANSLATION
2417  47C0  3A  4C  48      13.0  BAD1C:   LDA   BAD2          ;IS A BAD POINTER SPECIFIED
2418  47C3  A7                4.0           ANA   A             ;SET CONDITION BITS
2419  47C4  CA  CB  47        10.0           JZ    BAD2C         ;NO-CONTINUE
2420  47C7  B9                4.0           CMP   C             ;YES-IS IT THIS DATA POSITION
```

```
2421  47C8  CA  D6  47   10.0          JZ    INV4X5          ;YES-GO PERFORM INCORRECT TRANSLATION
2422  47CB  3A  4D  48   13.0  BAD2C:  LDA   BAD3            ;IS A BAD POINTER SPECIFIED
2423  47CE  A7             4.0          ANA   A               ;SET CONDITION BITS
2424  47CF  CA  DD  47   10.0          JZ    BAD3C           ;NO-PERFORM PROPER TRANSLATION
2425  47D2  B9             4.0          CMP   C               ;YES-IS IT THIS DATA POSITION
2426  47D3  C2  DD  47   10.0          JNZ   BAD3C           ;NO-GO PERFORM PROPER TRANSLATION
2427  47D6  3A  49  48   13.0  INV4X5: LDA   GP4X5           ;GET THE GROUP POSITION COUNT
2428  47D9  3C             4.0          INR   A               ;INCREMENT-TO INCORRECT TRANSLATION
2429  47DA  32  49  48   13.0          STA   GP4X5           ;STORE IT
2430  47DD  11  2B  48   10.0  BAD3C:  LXI   D,TAB4X5        ;LOAD ADDRESS OF TRANSLATION TABLE
2431  47E0  26  00         7.0          MVI   H,0             ;CLEAR REGISTER H
2432  47E2  3A  49  48   13.0          LDA   GP4X5           ;GET GROUP POSITION COUNT
2433  47E5  E6  0F         7.0          ANI   $0F
2434  47E7  6F             4.0          MOV   L,A
2435  47E8  19            10.0          DAD   D               ;ADD GROUP POSITION COUNT TO TABLE
2436  47E9  7E             7.0          MOV   A,M             ;GET TRANSLATED DATA
2437  47EA  32  4A  48   13.0          STA   TRNTMP
2438                                ;''TRNTMP'' = THE TRANSLATED DATA
2439
2440  47ED  11  3F  48   10.0          LXI   D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
2441  47F0  06  05         7.0          MVI   B,5             ;SET UP LOOP COUNT
2442  47F2  3A  4A  48   13.0  F4X5:   LDA   TRNTMP          ;GET TRANSLATED DATA
2443  47F5  13             6.0          INX   D               ;POINT TO PARITY POSITION
2444  47F6  A7             4.0          ANA   A               ;CLEAR CARRY BIT
2445  47F7  1F             4.0          RAR                   ;SHIFT OUT A BIT
2446  47F8  32  4A  48   13.0          STA   TRNTMP          ;STORE TRANSLATED DATA
2447  47FB  1A             7.0          LDAX  D               ;GET THE PARITY BIT (ALWAYS ZERO)
2448  47FC  17             4.0          RAL                   ;ROLL IN THE CARRY BIT
2449  47FD  12             7.0          STAX  D               ;STORE AS THE PARITY BIT
2450  47FE  13             6.0          INX   D               ;UPDATE TABLE POINTER
2451  47FF  05             4.0          DCR   B               ;DECREMENT THE LOOP COUNT
2452  4800  C2  F2  47   10.0          JNZ   F4X5            ;DO UNTIL LOOP COUNT=0
2453
2454                          ;NOW THE TRANSLATED DATA TABLE HAS BEEN UPDATED
2455
2456  4803  0C             4.0          INR   C               ;DECREMENT THE TRACK COUNT
2457  4804  79             4.0          MOV   A,C
2458  4805  FE  0A         7.0          CPI   10
2459  4807  C2  84  47   10.0          JNZ   B4X5            ;DO UNTIL ALL TRACKS TRANSLATED
2460  480A  E1            10.0          POP   H               ;RESTORE REGISTER H + L
2461  480B  D1            10.0          POP   D               ;RESTORE REGISTER D + E
2462  480C  C1            10.0          POP   B               ;RESTORE REGISTER B + C
2463  480D  F1            10.0          POP   PSW             ;RESTORE REGISTER A + PSW
2464  480E  C9            10.0          RET                   ;RETURN TO USER
2465
```

```
                                        .SBTTL  SUBROUTINE CALCULATE POINTER
                                        ;*THE POINTERS PROVIDED FOR THE TRACK IN ERROR ARE ALSO DERIVED FROM THE
                                        ;*INPUT/EXPECTED DATA AS PER THE FOLLOWING TABLE.
                                        ;*
                                        ;*      INPUT/EXPECTED
                                        ;*          DATA                        POINTER GENERATED
                                        ;*      -------------                   -----------------
                                        ;*
                                        ;*         XX00XXXX                     HISTORY
                                        ;*         XX01XXXX                     PHASE TIE
                                        ;*         XX10XXXX                     AMTIE
                                        ;*         XX11XXXX                     ILLEGAI 5X4
                                        ;*
                                        ;*      X = DON'T CARE BITS
480F                                    SP
 (1)                                    ;*********************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------

480F  F5                      12.0  POINTER: PUSH   PSW             ;SAVE THE ACCUMULATOR
4810  C5                      12.0           PUSH   B               ;SAVE B + C
4811  3A  B0  48              13.0           LDA    INDAT           ;GET INPUT DATA
4814  E6  30                   7.0           ANI    $30             ;REMOVE THE WORKING BITS
4816  07                       4.0           RLC                    ;POSITION THE DATA
4817  07                       4.0           RLC
4818  07                       4.0           RLC
4819  07                       4.0           RLC
481A  21  27  48              10.0           LXI    H,POINTBL       ;GET BASE ADDRESS OF TABLE
481D  4F                       4.0           MOV    C,A
481E  06  00                   7.0           MVI    B,0
4820  09                      10.0           DAD    B
4821  7E                       7.0           MOV    A,M             ;GET THE ECC POINTER
4822  D3  0A                  10.0           OUT    RTIEB           ;SET THE TIE BUS VALUE
4824  C1                      10.0           POP    B               ;RESTORE B + C
4825  F1                      10.0           POP    PSW             ;RESTORE ACCUMULATOR
4826  C9                      10.0           RET

4827  01                           POINTBL:  .BYTE  1
4828  02                                     .BYTE  2
4829  04                                     .BYTE  4
482A  08                                     .BYTE  8
```

```
                                .SBITL  TABLE 4 X 5 TRANSLATION

                                ;THIS TABLE TRANSLATED DATA FOR AS FOLLOWS
                                ;  INPUT GROUP POSITIONS          OUTPUT GROUP POSITIONS
                                ;   1 2 3 4 / 5 6 7 8             10 9 8 7 6 / 5 4 3 2 1
                                ;
                                ;       0000                          10011
                                ;       0001                          11011
                                ;       0010                          01001
                                ;       0011                          11001
                                ;       0100                          10111
                                ;       0101                          10101
                                ;       0110                          01101
                                ;       0111                          11101
                                ;       1000                          01011
                                ;       1001                          10010
                                ;       1010                          01010
                                ;       1011                          11010
                                ;       1100                          01111
                                ;       1101                          10110
                                ;       1110                          01110
                                ;       1111                          11110

482B  13                        TAB4X5: .BYTE   @23
482C  1B                                .BYTE   @33
482D  09                                .BYTE   @11
482E  19                                .BYTE   @31
482F  17                                .BYTE   @27
4830  15                                .BYTE   @25
4831  0D                                .BYTE   @15
4832  1D                                .BYTE   @35
4833  0B                                .BYTE   @13
4834  12                                .BYTE   @22
4835  0A                                .BYTE   @12
4836  1A                                .BYTE   @32
4837  0F                                .BYTE   @17
4838  16                                .BYTE   @26
4839  0E                                .BYTE   @16
483A  1E                                .BYTE   @36
```

```
                                        .SBTTL  SUBROUTINE VARIABLES

                        TRNIN:  .BLKB   4       ;TABLE CONTAINING THE 4 DATA BYTES TO
                                                ;BE TRANSLATED
                        TRNOUT: .BLKB   10      ;TRANSLATED DATA TABLE DATA AFTER 4X5
                                                ;TRANSLATION
                        GP4X5:  .BYTE   0       ;ACCUMULATOR FOR THE 4 BIT SUBGROUP TO
                                                ;BE TRANSLATED
                        TRNTMP: .BYTE   0       ;TEMP. STORAGE FOR THE TRANSLATED SUBGROUP.
                        BAD1:   .BYTE   0       ;BAD TRANSLATION POINTER 1
                        BAD2:   .BYTE   0       ;BAD TRANSLATION POINTER 2
                        BAD3:   .BYTE   0       ;BAD TRANSLATION POINTER 3
```

```
2562                                    .SBTTL  SUBROUTINE CLEAR ECC
2563  484E                              S
 (1)                                    ; ************************************************************************
2564                                    ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION ''ECCCHR'' -
2565                                    ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
2566                                    ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
2567  484E                              S
 (1)                                    ; ************************************************************************
2568
2569  484E  AF                   4.0   CLECC:  XRA     A               ;CLEAR THE ACCUMULATOR
2570  484F  32   76   48        13.0          STA     ECCCHR          ;CLEAR THE ECC CHARACTER
2571  4852  C9                  10.0          RET                     ;RETURN TO USER
2572
```

```
                                        .SBTTL  SUBROUTINE CALCUALTE ECC CHARACTER
                                        $
                                        ; ************************************************************************
 (1)
                                        ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
                                        ;THE CONTENTS OF "ECCCHR" TO UPDATE THE CONTENTS OF "ECCCHR" ACCORDING
                                        ;TO THE ANSI STANDARD ECC POLYNOMIAL.
                                        $
                                        ; ************************************************************************
 (1)

4853 21 76 48      10.0  ECC:    LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
4856 AE             7.0          XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
4857 5F             4.0          MOV     E,A             ;SAVE XOR RESULT IN E
4858 E6 10          7.0          ANI     $10             ;IS BIT #4 OF RESULT SET
485A 7B             4.0          MOV     A,E             ;RESTORE XOR RESULT FROM B
485B CA 60 48      10.0          JZ      ECC1            ;CONTINUE IF BIT #4 RESET
485E EE 23          7.0          XRI     $23             ;ELSE-XOR WITH 23
4860 5F             4.0  ECC1:   MOV     E,A             ;STORE THE ECC RESULT IN E
                                                         ;
4861 AF             4.0          XRA     A               ;CLEAR A
4862 4F             4.0          MOV     C,A             ;CLEAR THE TRANSLATE RESULT
4863 21 6E 48      10.0          LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
4866 CD 77 48      18.0          CALL    TRANS           ;TRANSLATE THE BITS
4869 79             4.0          MOV     A,C             ;GET THE TRANSLATED RESULT
486A 32 76 48      13.0          STA     ECCCHR          ;STORE RESULT
486D C9            10.0          RET

                                 ECCTBL: $08             ;BIT 0 = POSITION 3
                                         $20             ;BIT 1 = POSITION 5
                                         $02             ;BIT 2 = POSITION 1
                                         $40             ;BIT 3 = POSITION 6
                                         $80             ;BIT 4 = POSITION 7
                                         $01             ;BIT 5 = POSITION 0
                                         $10             ;BIT 6 = POSITION 4
                                         $04             ;BIT 7 = POSITION 2

4876 00                          ECCCHR: .BYTE   0
```

```
                                        .SBTTL  SUBROUTINE POLYNOMIAL BIT TRANSLATION

                                        ;
                                        ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
                                        ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
                                        ;

4877  06  01          7.0  TRANS:  MVI   B,1          ;INIT 'B' TO BIT POSITION 0
4879  7B              4.0  TRANS1: MOV   A,E          ;GET CHAR TO BE TRANSLATED
487A  A0              4.0          ANA   B          ;SEE IF BIT POSITION IN 'B' IS SET
487B  CA  81  48     10.0          JZ    TRANS2     ;DO NEXT BIT POSITION IF NOT SET
487E  79              4.0          MOV   A,C        ;GET PREVIOUS RESULT OF 'OR'
487F  B6              7.0          ORA   M          ;'OR' IN NEW POSITION
4880  4F              4.0          MOV   C,A        ;SAVE RESULT

4881  78              4.0  TRANS2: MOV   A,B
4882  07              4.0          RLC              ;POSITION MASK TO NEXT BIT
4883  47              4.0          MOV   B,A
4884  D8             12.0          RC               ;EXIT WHEN ALL POSITIONS DONE
4885  23              6.0          INX   H          ;POINT TO NEXT TABLE ENTRY
4886  C3  79  48     10.0          JMP   TRANS1     ;PROCESS NEXT BIT
```

```
                                .SBTTL  SUBROUTINE TRK
4889                            SSUB
                                ; ************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*GENERATE BAD TRACK TRANSLATION
4889                            SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE GENRATES A TRACK NUMBER FOR THE BADT4X5 SUBROUTINE
                                ;*BASED ON THE VALUE OF INDAT.
                                ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
                                ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
                                ;*MESSAGE, AND TRACK SIMULATED IN ERROR CAN BE DETERMINED FROM THE
                                ;*FOLLOWING TABLE.
                                ;*
                                ;*      INPUT/EXPECTED          DATA BIT POSITION       TRACK WITH ECC
                                ;*          DATA                WITH ECC CORRECTION       CORRECTION
                                ;*      ----------------        -------------------     -------------
                                ;*
                                ;*          XXXX0000                    0                   2
                                ;*          XXXX0001                    1                   8
                                ;*          XXXX0010                    2                   1
                                ;*          XXXX0011                    3                   9
                                ;*          XXXX0100                    4                   3
                                ;*          XXXX0101                    5                   5
                                ;*          XXXX0110                    6                   6
                                ;*          XXXX0111                    7                   7
                                ;*          XXXX1000                    P                   4
                                ;*          XXXX1001                    1                   8
                                ;*          XXXX1010                    2                   1
                                ;*          XXXX1011                    3                   9
                                ;*          XXXX1100                    4                   3
                                ;*          XXXX1101                    5                   5
                                ;*          XXXX1110                    6                   6
                                ;*          XXXX1111                    7                   7
                                ;*
                                ;*      X = DON'T CARE BITS
4889                            S
                                ; ************************************************************************

4889  3A  B0  48      13.0  TRK:    LDA     INDAT           ;GET INPUT DATA
488C  E6  0F              7.0       ANI     $0F             ;REMOVE 4 BITS
488E  FE  08              7.0       CPI     $8              ;PARITY TRACK?
4890  CA  95  48        10.0       JZ      TRKA            ;YES
4893  E6  07              7.0       ANI     $7              ;NO-IGNORE BIT3
4895  3C                  4.0  TRKA:  INR     A               ;INCREMENT FOR SUBROUTINE
4896  32  4B  48        13.0       STA     BAD1            ;SAVE IT.
4899  3D                  4.0       DCR     A               ;PREPARE TO USE AS TABLE INDEX
489A  21  A6  48        10.0       LXI     H,TIMTBL        ;TIE BUS TIME TABLE START
489D  5F                  4.0       MOV     E,A             ;LOW HALF OF TABLE INDEX
489E  16  00              7.0       MVI     D,0             ;ZERO HIGH HALF
48A0  19                 10.0       DAD     D               ;PRODUCE TABLE ENTRY ADDRESS
```

```
2680  48A1  7E              7.0          MOV     A,M             ;GET TIE BUS TIME FROM TABLE
2681  48A2  32  AF  48      13.0         STA     TIETIM          ;SAVE VALUE
2682  48A5  C9              10.0         RET
2683  48A6  02                     TIMTBL: .BYTE   2               ;TIE BUS TIME FOR BIT 0
2684  48A7  03                             .BYTE   3               ;TIE BUS TIME FOR BIT 1
2685  48A8  00                             .BYTE   0               ;TIE BUS TIME FOR BIT 2
2686  48A9  01                             .BYTE   1               ;TIE BUS TIME FOR BIT 3
2687  48AA  04                             .BYTE   4               ;TIE BUS TIME FOR BIT 4
2688  48AB  08                             .BYTE   8               ;TIE BUS TIME FOR BIT 5
2689  48AC  07                             .BYTE   7               ;TIE BUS TIME FOR BIT 6
2690  48AD  05                             .BYTE   5               ;TIE BUS TIME FOR BIT 7
2691  48AE  06                             .BYTE   6               ;TIE BUS TIME FOR BIT P
2692
2693
```

```
                                        .SBTTL  PROGRAM VARIABLES

                                TIETIM: .BYTE   0               ;TIE BUS JAM TIME
                                INDAT:  .BYTE   0               ;DATA BYTE TO ECC
                                INDATP: .BYTE   0               ;DATA BYTE PARITY
                                UNITMP: .BYTE   0               ;UNIT MAP
                                SUNIT:  .BYTE   0               ;MASS BUS/TU SELECT BITS
                                AMTMSK: .WORD   0               ;AMTIE MASK WORD
                                DATAA:  .BYTE   0               ;ACTUAL DATA AFTER TRANSLATION
                                DATAAP: .BYTE   0               ;ACTUAL DATA PARITY AFTER TRANSLATION
                                DATAE:  .BYTE   0               ;EXPECTED DATA AFTER TRANSLATION
                                DATAEP: .BYTE   0               ;EXPECTED DATA PARITY AFTER TRANSLATION
                                        .END
```

```
A      =%0007   ADATA  = 0094   AMTIEP= 0001   AMTIE7= 0002
AMTMSK  48B4    ARAIDF= 0098   ASAVE   4F9B   ATTCD   4F97
AXNUM   4F91    B      =%0000   BAD     4773   BADST = 0090
BADT4X  4764    BAD1     484B   BAD1C   47C0   BAD2    484C
BAD2C   47CB    BAD3     484D   BAD3C   47DD   BIT0  = 0001
BIT1  = 0002    BIT15 = 8000   BIT2  = 0004   BIT3  = 0008
BIT4  = 0010    BIT5  = 0020   BIT6  = 0040   BIT7  = 0080
BIT8  = 0100    BIT9  = 0200   BRKPBC= 4F0A   BRKRAM= 4F10
BRKSTR= 4E60    BRKXCT= 4F00   BSAVE   4F9C   BYTCNT= 00D4
BYTEH   4F24    BYTEL   4F23   B4X5    4784   C      =%0001
CASCT   4F21    CASCTL= 00A0   CASSTA= 00A0   CATTH = 0089
CATTL = 0088    CBUSST= 00A1   CBYTH = 008B   CBYTL = 008A
CDG1H = 0087    CDG1L = 0086   CDG2H = 0093   CDG2L = 0092
CDG3H = 0095    CDG3L = 0094   CDVTH = 008D   CDVTL = 008C
CHPTIE= 0028    CHOTIE= 0020   CH1TIE= 0021   CH2TIE= 0022
CH3TIE= 0023    CH4TIE= 0024   CH5TIE= 0025   CH6TIE= 0026
CH7TIE= 0027    CKLOP = 2815   CLEAR   4739   CLECC   484E
CLKCTL= 00F0    CLOCK   4F26   CLRLP   473D   CMC0H = 0099
CMC0L = 0098    CMC1H = 009B   CMC1L = 009A   CMC2H = 009D
CMC2L = 009C    CMC3H = 009F   CMC3L = 009E   CMINH = 0097
CMINL = 0096    CNTCTL= 00D7   CRCWRD= 0018   CSAVE   4F9D
CSRLH = 0091    CSRLL = 0090   CTCH  = 0085   CTCL  = 0084
CTSTH = 008F    CTSTL = 008E   CXCTH = 0081   CXCTL = 0080
CXINH = 0083    CXINL = 0082   C.     = 0001   C.AVAI= 0080
C.DP  = 0008    C.DSE = 0010   C.DTU = 0003   C.DVA = 0008
C.FAIL= 00FC    C.FMT = 0070   C.FNCT= 003E   C.GO  = 0001
C.INTC= 00FE    C.MAIN= 0020   C.NSA = 0080   C.RCT = 00FC
C.SER = 0080    C.SHR = 0040   C.SKPC= 000F   C.TAPE= 0040
C.WCS = 0002    C4X5    478D   D      =%0002   DATAA   48B6
DATAAP  48B7    DATACT= 00D0   DATAE   48B8   DATAEP  48B9
DBUS    4F28    DBUSCT= 00C0   DBUSST= 00C0   DDRA  = 00D8
DDRAIN= 0010    DDRB  = 00D9   DDRBIN= 0002   DDRC  = 00DA
DDRCIN= 0001    DDRCO = 0088   DDRCTL= 00DB   DIAFLG  4F22
DIAGPG= 4300    DIAGRM= 4F90   DIARD = 000B   DONE1 = 0045
DONINT= 0010    DSAVE   4F9E   DSE   = 0006   DUMMU   460C  G
DUMMV   4652  G DUMMW   444C  G DUMMX   4419  G DUMMY   431E  G
D.ATH0= 0001    D.ATH1= 0002   D.EOTD= 0010   D.LAGC= 0020
D.NOTW= 0040    D.NTHR= 0004   D.TACH= 0008   D.WR4 = 0080
D4X5    477C    E      =%0003   ECC     4853   ECCBAD= 0042
ECCCHR  4876    ECCCOR= 0019   ECCOK = 0041   ECCSTA= 001A
ECCTBL  486E    ECCTST= 000E   ECC1    4860   EDATA = 0095
EOTCLR= 0003    ERFLG   4F93   ERLP  = 2809   ERLPA = 280F
ERLPB = 2812    ERLPE = 280C   ERNUM   4F90   ERRCNT= 00D6
ESAVE   4F9F    EXIT    4736   E.ACRC= 0010   E.AMT = 0020
E.CDP = 0080    E.CRC = 0080   E.PNTR= 0008   E.RPE = 0040
E.STEC= 0001    E.TTEC= 0002   E.UNC = 0004   E4X5    47A6
FIFORD= 006A    FORMAT  4F25   FOUND   4344   FWDTST= 0061
F4X5    47F2    GCRID = 0089   GCRSET= 00C2   GOODTM= 0092
GP4X5   4849    H      =%0004   HLSAVE  4FA0   IE    = 0008
INDAT   48B0    INDATP  48B1   INTSTA= 00E0   INV4X5  47D6
ITERA   4F9A    I.PWR = 0020   I.RMPE= 0040   I5.5  = 0010
I6.5  = 0020    I7.5  = 0040   KCALL = 005F   KCLR  = 007B
KDEP  = 003F    KENAB = 0078   KEXAM = 003E   KEYBRD= 00C8
KEY1  = 0078    KEY10 = 006D   KEY11 = 006E   KEY12 = 006F
```

```
KEY13 = 005C    KEY14 = 005D    KEY15 = 005E    KEY16 = 005F
KEY17 = 003C    KEY18 = 003D    KEY19 = 003E    KEY2  = 0079
KEY20 = 003F    KEY3  = 007A    KEY4  = 007B    KEY5  = 0074
KEY6  = 0075    KEY7  = 0076    KEY8  = 0077    KEY9  = 006C
KINTA = 006F    KLDAD = 003D    KN0   = 003C    KN1   = 005C
KN2   = 005D    KN3   = 005E    KN4   = 006C    KN5   = 006D
KN6   = 006E    KN7   = 0074    KN8   = 0075    KN9   = 0076
KU2   = 0079    KU3   = 007A    KU8   = 0077    L     =%0005
LBLANK= 000F    LCE   = 000B    LCH   = 000C    LCL   = 000D
LCP   = 000E    LC0   = 0000    LC1   = 0001    LC2   = 0002
LC3   = 0003    LC4   = 0004    LC5   = 0005    LC6   = 0006
LC7   = 0007    LC8   = 0008    LC9   = 0009    LDLEDA= 00CA
LDLEDB= 00CB    LDLEDC= 00CC    LDLEDD= 00CD    LDLEDE= 00CE
LDLEDF= 00CF    LKDIAG= 2800    LKKBD = 004C    LKKEY = 0049
LKLWMG= 0058    LKLWMP= 0055    LKLWPG= 0052    LKLWPP= 004F
LKMOD7= 0046    LKOPR = 0046    LPFLG   4F94    LPNUM   4F92
M     =%0006    MBSEL = 00E0    MB.A  = 0008    MB.B  = 0004
MEMTOP= 4FFF    MINUS = 000A    MM    = 8000    MSE   = 0008
MSGN  = 000F    MTACLR= 0000    MT.ARA= 0020    MT.CPE= 0080
MT.DSE= 0001    MT.FWD= 0040    MT.INH= 0008    MT.LWR= 0004
MT.MOT= 0002    MT.NWT= 0080    MT.PEC= 0040    MT.PSB= 0004
MT.PS0= 0001    MT.PS1= 0002    MT.REV= 0020    MT.WRT= 0010
MT.Z  = 0008    M.ATA = 0080    M.CAPE= 0020    M.CONT= 0080
M.DEM = 0020    M.EBL = 0010    M.EXC = 0008    M.FAIL= 0008
M.ILR = 0010    M.INIT= 0010    M.OCC = 0020    M.ONLI= 0001
M.PE  = 0040    M.PORT= 0080    M.RDEN= 0002    M.RDPE= 0008
M.RUN = 0004    M.SCLK= 0001    M.TRA = 0040    M.UNIT= 0007
M.WCLK= 0040    M.WCLN= 0080    M.WREN= 0080    M5.5  = 0001
M6.5  = 0002    M7.5  = 0004    NOTCAP= 0088    OKAY  = 00FF
OPRRAM= 4300    OPSTRT= 0058    OPVER = 0040    PADCNT= 00D5
PADCRC= 0080    PDIAG = 0048    PEID  = 008A    PENAB = 004C
PESET = 0001    PL    = 00B1    POINTB  4827    POINTE  480F
PO4X5   4793    PRDD  = 004C    PRENF = 009C    PS    = 00B2
PSTAT = 0048    PSW   =%0009    P.AMTP= 0001    P.BCTC= 0040
P.CMDP= 0020    P.INTE= 0080    P.LCS = 0040    P.LWR = 0020
P.RDP = 0002    P.RPEN= 0004    P.RPST= 0002    P.RPOE= 0020
P.RP1E= 0010    P.RP2E= 0020    P.RP3E= 0010    P.SING= 0080
P.STAT= 0002    P.STPE= 0080    P.TACH= 0008    P.TUPR= 0010
P.WCSP= 0004    P.WDS = 0040    P.WFLP= 0001    P.WPEN= 0010
P.WPOE= 0008    P.WP1E= 0004    P.WP2E= 0008    P.WP3E= 0004
P.5VOK= 0002    QUE   = 281B    QUEM  = 281E    RAMT  = 0010
RARA  = 0006    RARAI = 0004    RCHBD0= 0048    RCHBD1= 0047
RCHOK = 0046    RCHTST= 000C    RCLRT = 000D    RCMD  = 000B
RCMLP = 0003    RCONT = 0080    RDATA = 0017    RDCLK = 0010
RDON  = 0011    READG = 0007    REND  = 0014    REQST = 2806
RESCHR= 00D1    REVTST= 0064    REWIND= 0004    RFIFOL= 0008
RGCLK = 0002    RGCRI = 0003    RIBG  = 0001    RILL  = 0012
RINST = 000C    RMCTST= 0008    RMK2  = 0013    RNOP  = 0000
RPATH = 0001    RPBAD = 0044    RPCHI = 0001    RPCLK = 0003
RPCTL = 0009    RPEI  = 0002    RPFAIL= 0000    RPF1  = 009D
RPF2  = 009E    RPOK  = 0043    RPOSTN= 0016    RPSTA = 0015
RRCMT = 000A    RSTAT = 0002    RTIEB = 000A    RTIER = 0030
RTM   = 0005    RUNKI = 0009    RUPTST= 005E    RWDUNL= 0005
R.AMT = 0001    R.BOP = 0008    R.DATA= 0040    R.DON = 0002
```

```
R.DRDY= 0010        R.END = 0010        R.ILL = 0004        R.JVOK= 0004
R.MK2 = 0008        R.PLOD= 0008        R.PL00= 0010        R.PL01= 0020
R.POST= 0020        R.STNM= 0002        R.STOP= 0004        R.STPC= 0001
R.TBJN= 0080        R.TSTD= 0040        R.VOK = 0080        R00H  = 0081
R00L  = 0080        R01H  = 0083        R01L  = 0082        R02H  = 0085
R02L  = 0084        R03H  = 0087        R03L  = 0086        R04H  = 0089
R04L  = 0088        R05H  = 008B        R05L  = 008A        R06H  = 008D
R06L  = 008C        R07H  = 008F        R07L  = 008E        R10H  = 0091
R10L  = 0090        R11H  = 0093        R11L  = 0092        R12H  = 0095
R12L  = 0094        R13H  = 0097        R13L  = 0096        R14H  = 0099
R14L  = 0098        R15H  = 009B        R15L  = 009A        R16H  = 009D
R16L  = 009C        R17H  = 009F        R17L  = 009E        R7.5  = 0010
SELCLR= 0000        SETATA= 00A1        SID   = 0080        SOD   = 0080
SOE   = 0040        SP    =%0008        SSCLK = 0040        SSTEP = 0005
STACK = 4FFF        STATRM= 4F20        STPCT   4F20        STRSP = 5000
SUNIT   48B3        TAB4X5  482B        TADR00= 0080        TADR01= 0081
TADR02= 0082        TADR03= 0083        TADR04= 0084        TADR05= 0085
TADR06= 0086        TADR07= 0087        TADR10= 0088        TADR11= 0089
TADR12= 008A        TADR13= 008B        TAMT  = 0044        TASEL = 0080
TCMD  = 0040        TC.FWD= 0040        TC.INH= 0008        TC.LWR= 0004
TC.REV= 0020        TC.WRT= 0010        TEMP    4F99        TEST1   4300
TEST2   4533        TIETIM  48AF        TIMTBL  48A6        TMF   = 0099
TMRDY = 0040        TRANS   4877        TRANS1  4879        TRANS2  4881
TRK     4889        TRKA    4895        TRKENA= 00D2        TRNIN   483B
TRNOUT  483F        TRNTMP  484A        TSET  = 2803        TSTEND= 281B
TSTS  = 0040        TST01A  4422        TST01C  4379        TST01F  4391
TST01G  43BA        TST01H  43E8        TST01I  439D        TST01K  4455
TST01L  446B        TST01M  4483  G     TST01N  44CD        TST01Q  44AE
TST01R  44A5  G     TST01S  44C2        TST01T  44BC  G     TST01U  44EA  G
TST01V  450C  G     TST01X  4351        TST01Y  4457        TST02A  4615
TST02B  4619        TST02C  456F        TST02F  4587        TST02G  45AD
TST02H  45DB        TST02I  4593        TST02K  465B        TST02L  4671
TST02M  4689  G     TST02N  46D3        TST02Q  46B4        TST02R  46AB  G
TST02S  46C8        TST02T  46C2  G     TST02U  46F0  G     TST02V  4712  G
TST02X  4542        TST02Y  465D        TUSEL0= 00D1        TUSEL1= 00D2
TU78  = 0010        T.ATH0= 0001        T.ATH1= 0002        T.BOT = 0004
T.EOT = 0002        T.FPT = 0001        T.NTHR= 0004        T.ONL = 0020
T.PES = 0008        T.PSBJ= 0020        T.PSOJ= 0008        T.PS1J= 0010
T.RDY = 0080        T.RDYO= 0040        T.RWD = 0010        T.SCLK= 0002
T4X5    4768        UIBG  = 00A1        UNITMP  48B2        UNITSL  4327
VALFC   4F98        VALTB   4F95        VELTST= 005B        WDR.P = 0010
WMCCTL= 00D3        WMCERR= 00DA        WMCSTA= 00D0        WRTCLK= 0000
WRTDAT= 00D3        W.ACRC= 0004        W.CRC = 0008        W.DIAG= 0002
W.DONN= 0040        W.ECC = 0010        W.ENAB= 0080        W.ERR = 0020
W.FMT = 0070        W.GCR = 0010        W.LEFT= 0004        W.ONES= 0020
W.RESI= 0002        W.REV = 0004        W.ROME= 0010        W.RST = 0001
W.SKIP= 000F        W.WRIT= 0008        W.XFER= 0020        X     =%000A
X.DONN= 0080        X.ENAB= 0040        X.PEPE= 00C2        X.ROME= 0001
X.WCLK= 0001        Y     =%000B        .     = 48BA
```

ERRORS DETECTED: 0

```
*ECC2.A78/PTP,ECC2=NLIST,PARAM,MACRO,LIST,ECC2
RUN-TIME: 5 7 0 SECONDS
CORE USED:  10K
```

     3   "MEMORY MAP" EXTENDED ADDRESS BIT
    19   READ MICRO CONTROLLER CONTROL & STATUS WORDS
   139   TRANSPORT/PORT COMMAND + STATUS WORDS
   380   COMMON ADDRESS SPACE REGISTERS
   444   MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
   485   MASSBUS CONTROL WORD DEFINITIONS
   557   WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
   709   KEYBOARD/DISPLAY CONTROL + STATUS WORD
   797   MISCELLANEOUS VARIABLE DEFINITIONS
   801   RAM ADDRESS DEFINITIONS
   819   LINKAGE TABLE ADDRESSES
   842   BIT DEFINITIONS
   856   RIM AND SIM INSTRUCTION BIT DEFINITIONS
   877   INTERRUPTABLE INSTRUCTION MACROS
   924   DIAGNOSTIC VARIABLES
   945   COMMON VARIABLE STORAGE
   959   MACRO CALL DEFINITIONS
  1047   TM78 MICRO-DIAGNOSTIC MACRO CALLS
  1101   8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
  1333   TEST 1 - ECC BAD DATA ONE TRACK INCORRECT POINTER
  1835   SUBROUTINE CLEAR ALL TU PORTS
  1880   SUBROUTINE 4X5 TRANSLATE A SUBGROUP
  2024   SUBROUTINE CALCULATE POINTER
  2064   TABLE 4 X 5 TRANSLATION
  2105   SUBROUTINE VARIABLES
  2119   SUBROUTINE CLEAR ECC
  2131   SUBROUTINE CALCUALTE ECC CHARACTER
  2167   SUBROUTINE POLYNOMIAL BIT TRANSLATION
  2190   SUBROUTINE TRK
  2252   PROGRAM VARIABLES

```
                        ;*************************************************************************
                        ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                        ;ERROR MACRO CALLS
                        ;
                        ;       1  -    REQUEST HOST CPU TO PRINT:
                        ;               "BYTE/SCLK COUNT NUMBER = LLL"
                        ;               WHERE:
                        ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                        ;                     (THE BYTE COUNT REGISTER 16 BITS).
                        ;
                        ;       2  -    REQUEST HOST CPU TO PRINT:
                        ;               DATA FORMAT = MM
                        ;               SKIP COUNT  = NN
                        ;               WHERE:
                        ;               MM = DATA FORMAT FROM CAS REGISTER 2
                        ;               NN = SKIP COUNT FROM CAS REGISTER 2
                        ;
                        ;       3  -    REQUEST HOST CPU TO PRINT:
                        ;               BYTE-SCLK COUNT = LLL
                        ;               DATA FORMAT = MM
                        ;               SKIP COUNT  = NN
                        ;               WHERE:
                        ;               LLL = AS ABOVE
                        ;               MM  = AS ABOVE
                        ;               NN  = AS ABOVE
                        ;
                        ;       4  -    REQUEST HOST TO PRINT:
                        ;               TRANSITION COUNT - LLL
                        ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
                        ;
                        ;       5  -    REQUEST HOST CPU TO PRINT:
                        ;               EXPECTED 18 BITS =      E EEEEEE
                        ;               ACTUAL 18 BITS =        A AAAAAA
                        ;
                        ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                        ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                        ;               OF CAS REG 15 LOW BYTE.
                        ;
                        ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                        ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                        ;               17 LOW BYTE SIGN BIT.
                        ;
                        ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                        ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                        ;               AND/OR ACTUAL DATA.
                        ;
                        ;       7  -    REQUEST HOST CPU TO PRINT:
                        ;               "SUBGROUP NUMBER = LLL"
                        ;               WHERE:
                        ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                        ;                     (THE BYTE COUNT REGISTER 16 BITS).
                        ;
                        ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                        ;*************************************************************************
```

```
1016                    ;*************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO·
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ;*************************************************************************
1035                    ;*************************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ;*************************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                        .TITLE  ECC3 - ECC CONTROLLER PART #3
                        .SBTTL  TEST 1 - ECC BAD DATA ONE TRACK INCORRECT POINTER
                        ;ID     ECC3-ERROR CORRECTION CONTROLLER PART #3
4300                    ST
                        ; ***************************************************************************
                        ;*TEST TITLE
                        ;*---- -----
                        ;*ECC BAD DATA ONE TRACK INCORRECT POINTER
4300                    SD
                        ; ***************************************************************************
                        ;*DESCRIPTION
                        ;*-----------
                        ;*THIS TEST CHECKS THE ABILITY OF THE ECC CONTROLLER TO PERFORM SINGLE
                        ;*TRACK ERROR CORRECTION WITH INCORRECT DATA IN A SINGLE TRACK, AND A
                        ;*INCORRECT POINTER ALSO PROVIDED.  THE PROGRAM CHECKS THAT THE DATA IS
                        ;*CORRECTED, AND THAT THE SINGLE TRACK ERROR CORRECT STATUS BIT AND THE
                        ;*POINTER ERROR STATUS BIT ARE SET.
                        ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
                        ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
                        ;*MESSAGE, AND THE DATA BIT POSITION OF THE INCORRECT POINTER
                        ;*CAN BE DETERMINED FROM THE FOLLOWING TABLE.
                        ;*
                        ;*      INPUT/EXPECTED          DATA BIT POSITION       BIT POSITION OF
                        ;*          DATA                WITH ECC CORRECTION       INCORRECT POINTER
                        ;*      -------------           -------------------     -----------------
                        ;*
                        ;*          XXXX0000                    0                       1
                        ;*          XXXX0001                    1                       2
                        ;*          XXXX0010                    2                       3
                        ;*          XXXX0011                    3                       4
                        ;*          XXXX0100                    4                       5
                        ;*          XXXX0101                    5                       6
                        ;*          XXXX0110                    6                       7
                        ;*          XXXX0111                    7                       P
                        ;*          XXXX1000                    P                       0
                        ;*          XXXX1001                    1                       2
                        ;*          XXXX1010                    2                       3
                        ;*          XXXX1011                    3                       4
                        ;*          XXXX1100                    4                       5
                        ;*          XXXX1101                    5                       6
                        ;*          XXXX1110                    6                       7
                        ;*          XXXX1111                    7                       P
                        ;*
                        ;*      X = DON'T CARE BITS
                        ;*
                        ;*THE POINTERS PROVIDED FOR THE TRACK IN ERROR ARE ALSO DERIVED FROM THE
                        ;*INPUT/EXPECTED DATA AS PER THE FOLLOWING TABLE.
                        ;*
                        ;*      INPUT/EXPECTED
                        ;*          DATA                    POINTER GENERATED
                        ;*      -------------               -----------------
                        ;*
                        ;*          XX00XXXX                HISTORY
                        ;*          XX01XXXX                PHASE TIE
```

```
                        ;*          XX10XXXX                    AMTIE
                        ;*          XX11XXXX                    ILLEGAL 5X4
                        ;*
                        ;*       X = DON'T CARE BITS
4300                    SP
 (1)                    ; ***************************************************************************
 (1)                    ;*PROCEDURE
 (1)                    ;*---------
                        ;*BGNTST
                        ;*     SET NORMAL READ PATH CLOCK
                        ;*     CALL SUBROUTINE CLEAR
                        ;*     SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
                        ;*     CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
                        ;*     CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
                        ;*     CLEAR THE INPUT DATA
                        ;*     CLEAR BAD TRANSLATION POINTERS
                        ;*   BGNDO
                        ;*   : ENABLE THE READ PATH CLOCK
                        ;*   : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
                        ;*   : CLOCK THE FIFO'S
                        ;*   : ISSUE CLEAR ALL COMMAND
                        ;*   : WAIT
                        ;*   : STOP THE READ PATH, SET TIE BUS JAM AND SET PLO DISABLE
                        ;*   : CALL SUBROUTINE CLECC
                        ;*   : SET THE LOOP COUNT TO 7
                        ;*   : BGNDO
                        ;*   : :   GET THE INPUT DATA BYTE
                        ;*   : :   CALL SUBROUTINE ECC
                        ;*   : :   DECREMENT THE LOOP COUNT
                        ;*   : :   DO UNTIL THE LOOP COUNT=0
                        ;*   : ENDDO
                        ;*   : SET THE LOOP COUNT TO 3
                        ;*   : BGNDO
                        ;*   : :   FILL THE TRANSLATOR SUBROUTINE BUFFER WITH THE DATA BYTE
                        ;*   : :   CALL SUBROUTINE TRK
                        ;*   : :   CALL SUBROUTINE BADT4X5
                        ;*   : :   INIT THE LOOP COUNT TO 5
                        ;*   : :   BGNDO
                        ;*   : :   :   MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
                        ;*   : :   :   INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
                        ;*   : :   :   MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
                        ;*   : :   :   CLOCK THE DATA INTO THE FIFO
                        ;*   : :   :   DECREMENT THE LOOP COUNT
                        ;*   : :   :   INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
                        ;*   : :   :   DO UNTIL LOOP COUNT = 0
                        ;*   : :   ENDDO
                        ;*   : :   FILL THREE BYTES OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE DATA BYTE
                        ;*   : :   FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
                        ;*   : :   CALL SUBROUTINE TRK
                        ;*   : :   CALL SUBROUTINE BADT4X5
                        ;*   : :   INIT THE LOOP COUNT TO 5
                        ;*   : :   BGNDO
                        ;*   : :   :   MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
                        ;*   : :   :   MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
```

```
1431                    ;*  : : :  CLOCK THE DATA INTO THE FIFO
1432                    ;*  : : :  DECREMENT THE LOOP COUNT
1433                    ;*  : : :  DO UNTIL THE LOOP COUNT = 0
1434                    ;*  : : ENDDO
1435                    ;*  : : DECREMENT THE LOOP COUNT
1436                    ;*  : : DO UNTIL THE LOOP COUNT = 0
1437                    ;*  : ENDDO
1438                    ;*  : ISSUE DIAGNOSTIC READ COMMAND
1439                    ;*  : SET UP WATCHDOG TIMER COUNT
1440                    ;*  : BGNDO
1441                    ;*  : :    SINGLE STEP THE READ PATH
1442                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1443                    ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1444                    ;*  : ENDDO
1445                    ;*  : IF WATCHDOG TIMER=0
1446                    ;*  : :    THEN-ERROR
1447                    ;*  : :    ELSE-CONTINUE
1448                    ;*  : ENDIF
1449                    ;*  : GET THE TIE BUS TABLE ADDRESS
1450                    ;*  : SINGLE STEP THE READ PATH
1451                    ;*  : INIT THE LOOP COUNT TO 0
1452                    ;*  : BGNDO
1453                    ;*  : :    IF LOOP COUNT=TIE BUS JAM COUNT
1454                    ;*  : :    :  THEN-CALL SUBROUTINE POINTER
1455                    ;*  : :    :  ELSE-CLEAR TIE BUS POINTER
1456                    ;*  : :    ENDIF
1457                    ;*  : :    LOAD THE TIE BUS REGISTER
1458                    ;*  : :    SINGLE STEP THE READ PATH
1459                    ;*  : :    DECREMENT THE LOOP COUNT
1460                    ;*  : :    DO UNTIL LOOP COUNT = 9
1461                    ;*  : ENDDO
1462                    ;*  : SET UP WATCHDOG TIMER COUNT
1463                    ;*  : BGNDO
1464                    ;*  : :  SINGLE STEP THE READ PATH
1465                    ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
1466                    ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
1467                    ;*  : ENDDO
1468                    ;*  : IF WATCHDOG TIMER=0
1469                    ;*  : :  THEN-ERROR-EXIT TEST
1470                    ;*  : :  ELSE-CONTINUE
1471                    ;*  : ENDIF
1472                    ;*  : SINGLE STEP THE READ PATH
1473                    ;*  : INIT THE  LOOP COUNT TO 7
1474                    ;*  : BGNDO
1475                    ;*  : :  SINGLE STEP THE READ PATH
1476                    ;*  : :  COMPARE CORRECTED DATA WITH INPUT DATA
1477                    ;*  : :  IF NOT EQUAL
1478                    ;*  : :  :  THEN-ECC ERROR
1479                    ;*  : :  :  ELSE-CONTINUE
1480                    ;*  : :  ENDIF
1481                    ;*  : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
1482                    ;*  : :  IF NOT EQUAL
1483                    ;*  : :  :  THEN-ECC ERROR
1484                    ;*  : :  :  ELSE-CONTINUE
```

```
1485                                ;*  : :  ENDIF
1486                                ;*  : :  DECREMENT THE LOOP COUNT
1487                                ;*  : :  DO UNTIL THE LOOP COUNT = 0
1488                                ;*  : ENDDO
1489                                ;*  : SINGLE STEP THE READ PATH
1490                                ;*  : IF EXPECTED NOT EQUAL
1491                                ;*  : :  THEN-ECC ERROR
1492                                ;*  : :  ELSE-CONTINUE
1493                                ;*  : ENDIF
1494                                ;*  : INPUT THE ECC STATUS
1495                                ;*  : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
1496                                ;*  : IF THE RESULTING STATUS=SINGLE TRACK ERROR CORRECT + POINTER ERROR
1497                                ;*  : :  THEN-CONTINUE
1498                                ;*  : :  ELSE-ERROR
1499                                ;*  : ENDIF
1500                                ;*  : INCREMENT THE INPUT DATA
1501                                ;*  : DO UNTIL INPUT DATA = 0
1502                                ;*  ENDDO
1503                                ;*ENDTST
1504  4300                          SE
 (1)                                ; ***************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1505                                ;*ECC3 MICRO TEST 01
1506                                ;*ECC3 MICRO ERROR 01
1507                                ;*ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
1508                                ;*M8951, M8950'S
1509                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1510                                ;*FATAL ERROR - TEST ABORTED
1511                                ;*
1512                                ;*ECC3 MICRO TEST 01
1513                                ;*ECC3 MICRO ERROR 02
1514                                ;*ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
1515                                ;*M8951, M8950'S
1516                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
1517                                ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1518                                ;*'BYTE/SCLK COUNT NUMBER = LLL''
1519                                ;*FATAL ERROR - MICRO TEST ABORTED
1520                                ;*
1521                                ;*ECC3 MICRO TEST 01
1522                                ;*ECC3 MICRO ERROR 03
1523                                ;*ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
1524                                ;*M8951, M8950'S
1525                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
1526                                ;*SENT TO THE ECC
1527                                ;*'BYTE/SCLK COUNT NUMBER = LLL''
1528                                ;*FATAL ERROR - MICRO TEST ABORTED
1529                                ;*
1530                                ;*ECC3 MICRO TEST 01
1531                                ;*ECC3 MICRO ERROR 04
1532                                ;*ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
1533                                ;*M8951, M8950'S
1534                                ;*CORRECTED DATA FROM THE ECC-INCORRECT
1535                                ;*'BYTE/SCLK COUNT NUMBER = LLL''
```

```
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*ECC3 MICRO TEST 01
                                        ;*ECC3 MICRO ERROR 05
                                        ;*ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
                                        ;*'BYTE/SCLK COUNT NUMBER = LLL"
                                        ;*
                                        ;*ECC3 MICRO TEST 01
                                        ;*ECC3 MICRO ERROR 06
                                        ;*ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA PARITY BIT = ONE SHOULD - ZERO
                                        ;*'BYTE/SCLK COUNT NUMBER = LLL"
                                        ;*
                                        ;*ECC3 MICRO TEST 01
                                        ;*ECC3 MICRO ERROR 07
                                        ;*ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
                                        ;*M8951, M8950'S
                                        ;*CORRECTED ECC CHARACTER INCORRECT
                                        ;*'BYTE/SCLK COUNT NUMBER = LLL"
                                        ;*
                                        ;*ECC3 MICRO TEST 01
                                        ;*ECC3 MICRO ERROR 10
                                        ;*ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
                                        ;*M8951, M8950'S
                                        ;*ECC STATUS INCORRECT AFTER A DATA GROUP
                                        ;*'BYTE/SCLK COUNT NUMBER = LLL"
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4300                                    S
 (1)                                    ; ************************************************************************************
                                        TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1)  4300  3E  01            7.0                MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                CALL    TSET            ;SETUP THE TEST
                                        ;%ECC3-ECC CONTROLLER-BAD DATA ONE TRACK INCORRECT POINTER
                                        ;&M8951, M8950'S
                                                REQ     @7,0,0,0,0
 (1)  4305  CD  06  28       18.0                CALL    REQST
 (1)  4308  00                                   .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4309  00  00                               .WORD   0               ;SYSTEM "0" COUNT
 (1)  430B  00  00                               .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                   .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1)  430E  07                                   .BYTE   @7              ;REQUEST CODE
                                                RIN     R12L
 (1)  430F  DB  94           10.0                IN      R12L            ;READ R12L INTO AC
 (1)  4311  7F                4.0                MOV     A,A             ;RETRY LINK
                                                STA     UNITMP
                                                ANA     A
                                                JNZ     UNITSL
```

```
1579  4319                                          ERR     EXIT,DUMMY
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28        18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0001                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                            .BYTE
 (1)  431E  CD  15  28        18.0                DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  40  45        10.0                        JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1580                                    ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1581                                    ;>FATAL ERROR - TEST ABORTED
1582  4324  C3  40  45        10.0                JMP     EXIT
1583
1584  4327  06  00              7.0   UNITSL: MVI     B,0
1585  4329  3A  BC  46         13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1586  432C  E6  01               7.0           ANI     @01             ;PORT 0?
1587  432E  C2  44  43         10.0           JNZ     FOUND           ;YES-GO USE PORT #0
1588  4331  04                   4.0           INR     B               ;NO-UPDATE POINTER TO PORT #1
1589  4332  3A  BC  46         13.0           LDA     UNITMP          ;DID THE USER SPFCIFY TU
1590  4335  E6  02               7.0           ANI     @02             ;PORT 1?
1591  4337  C2  44  43         10.0           JNZ     FOUND           ;YES-GO USE PORT #1
1592  433A  04                   4.0           INR     B               ;NO-UPDATE POINTER TO PORT #2
1593  433B  3A  BC  46         13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1594  433E  E6  04               7.0           ANI     @04             ;PORT 2?
1595  4340  C2  44  43         10.0           JNZ     FOUND           ;YES-GO USE PORT #2
1596  4343  04                   4.0           INR     B               ;NO-ASSUME PORT #3
1597  4344  CD  43  45         18.0   FOUND:  CALL    CLEAR           ;CLEAR ALL TU PORTS
1598  4347  DB  E0              10.0           IN      INTSTA
1599  4349  E6  80               7.0           ANI     BIT7
1600  434B  B0                   4.0           ORA     B
1601  434C  D3  E0              10.0           OUT     MBSEL
1602
1603  434E  3E  10               7.0   TST01X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
1604  4350  D3  F0              10.0           OUT     CLKCTL
1605  4352  3E  10               7.0           MVI     A,W.GCR
1606  4354  D3  D3              10.0           OUT     WMCCTL
1607  4356  3E  04               7.0           MVI     A,P.RPEN
1608  4358  D3  4C              10.0           OUT     PENAB
1609
1610  435A  3E  60               7.0           MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
1611  435C  D3  48              10.0           OUT     PDIAG           ;LCS, LWR AND DPEN
1612  435E  AF                   4.0           XRA     A
1613  435F  D3  44              10.0           OUT     TAMT
1614  4361  D3  D2              10.0           OUT     TRKENA          ;CLEAR ALL TRACKS FROM
1615  4363  D3  0A              10.0           OUT     RTIEB           ;CLEAR THE TIE BUS REGISTER
1616  4365  D3  D2              10.0           OUT     TRKENA
1617  4367  32  55  46         13.0           STA     BAD1            ;CLEAR THE INVALID
1618  436A  32  56  46         13.0           STA     BAD2            ;TRACK TRANSLATIONS
1619  436D  32  57  46         13.0           STA     BAD3            ;POINTERS
1620  4370  32  BA  46         13.0           STA     INDAT           ;CLEAR THE INPUT DATA BYTE
1621  4373                                     ROUT    R05H
 (1)  4373  D3  8B              10.0                   OUT     R05H            ;WRITE AC INTO R05H
 (1)  4375  7F                   4.0                   MOV     A,A             ;RETRY LINK
1622
1623  4376  3E  A8               7.0   TST01C: MVI     A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
```

```
1624                                                                    ;SET TIE BUS JAM AND PLO DISABLE
1625  4378  D3  09          10.0          OUT     RPCTL
1626  437A  D3  08          10.0          OUT     RFIFOL                ;CLOCK THE FIFO'S
1627  437C  3E  0D           7.0          MVI     A,RCLRT               ;ISSUE CLEAR ALL COMMAND
1628  437E  D3  0B          10.0          OUT     RCMD
1629
1630  4380  00               4.0          NOP                           ;WAIT
1631  4381  00               4.0          NOP
1632  4382  00               4.0          NOP
1633  4383  00               4.0          NOP
1634  4384  00               4.0          NOP
1635
1636  4385  3E  A9           7.0          MVI     A,R.PLO1!R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
1637                                                                    ;SET TIE BUS JAM AND PLO DISABLE
1638  4387  D3  09          10.0          OUT     RPCTL
1639  4389  CD  58  46      18.0          CALL    CLECC                 ;CLEAR THE ECC CHARACTER
1640  438C  16  07           7.0          MVI     D,@7                  ;SET UP THE LOOP COUNT
1641  438E  3A  BA  46       13.0  TST01F: LDA     INDAT                 ;GET THE INPUT DATA
1642  4391  CD  5D  46       18.0          CALL    ECC                   ;CALCULATE ECC FOR
1643  4394  15               4.0          DCR     D                     ;THE 7 INPUT BYTES
1644  4395  C2  8E  43      10.0          JNZ     TST01F
1645
1646  4398  0E  03           7.0          MVI     C,@3                  ;INIT THE LOOP COUNT
1647  439A  3A  BA  46       13.0  TST01I: LDA     INDAT                 ;GET THE INPUT DATA
1648  439D                                ROUT    RO5L
 (1)  439D  D3  8A          10.0                  OUT     RO5L                  ;WRITE AC INTO RO5L
 (1)  439F  7F               4.0                  MOV     A,A                   ;RETRY LINK
1649  43A0  32  45  46       13.0          STA     TRNIN                 ;FILL THE TRANSLATOR
1650  43A3  32  46  46       13.0          STA     TRNIN+1               ;SUBROUTINE INPUT
1651  43A6  32  47  46       13.0          STA     TRNIN+2               ;BUFFER
1652  43A9  32  48  46       13.0          STA     TRNIN+3               ;
1653  43AC  CD  93  46       18.0          CALL    TRK
1654  43AF  CD  6E  45       18.0          CALL    BADT4X5               ;TRANSLATE THE SUBGROUP
1655
1656  43B2  21  49  46       10.0          LXI     H,TRNOUT              ;GET POINTER TO TRANSLATE DATA TABLE
1657  43B5  06  05           7.0          MVI     B,@5                  ;SET UP LOOP COUNT
1658  43B7  7E               7.0  TST01G: MOV     A,M                   ;GET A DATA BYTE
1659  43B8  D3  40          10.0          OUT     TCMD                  ;STORE DATA IN COMMAND ADDRESS
1660  43BA  23               6.0          INX     H                     ;POINT TO DATA PARITY
1661  43BB  7E               7.0          MOV     A,M                   ;GET THE DATA PARITY
1662
1663  43BC  07               4.0          RLC                           ;POSITION FOR OUTPUT
1664  43BD  F6  60           7.0          ORI     P.LWR!P.LCS           ;OR IN CONTROL BITS
1665  43BF  D3  48          10.0          OUT     PDIAG                 ;OUTPUT THE DATA PARITY
1666
1667  43C1  D3  08          10.0          OUT     RFIFOL                ;CLOCK DATA INTO THE FIFO'S
1668
1669  43C3  23               6.0          INX     H                     ;UPDATE THE TABLE POINTER
1670  43C4  05               4.0          DCR     B                     ;DECREMENT LOOP COUNT
1671  43C5  C2  B7  43      10.0          JNZ     TST01G                ;DO UNTIL LOOP COUNT = 0
1672
1673  43C8  3A  BA  46       13.0          LDA     INDAT                 ;GET THE INPUT DATA
1674  43CB  32  45  46       13.0          STA     TRNIN                 ;FILL THE FIRST THREE
1675  43CE  32  46  46       13.0          STA     TRNIN+1               ;BYTES OF THE TRANSLATOR
```

```
1676  43D1  32  47  46   13.0          STA     TRNIN+2         ;SUBROUTINE WITH THE INPUT DATA
1677  43D4  3A  80  46   13.0          LDA     ECCCHR          ;STORE THE CALCULATED
1678  43D7  32  48  46   13.0          STA     TRNIN+3         ;ECC CHARACTER AS THE LAST CHARACTER
1679  43DA  CD  93  46   18.0          CALL    TRK
1680
1681  43DD  CD  6E  45   18.0          CALL    BADT4X5         ;TRANSLATE THE SECOND SUBGROUP
1682
1683  43E0  21  49  46   10.0          LXI     H,TRNOUT        ;GET THE POINTER TO INPUT DATA TABLE
1684  43E3  06  05        7.0          MVI     B,@5            ;SET UP THE LOOP COUNT
1685  43E5  7E            7.0  TST01H: MOV     A,M             ;GET A DATA BYTE
1686  43E6  D3  40       10.0          OUT     TCMD            ;STORE DATA IN CMD ADDRESS
1687  43E8  23            6.0          INX     H               ;POINT TO DATA PARITY BIT
1688  43E9  7E            7.0          MOV     A,M             ;GET THE DATA PARITY
1689  43EA  07            4.0          RLC                     ;POSITION FOR OUTPUT
1690  43EB  F6  60        7.0          ORI     P.LWR!P.LCS     ;OR IN CONTROL BITS
1691  43ED  D3  48       10.0          OUT     PDIAG           ;OUTPUT THE DATA PARITY
1692  43EF  D3  08       10.0          OUT     RFIFOL          ;CLOCK DATA INTO THE FIFOS
1693  43F1  23            6.0          INX     H               ;UPDATE THE TABLE POINTER
1694  43F2  05            4.0          DCR     B               ;DECREMENT THE LOOP COUNT
1695  43F3  C2  E5  43   10.0          JNZ     TST01H          ;DO UNTIL LOOP COUNT=0
1696
1697  43F6  0D            4.0          DCR     C               ;PUT THE GROUP IN
1698  43F7  C2  9A  43   10.0          JNZ     TST01I          ;THE FIFO TWICE
1699
1700  43FA  3E  0B        7.0          MVI     A,DIARD         ;LOAD THE DIAGNOSTIC READ
1701  43FC  D3  0B       10.0          OUT     RCMD            ;COMMAND
1702  43FE  11  01  00   10.0          LXI     D,1             ;SET WATCH DOG INCREMENT
1703  4401  21  A8  FD   10.0          LXI     H,-600          ;SET WATCH DOG COUNT TO 600
1704  4404  D3  0C       10.0  2$:     OUT     RINST           ;STEP THE READ PATH
1705  4406  DB  01       10.0          IN      RPCHI           ;DATA READY SET?
1706  4408  E6  10        7.0          ANI     R.DRDY
1707  440A  C2  1F  44   10.0          JNZ     TST01A          ;YES-GO PROCESS
1708  440D  19           10.0          DAD     D               ;WATCH DOG TIMEOUT?
1709  440E  D2  04  44   10.0          JNC     2$              ;NO-CONTINUE
1710  4411                              ERR     TST01C,DUMMU,1
 (1)                                  ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4411  CD  09  28   18.0                   CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4414  02                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4415  01                                 .BYTE   1
 (1)  4416  CD  15  28   18.0          DUMMU:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4419  DA  76  43   10.0                  JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1711                                ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
1712                                ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1713                                ;<FATAL ERROR - MICRO TEST ABORTED
1714  441C  C3  40  45   10.0          JMP     EXIT
1715
1716  441F  D3  0C       10.0  TST01A: OUT     RINST
1717  4421  06  00        7.0          MVI     B,$0            ;INIT THE BIT TIME COUNT (LOOP COUNT)
1718  4423  00            4.0  TST01B: NOP
1719  4424  3A  B9  46   13.0          LDA     TIETIM          ;GET THE TIME FOR TIE BUS JAMMING (INCORRECT POINTER)
1720  4427  B8            4.0          CMP     B               ;IS IT TIME TO JAM THE TIE BUS?
1721  4428  C2  31  44   10.0          JNZ     4$              ;NO - CONTINUE
1722  442B  CD  19  46   18.0          CALL    POINTER         ;YES - GO GET A REAL POINTER
```

```
1723  442E  C3  3.  44      10.0          JMP     5$              ;
1724  4431  AF               4.0    4$:    XRA     A               ;IF NOT THE BAD TRACK, PROVIDE A GOOD POINTER
1725  4432  D3  0A           10.0          OUT     RTIEB           ;LOAD THE POINTER
1726  4434  D3  0C           10.0    5$:   OUT     RINST           ;STEP CLOCK
1727  4436  04               4.0           INR     B               ;DECREMENT THE LOOP COUNT
1728  4437  78               4.0           MOV     A,B
1729  4438  FE  09           7.0           CPI     $9              ;DONE
1730  443A  C2  23  44        10.0          JNZ     TST01B          ;DO UNTIL LOOP COUNT = ZERO
1731  443D  AF               4.0           XRA     A
1732  443E  D3  0A           10.0          OUT     RTIEB           ;CLEAR THE TIE BUS REGISTER
1733  4440  D3  0C           10.0          OUT     RINST           ;CLOCK THE READ PATH
1734  4442  D3  0C           10.0          OUT     RINST           ;CLOCK THE READ PATH
1735  4444  11  01  0C       10.0          LXI     D,1             ;SET WATCHDOG TIMER INCREMENT
1736  4447  21  A8  FD       10.0          LXI     H,-600          ;SET WATCHDOG COUNT TO 600
1737  444A  D3  0C           10.0    3$:   OUT     RINST           ;SINGLE STEP THE READ PATH
1738  444C  DB  01           10.0          IN      RPCHI           ;DATA READY SET?
1739  444E  E6  10           7.0           ANI     R.DRDY
1740  4450  C2  65  44        10.0          JNZ     TST01K          ;YES-GO PROCESS
1741  4453  19               10.0          DAD     D               ;NO-WATCHDOG TIMEOUT?
1742  4454  D2  4A  44        10.0          JNC     3$              ;NO-CONTINUE
1743  4457                                 ERR     TST01C,DUMMV,1
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4457  CD  09  28        18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0003                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  445A  03                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  445B  01                          .BYTE   1
 (1)  445C  CD  15  28        18.0    DUMMV:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  445F  DA  76  43        10.0            JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1744                          ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
1745                          ;>SENT TO THE ECC
1746                          ;<FATAL ERROR - MICRO TEST ABORTED
1747  4462  C3  40  45        10.0          JMP     EXIT
1748
1749  4465  16  01             7.0    TST01K: MVI     D,@1            ;INIT THE LOOP COUNT
1750  4467  AF               4.0    TST01Y: XRA     A
1751  4468                                 ROUT    ROSH
 (1)  4468  D3  8B           10.0            OUT     ROSH            ;WRITE AC INTO ROSH
 (1)  446A  7F               4.0             MOV     A,A             ;RETRY LINK
1752  446B  3A  BA  46        13.0          LDA     INDAT           ;GET THE INPUT DATA
1753  446E                                 ROUT    EDATA
 (1)  446E  D3  95           10.0            OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4470  7F               4.0             MOV     A,A             ;RETRY LINK
1754  4471  A7               4.0           ANA     A               ;SET CONDITION BITS
1755  4472  3E  00             7.0           MVI     A,0             ;CLEAR THE ACC.
1756  4474  E2  78  44        10.0          JPO     1$              ;ODD # OF ONES - CONTINUE
1757  4477  3C               4.0           INR     A               ;EVEN # OF ONES - SET PARITY
1758  4478  32  BB  46        13.0    1$:   STA     INDATP          ;STORE THE PARITY BIT
1759  447B  D3  0C           10.0    TST01L: OUT     RINST           ;SINGLE STEP THE READ LOGIC
1760  447D  3A  BA  46        13.0          LDA     INDAT
1761  4480  47               4.0           MOV     B,A
1762  4481  DB  19           10.0          IN      ECCCOR
1763  4483  B8               4.0           CMP     B
1764  4484  CA  93  44        10.0          JZ      TST01M
1765  4487                                 ROUT    ADATA
```

```
 (1)  4487  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4489  7F               4.0                MOV     A,A             ;RETRY LINK
1766  448A  78               4.0          MOV    A,B             ;GET THE BYTE COUNT
1767  448B                                ROUT   R05L            ;STORE IN CAS
 (1)  448B  D3  8A          10.0                OUT     R05L            ;WRITE AC INTO R05L
 (1)  448D  7F               4.0                MOV     A,A             ;RETRY LINK
1768  448E                                ERRB   TST01C,TST01M,@1
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  448E  CD  12  28      18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0004                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4491  04                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4492  01                                          .BYTE   @1              ;PRINT ROUTINE NUMBER
 (1)  4493  CD  15  28      18.0                TST01M::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4496  DA  76  43      10.0                        JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1769                                ;>CORRECTED DATA FROM THE ECC-INCORRECT
1770  4499  DB  15          10.0          IN     RPSTA           ;GET THE ECC PARITY BIT
1771  449B  E6  80           7.0          ANI    E.CDP           ;GET THE CORRECTED PARITY BIT
1772  449D  07               4.0          RLC                    ;POSITION BIT FOR COMPARE
1773  449E  4F               4.0          MOV    C A             ;SAVE IN REG. C
1774  449F  3A  BB  46      13.0          LDA    INDATP          ;GET THE EXPECTED PARITY BIT
1775  44A2  A7               4.0          ANA    A               ;SET THE CONDITION BITS
1776  44A3  CA  BE  44      10.0          JZ     TST01QO         ;GO CHECK FOR PARITY=0
1777  44A6  79               4.0          MOV    A,C             ;ELSE CHECK FOR PARITY=1
1778  44A7  FE  01           7.0          CPI    @1
1779  44A9  CA  B5  44      10.0          JZ     TST01RG
1780  44AC  78               4.0          MOV    A,B
1781  44AD                                ROUT   R05L
 (1)  44AD  D3  8A          10.0                OUT     R05L            ;WRITE AC INTO R05L
 (1)  44AF  7F               4.0                MOV     A,A             ;RETRY LINK
1782  44B0                                ERR    TST01C,TST01RG,@1
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44B0  CD  09  28      18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)      0005                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44B3  05                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44B4  01                                          .BYTE   @1
 (1)  44B5  CD  15  28      18.0                TST01RG::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44B8  DA  76  43      10.0                        JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1783                                ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
1784                                ;<
1785  44BB  C3  D2  44      10.0          JMP    TST01SZ         ;CONTINUE WITH TEST
1786  44BE  79               4.0    TST01QO: MOV   A,C             ;CHECK FOR PARITY = ZERO
1787  44BF  A7               4.0          ANA    A               ;SET CONDITION BITS
1788  44C0  CA  CC  44      10.0          JZ     TST01TP         ;CONTINUE IF ZERO
1789  44C3  78               4.0          MOV    A,B
1790  44C4                                ROUT   R05L
 (1)  44C4  D3  8A          10.0                OUT     R05L            ;WRITE AC INTO R05L
 (1)  44C6  7F               4.0                MOV     A,A             ;RETRY LINK
```

```
1792  44C7                                    ERR     TST01C,TST01TP,@1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44C7  CD  09  28        18.0                     CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0006                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44CA  06                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44CB  01                                         .BYTE   @1
 (1)  44CC  CD  15  28        18.0               TST01TP::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44CF  DA  76  43        10.0                     JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1793                                    ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
1794                                    ;<
1795  44D2  7A                 4.0      TST01SZ: MOV    A,D             ;GET THE CHARACTER COUNT
1796  44D3  3C                 4.0               INR     A               ;INCREMENT
1797  44D4  FE  08             7.0               CPI     $8              ;DONE?
1798  44D6  CA  DD  44        10.0               JZ      TST01N          ;YES-GO CHECK ECC
1799  44D9  57                 4.0               MOV     D,A             ;NO-CONTINUE
1800  44DA  C3  67  44        10.0               JMP     TST01Y
1801
1802  44DD  D3  0C            10.0      TST01N: OUT     RINST           ;SINGLE STEP THE READ PATH
1803  44DF  3A  80  46        13.0               LDA     ECCCHR          ;GET THE EXPECTED ECC CHARACTER
1804  44E2                               ROUT    EDATA           ;SAVE IN CAS
 (1)  44E2  D3  95            10.0                     OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44E4  7F                 4.0                     MOV     A,A             ;RETRY LINK
1805  44E5  47                 4.0               MOV     B,A             ;SAVE IN REGISTER B
1806  44E6  DB  19            10.0               IN      ECCCOR
1807  44E8                               ROUT    ADATA
 (1)  44E8  D3  94            10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44EA  7F                 4.0                     MOV     A,A             ;RETRY LINK
1808  44EB  B8                 4.0               CMP     B
1809  44EC  CA  FA  44        10.0               JZ      TST01U
1810  44EF  3A  BA  46        13.0               LDA     INDAT           ;GET THE DATA/COUNT
1811  44F2                               ROUT    R05L            ;SAVE IN THE CAS
 (1)  44F2  D3  8A            10.0                     OUT     R05L            ;WRITE AC INTO R05L
 (1)  44F4  7F                 4.0                     MOV     A,A             ;RETRY LINK
1812  44F5                               ERR     TST01C,TST01U,1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44F5  CD  09  28        18.0                     CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0007                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44F8  07                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44F9  01                                         .BYTE   1
 (1)  44FA  CD  15  28        18.0               TST01U::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44FD  DA  76  43        10.0                     JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1813                                    ;>CORRECTED ECC CHARACTER INCORRECT
1814                                    ;<
```

```
1816  4500  DB  1A          10.0          IN      ECCSTA          ;GET THE ECC STATUS
1817  4502  E6  0F           7.0          ANI     @17             ;REMOVE UNWANTED BITS
1818  4504  FE  09           7.0          CPI     E.STEC!E.PNTR
1819  4506  CA  1C  45      10.0          JZ      TST01V
1820  4509                               ROUT    ADATA
 (1)  4509  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  450B  7F               4.0                  MOV     A,A             ;RETRY LINK
1821  450C  3E  09           7.0          MVI     A,E.STEC!E.PNTR
1822  450E                               ROUT    EDATA
 (1)  450E  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4510  7F               4.0                  MOV     A,A             ;RETRY LINK
1823  4511  3A  BA  46      13.0          LDA     INDAT           ;GET THE DATA/COUNT
1824  4514                               ROUT    R05L            ;SAVE IN THE CAS
 (1)  4514  D3  8A          10.0                  OUT     R05L            ;WRITE AC INTO R05L
 (1)  4516  7F               4.0                  MOV     A,A             ;RETRY LINK
1825  4517                               ERRB    TST01C,TST01V,1
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4517  CD  12  28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0008                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  451A  08                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  451B  01                                 .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  451C  CD  15  28      18.0          TST01V::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  451F  DA  76  43      10.0                  JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1826                                   ;>ECC STATUS INCORRECT AFTER A DATA GROUP
1827                                   :<
1828  4522  3A  BA  46      13.0          LDA     INDAT
1829  4525  3C               4.0          INR     A
1830  4526  32  BA  46      13.0          STA     INDAT
1831  4529  C2  76  43      10.0          JNZ     TST01C
1832  452C                               ENDTST  TST01X
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  452C                             REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  452C  CD  06  28      18.0                          CALL    REQST
 (2)  452F  00                                         .BYTE                   ;DATA PATTERN NUMBER
 (2)  4530  00  00                                     .WORD                   ;SYSTEM '"' COUNT
 (2)  4532  00  00                                     .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4534  00                                         .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4535  07                                         .BYTE   7               ;REQUEST CODE
 (1)  4536  3A  9A  4F      13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4539  3D               4.0          DCR     A               ;DOWNCOUNT
 (1)  453A  32  9A  4F      13.0          STA     ITERA           ;SAVE COUNT
 (1)  453D  F2  4E  43      10.0          JP      TST01X          ;DO TEST UNTIL TILL = 0
1833  4540  C3  18  28      10.0  EXIT:   JMP     TSTEND
```

```
                                .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
4543                            SSUB
 (1)                            ; ************************************************************************
 (1)                            ;*SUBROUTINE TITLE
 (1)                            ;*---------- -----
                                ;*CLEAR ALL TU PORTS
4543                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                ;*AND LOOP MODES.
4543                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNSUB
                                ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                ;*    CLEAR PORT SELECT FOR TRANSPORT
                                ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                ;*    CLEAR PORT AMTIE WORD
                                ;*ENDSUB
4543                            S
 (1)                            ; ************************************************************************
4543  F5              12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
4544  C5              12.0          PUSH    B
4545  06  00           7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
4547  DB  E0          10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
4549  E6  80           7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
454B  B0               4.0          ORA     B               ;ADD IN THE SELECTED PORT #
454C  D3  E0          10.0          OUT     MBSEL           ;RESET TO THIS PORT
454E  3E  80           7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
4550  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
4552  AF               4.0          XRA     A               ;CLEAR TU COMMAND A
4553  D3  40          10.0          OUT     TCMD            ;
4555  3E  81           7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
4557  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
4559  3E  00           7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
455B  D3  40          10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
455D  AF               4.0          XRA     A
455E  D3  44          10.0          OUT     TAMT            ;CLEAR AMTIE WORD
4560  D3  48          10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
4562  D3  4C          10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
4564  04               4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
4565  78               4.0          MOV     A,B
4566  FE  04           7.0          CPI     4               ;DONE?
4568  C2  47  45      10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
456B  C1              10.0          POP     B               ;RESET B & C
456C  F1              10.0          POP     PSW             ;ALL DONE
456D  C9              10.0          RET                     ;EXIT
```

```
                                .SBTTL  SUBROUTINE 4X5 TRANSLATE A SUBGROUP
456E                            SSUB
 (1)                            ; ***************************************************************************
 (1)                            ;*SUBROUTINE TITLE
 (1)                            ;*---------- -----
                                ;*4X5 TRANSLATE A SUBGROUP
456E                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS SUBROUTINE IS USED TO 4X5 TRANSLATE A 4 BYTE DATA SUBGROUP INTO
                                ;*A 5 BYTE SUBGROUP AS RECORDED ON TAPE OR READ FROM TAPE, AT THE ENTRY
                                ;*T4X5.  AT ENTRY BADT4X5 THE VARIABLES BAD 1, 2, 3 CAN CONTAIN NUMBERS
                                ;*FROM 1-9 TO INDICATE TRACKS THAT WILL BE TRANSLATED TO A VALID VALUE,
                                ;*BUT NOT CORRECT FOR THE INPUT DATA.
456E                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNSUB
                                ;*    CLEAR THE BAD TRANSLATION POINTERS
                                ;*    GET THE 4 (INPUT) DATA BYTES TO BE TRANSLATED
                                ;*    CLEAR THE TRANSLATED (OUTPUT) BUFFER (OUTPUT BUFFER IS 9X5)
                                ;*    SET LOOP COUNT TO 1
                                ;*    BGNDO
                                ;*    :    CALCULATE ODD PARITY FOR THE FOUR (INPUT) DATA BYTES (INPUT BUFFER IS NOW 9X4
                                ;*    :    ROTATE EACH (5) ENTRY OF THE TRANSLATED (OUTPUT) BUFFER RIGHT
                                ;*    :    GET THE LEAST SIGNIFICANT BIT OF EACH (INPUT) DATA BYTE AND
                                ;*    :        FORM A TABLE INDEX (MS BIT OF INDEX = LS BIT OF THE FIRST
                                ;*    :        BYTE OF THE INPUT DATA, ETC.)
                                ;*    :    REMOVE THE LS BIT OF EACH INPUT DATA BYTE (THE ONE JUST USED
                                ;*    :        TO BUILD THE INDEX)
                                ;*    :    USE THE 4 BIT INDEX TO GET THE TRANSLATION VALUE
                                ;*    :    PUT ONE BIT OF THE TRANSLATION VALUE, AS THE MOST SIGNIFICANT
                                ;*    :        BIT EACH OF THE 5 TRANSLATED DATA BYTES (LS BIT OF THE
                                ;*    :        TRANSLATION VALUE REPLACES MS BIT OF THE FIRST TRANSLATED
                                ;*    :        DATA BYTE.
                                ;*    :    INCREMENT THE LOOP COUNT
                                ;*    :    DO UNTIL THE LOOP COUNT = 10
                                ;*    ENDDO
                                ;*ENDSUB
456E                            S
 (1)                            ; ***************************************************************************

456E  F5              12.0  BADT4X5: PUSH   PSW             ;SAVE REGISTER A + PSW
456F  C3  7D  45      10.0          JMP    BAD
4572  F5              12.0  T4X5:    PUSH   PSW             ;SAVE REGISTER A + PSW
4573  AF               4.0          XRA    A               ;CLEAR A
4574  32  55  46      13.0          STA    BAD1            ;CLEAR BAD TRANSLATION 1
4577  32  56  46      13.0          STA    BAD2            ;CLEAR BAD TRANSLATION 2
457A  32  57  46      13.0          STA    BAD3            ;CLEAR BAD TRANSLATION 3
457D  C5              12.0  BAD:     PUSH   B               ;SAVE REGISTER B + C
457E  D5              12.0          PUSH   D               ;SAVE REGISTER D + E
457F  E5              12.0          PUSH   H               ;SAVE REGISTER H + L
```

```
1924                                        ;FOUR DATA BYTES ARE NOW IS THE BUFFER TABLE
1925                                        ;CLEAR THE TRANSLATED DATA TABLE.
1926  4580  06  0A          7.0           MVI     B,10            ;SET UP LOOP COUNT
1927  4582  11  49  46      10.0          LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
1928  4585  AF              4.0           XRA     A               ;CLEAR A
1929  4586  12              7.0   D4X5:   STAX    D               ;STORE IN THE TABLE
1930  4587  13              6.0           INX     D               ;UPDATE TABLE POINTER
1931  4588  05              4.0           DCR     B               ;DECREMENT LOOP COUNT
1932  4589  C2  86  45      10.0          JNZ     D4X5            ;DO UNTIL LOOP COUNT=0
1933  458C  0E  01          7.0           MVI     C,1             ;SET UP TRACK COUNT
1934  458E  06  04          7.0   B4X5:   MVI     B,4             ;SET UP BIT COUNT
1935  4590  11  45  46      10.0          LXI     D,TRNIN
1936  4593  AF              4.0           XRA     A               ;CLEAR THE GROUP POSITION COUNT
1937  4594  32  53  46      13.0          STA     GP4X5
1938  4597  1A              7.0   C4X5:   LDAX    D               ;GET A DATA BYTE
1939  4598  A7              4.0           ANA     A               ;SET CONDITION BITS
1940  4599  E2  9D  45      10.0          JPO     PO4X5           ;ODD PARITY LEAVE CARRY CLEAR
1941  459C  37              4.0           STC                     ;EVEN PARITY SET CARRY
1942  459D  1F              4.0   PO4X5:  RAR                     ;SHIFT OUT DESIRED BIT
1943  459E  12              7.0           STAX    D               ;STORE RESULT BACK IN TEMP TABLE
1944  459F  3A  53  46      13.0          LDA     GP4X5           ;GET THE GROUP POSITION BYTE
1945  45A2  17              4.0           RAL                     ;PUT IN THIS DATA BIT
1946  45A3  32  53  46      13.0          STA     GP4X5           ;SAVE THE UPDATED GROUP POSITION BYTE
1947  45A6  13              6.0           INX     D               ;UPDATE THE TABLE POINTER
1948  45A7  05              4.0           DCR     B               ;DECREMENT THE BIT COUNT
1949  45A8  C2  97  45      10.0          JNZ     C4X5            ;DO UNTIL ALL 4 BITS ARE RECEIVED
1950
1951                                        ;AT THIS POINT GP4X5 CONTAINS 4 BITS - ONE FROM EACH OF THE FOUR BYTES
1952
1953                                        ;PERFORM A DOUBLE PRECISION SHIFT RIGHT ON THE TRANSLATED DATA
1954  45AB  06  05          7.0           MVI     B,5             ;SET UP LOOP COUNT
1955  45AD  11  49  46      10.0          LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
1956  45B0  13              6.0   E4X5:   INX     D               ;POINT TO PARITY BIT
1957  45B1  1A              7.0           LDAX    D               ;GET PARITY BIT
1958  45B2  A7              4.0           ANA     A               ;CLEAR THE CARRY BIT
1959  45B3  1F              4.0           RAR                     ;SHIFT TO THE CARRY BIT
1960  45B4  12              7.0           STAX    D               ;STORE IT BACK
1961  45B5  1B              6.0           DCX     D               ;DECREMENT TO DATA BITS
1962  45B6  1A              7.0           LDAX    D               ;GET THE DATA BITS
1963  45B7  1F              4.0           RAR                     ;SHIFT IN CARRY BITS
1964  45B8  12              7.0           STAX    D               ;STORE BACK IN TABLE
1965  45B9  13              6.0           INX     D               ;POINT TO NEXT TABLE ENTRY
1966  45BA  13              6.0           INX     D               ;
1967  45BB  05              4.0           DCR     B               ;DECREMENT LOOP COUNT
1968  45BC  C2  B0  45      10.0          JNZ     E4X5            ;DO UNTIL LOOP COUNT=0
1969  45BF  3A  55  46      13.0          LDA     BAD1            ;IS A BAD POINTER SPECIFIED?
1970  45C2  A7              4.0           ANA     A               ;SET CONDITION BITS
1971  45C3  CA  CA  45      10.0          JZ      BAD1C           ;NO-CONTINUE
1972  45C6  B9              4.0           CMP     C               ;YES-IS IT THIS DATA POSITION?
1973  45C7  CA  E0  45      10.0          JZ      INV4X5          ;YES-GO PERFORM INCORRECT TRANSLATION
1974  45CA  3A  56  46      13.0  BAD1C:  LDA     BAD2            ;IS A BAD POINTER SPECIFIED
1975  45CD  A7              4.0           ANA     A               ;SET CONDITION BITS
1976  45CE  CA  D5  45      10.0          JZ      BAD2C           ;NO-CONTINUE
1977  45D1  B9              4.0           CMP     C               ;YES-IS IT THIS DATA POSITION
```

```
1978  45D2  CA  E0  45   10.0          JZ      INV4X5          ;YES-GO PERFORM INCORRECT TRANSLATION
1979  45D5  3A  57  46   13.0  BAD2C:  LDA     BAD3            ;IS A BAD POINTER SPECIFIED
1980  45D8  A7                4.0          ANA     A               ;SET CONDITION BITS
1981  45D9  CA  E7  45   10.0          JZ      BAD3C           ;NO-PERFORM PROPER TRANSLATION
1982  45DC  B9                4.0          CMP     C               ;YES-IS IT THIS DATA POSITION
1983  45DD  C2  E7  45   10.0          JNZ     BAD3C           ;NO-GO PERFORM PROPER TRANSLATION
1984  45E0  3A  53  46   13.0  INV4X5: LDA     GP4X5           ;GET THE GROUP POSITION COUNT
1985  45E3  3C                4.0          INR     A               ;INCREMENT-TO INCORRECT TRANSLATION
1986  45E4  32  53  46   13.0          STA     GP4X5           ;STORE IT
1987  45E7  11  35  46   10.0  BAD3C:  LXI     D,TAB4X5        ;LOAD ADDRESS OF TRANSLATION TABLE
1988  45EA  26  00             7.0          MVI     H,0             ;CLEAR REGISTER H
1989  45EC  3A  53  46   13.0          LDA#    GP4X5           ;GET GROUP POSITION COUNT
1990  45EF  E6  0F             7.0          ANI     $0F
1991  45F1  6F                4.0          MOV     L,A
1992  45F2  19                10.0         DAD     D               ;ADD GROUP POSITION COUNT TO TABLE
1993  45F3  7E                7.0          MOV     A,M             ;GET TRANSLATED DATA
1994  45F4  32  54  46   13.0          STA     TRNTMP
1995                                ;"TRNTMP" = THE TRANSLATED DATA
1996
1997  45F7  11  49  46   10.0          LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
1998  45FA  06  05             7.0          MVI     B,5             ;SET UP LOOP COUNT
1999  45FC  3A  54  46   13.0  F4X5:   LDA     TRNTMP          ;GET TRANSLATED DATA
2000  45FF  13                6.0          INX     D               ;POINT TO PARITY POSITION
2001  4600  A7                4.0          ANA     A               ;CLEAR CARRY BIT
2002  4601  1F                4.0          RAR                     ;SHIFT OUT A BIT
2003  4602  32  54  46   13.0          STA     TRNTMP          ;STORE TRANSLATED DATA
2004  4605  1A                7.0          LDAX    D               ;GET THE PARITY BIT (ALWAYS ZERO)
2005  4606  17                4.0          RAL                     ;ROLL IN THE CARRY BIT
2006  4607  12                7.0          STAX    D               ;STORE AS THE PARITY BIT
2007  4608  13                6.0          INX     D               ;UPDATE TABLE POINTER
2008  4609  05                4.0          DCR     B               ;DECREMENT THE LOOP COUNT
2009  460A  C2  FC  45   10.0          JNZ     F4X5            ;DO UNTIL LOOP COUNT=0
2010
2011                          ;NOW THE TRANSLATED DATA TABLE HAS BEEN UPDATED
2012
2013  460D  0C                4.0          INR     C               ;DECREMENT THE TRACK COUNT
2014  460E  79                4.0          MOV     A,C
2015  460F  FE  0A             7.0          CPI     10
2016  4611  C2  8E  45   10.0          JNZ     B4X5            ;DO UNTIL ALL TRACKS TRANSLATED
2017  4614  E1                10.0         POP     H               ;RESTORE REGISTER H + L
2018  4615  D1                10.0         POP     D               ;RESTORE REGISTER D + E
2019  4616  C1                10.0         POP     B               ;RESTORE REGISTER B + C
2020  4617  F1                10.0         POP     PSW             ;RESTORE REGISTER A + PSW
2021  4618  C9                10.0         RET                     ;RETURN TO USER
2022
```

```
                                .SBTTL  SUBROUTINE CALCULATE POINTER
                                ;*THE POINTERS PROVIDED FOR THE TRACK IN ERROR ARE ALSO DERIVED FROM THE
                                ;*INPUT/EXPECTED DATA AS PER THE FOLLOWING TABLE.
                                ;*
                                ;*      INPUT/EXPECTED
                                ;*          DATA                        POINTER GENERATED
                                ;*      --------------                  -----------------
                                ;*
                                ;*         XX00XXXX                     HISTORY
                                ;*         XX01XXXX                     PHASE TIE
                                ;*         XX10XXXX                     AMTIE
                                ;*         XX11XXXX                     ILLEGAL 5X4
                                ;*
                                ;*      X = DON'T CARE BITS
4619                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;
4619  F5              12.0  POINTER: PUSH   PSW             ;SAVE THE ACCUMULATOR
461A  C5              12.0          PUSH   B               ;SAVE B + C
461B  3A  BA  46      13.0          LDA    INDAT           ;GET INPUT DATA
461E  E6  30           7.0          ANI    $30             ;REMOVE THE WORKING BITS
4620  07               4.0          RLC                    ;POSITION THE DATA
4621  07               4.0          RLC
4622  07               4.0          RLC
4623  07               4.0          RLC
4624  21  31  46      10.0          LXI    H,POINTBL       ;GET BASE ADDRESS OF TABLE
4627  4F               4.0          MOV    C,A
4628  06  00           7.0          MVI    B,0
462A  09              10.0          DAD    B
462B  7E               7.0          MOV    A,M             ;GET THE ECC POINTER
462C  D3  0A          10.0          OUT    RTIEB           ;SET THE TIE BUS VALUE
462E  C1              10.0          POP    B               ;RESTORE B + C
462F  F1              10.0          POP    PSW             ;RESTORE ACCUMULATOR
4630  C9              10.0          RET
                                ;
4631  01                    POINTBL: .BYTE  1
4632  02                             .BYTE  2
4633  04                             .BYTE  4
4634  08                             .BYTE  8
                                ;
```

```
2064                                .SBTTL  TABLE 4 X 5 TRANSLATION
2065
2066                                ;THIS TABLE TRANSLATED DATA FOR AS FOLLOWS
2067                                ;  INPUT GROUP POSITIONS              OUTPUT GROUP POSITIONS
2068                                ;   1 2 3 4 / 5 6 7 8                 10 9 8 7 6 / 5 4 3 2 1
2069                                ;
2070                                ;       0000                              10011
2071                                ;       0001                              11011
2072                                ;       0010                              01001
2073                                ;       0011                              11001
2074                                ;       0100                              10111
2075                                ;       0101                              10101
2076                                ;       0110                              01101
2077                                ;       0111                              11101
2078                                ;       1000                              01011
2079                                ;       1001                              10010
2080                                ;       1010                              01010
2081                                ;       1011                              11010
2082                                ;       1100                              01111
2083                                ;       1101                              10110
2084                                ;       1110                              01110
2085                                ;       1111                              11110
2086
2087  4635  13                      TAB4X5: .BYTE   @23
2088  4636  1B                              .BYTE   @33
2089  4637  09                              .BYTE   @11
2090  4638  19                              .BYTE   @31
2091  4639  17                              .BYTE   @27
2092  463A  15                              .BYTE   @25
2093  463B  0D                              .BYTE   @15
2094  463C  1D                              .BYTE   @35
2095  463D  0B                              .BYTE   @13
2096  463E  12                              .BYTE   @22
2097  463F  0A                              .BYTE   @12
2098  4640  1A                              .BYTE   @32
2099  4641  0F                              .BYTE   @17
2100  4642  16                              .BYTE   @26
2101  4643  0E                              .BYTE   @16
2102  4644  1E                              .BYTE   @36
2103
```

```
2105                                    .SBTTL  SUBROUTINE VARIABLES
2106
2107  4645 0004                         TRNIN:  .BLKB   4               ;TABLE CONTAINING THE 4 DATA BYTES TO
2108                                                                    ;BE TRANSLATED
2109  4649 000A                         TRNOUT: .BLKB   10              ;TRANSLATED DATA TABLE DATA AFTER 4X5
2110                                                                    ;TRANSLATION
2111  4653   00                         GP4X5:  .BYTE   0               ;ACCUMULATOR FOR THE 4 BIT SUBGROUP TO
2112                                                                    ;BE TRANSLATED
2113  4654   00                         TRNTMP: .BYTE   0               ;TEMP. STORAGE FOR THE TRANSLATED SUBGROUP.
2114  4655   00                         BAD1:   .BYTE   0               ;BAD TRANSLATION POINTER 1
2115  4656   00                         BAD2:   .BYTE   0               ;BAD TRANSLATION POINTER 2
2116  4657   00                         BAD3:   .BYTE   0               ;BAD TRANSLATION POINTER 3
2117
```

```
                                        .SBTTL  SUBROUTINE CLEAR ECC
4658                                    S
 (1)                                    ; ************************************************************************
                                        ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION "ECCCHR" -
                                        ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
                                        ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
4658                                    S
 (1)                                    ; ************************************************************************

4658  AF                       4.0  CLECC:  XRA     A               ;CLEAR THE ACCUMULATOR
4659  32   80   46              13.0         STA     ECCCHR          ;CLEAR THE ECC CHARACTER
465C  C9                       10.0         RET                     ;RETURN TO USER
```

```
                                        .SBTTL  SUBROUTINE CALCUALTE ECC CHARACTER
465D                                    S
 (1)                                    ; ************************************************************************
                                        ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
                                        ;THE CONTENTS OF "ECCCHR" TO UPDATE THE CONTENTS OF "ECCCHR" ACCORDING
                                        ;TO THE ANSI STANDARD ECC POLYNOMIAL.
465D                                    S
 (1)                                    ; ************************************************************************

465D  21  80  46      10.0  ECC:   LXI   H,ECCCHR       ;LOAD ADDRESS OF ECC CHAR.
4660  AE                7.0          XRA   M             ;EXCLUSIVE OR CHAR. AND ECC
4661  5F                4.0          MOV   E,A           ;SAVE XOR RESULT IN E
4662  E6  10            7.0          ANI   $10           ;IS BIT #4 OF RESULT SET
4664  7B                4.0          MOV   A,E           ;RESTORE XOR RESULT FROM B
4665  CA  6A  46       10.0          JZ    ECC1          ;CONTINUE IF BIT #4 RESET
4668  EE  23            7.0          XRI   $23           ;ELSE-XOR WITH 23
466A  5F                4.0  ECC1:   MOV   E,A           ;STORE THE ECC RESULT IN E
                                                          ;
466B  AF                4.0          XRA   A             ;CLEAR A
466C  4F                4.0          MOV   C,A           ;CLEAR THE TRANSLATE RESULT
466D  21  78  46        10.0          LXI   H,ECCTBL      ;POINT TO ECC TABLE TO RE-POSITION
4670  CD  81  46        18.0          CALL  TRANS         ;TRANSLATE THE BITS
4673  79                4.0          MOV   A,C           ;GET THE TRANSLATED RESULT
4674  32  80  46        13.0          STA   ECCCHR        ;STORE RESULT
4677  C9               10.0          RET
                                                          ;
4678  08                      ECCTBL: $08                 ;BIT 0 = POSITION 3
4679  20                              $20                 ;BIT 1 = POSITION 5
467A  02                              $02                 ;BIT 2 = POSITION 1
467B  40                              $40                 ;BIT 3 = POSITION 6
467C  80                              $80                 ;BIT 4 = POSITION 7
467D  01                              $01                 ;BIT 5 = POSITION 0
467E  10                              $10                 ;BIT 6 = POSITION 4
467F  04                              $04                 ;BIT 7 = POSITION 2
                                                          ;
4680  00                      ECCCHR: .BYTE   0
                                                          ;
```

```
2167                                    .SBTTL  SUBROUTINE POLYNOMIAL BIT TRANSLATION
2168
2169                                    ;
2170                                    ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
2171                                    ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
2172                                    ;
2173
2174  4681  06  01          7.0  TRANS:  MVI   B,1          ;INIT 'B' TO BIT POSITION 0
2175  4683  7B              4.0  TRANS1: MOV   A,E          ;GET CHAR TO BE TRANSLATED
2176  4684  A0              4.0          ANA   B            ;SEE IF BIT POSITION IN 'B' IS SET
2177  4685  CA  88  46      10.0          JZ    TRANS2       ;DO NEXT BIT POSITION IF NOT SET
2178  4688  79              4.0          MOV   A,C          ;GET PREVIOUS RESULT OF 'OR'
2179  4689  B6              7.0          ORA   M            ;'OR' IN NEW POSITION
2180  468A  4F              4.0          MOV   C,A          ;SAVE RESULT
2181
2182  468B  78              4.0  TRANS2: MOV   A,B
2183  468C  07              4.0          RLC                ;POSITION MASK TO NEXT BIT
2184  468D  47              4.0          MOV   B,A
2185  468E  D8             12.0          RC                 ;EXIT WHEN ALL POSITIONS DONE
2186  468F  23              6.0          INX   H            ;POINT TO NEXT TABLE ENTRY
2187  4690  C3  83  46      10.0          JMP   TRANS1       ;PROCESS NEXT BIT
2188
```

```
                                .SBTTL  SUBROUTINE TRK
4693                            SSUB
 (1)                            ;*****************************************************************************
 (1)                            ;*SUBROUTINE TITLE
 (1)                            ;*---------- -----
                                ;*GENERATE BAD TRACK TRANSLATION
4693                            SD
 (1)                            ;*****************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS SUBROUTINE GENRATES A TRACK NUMBER FOR THE BADT4X5 SUBROUTINE
                                ;*BASED ON THE VALUE OF INDAT AND AN INCORRECT TIE BUS TIME (THE TIME FOR THE NEXT BIT).
                                ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
                                ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
                                ;*MESSAGE, AND TRACK SIMULATED IN ERROR CAN BE DETERMINED FROM THE
                                ;*FOLLOWING TABLE.
                                ;*
                                ;*      INPUT/EXPECTED          DATA BIT POSITION       BIT POSITION OF
                                ;*          DATA                WITH ECC CORRECTION       INCORRECT POINTER
                                ;*      -------------           -------------------     --------------
                                ;*
                                ;*          XXXX0000                    0                       1
                                ;*          XXXX0001                    1                       2
                                ;*          XXXX0010                    2                       3
                                ;*          XXXX0011                    3                       4
                                ;*          XXXX0100                    4                       5
                                ;*          XXXX0101                    5                       6
                                ;*          XXXX0110                    6                       7
                                ;*          XXXX0111                    7                       P
                                ;*          XXXX1000                    P                       0
                                ;*          XXXX1001                    1                       2
                                ;*          XXXX1010                    2                       3
                                ;*          XXXX1011                    3                       4
                                ;*          XXXX1100                    4                       5
                                ;*          XXXX1101                    5                       6
                                ;*          XXXX1110                    6                       7
                                ;*          XXXX1111                    7                       P
                                ;*
                                ;*      X = DON'T CARE BITS
                                ;*

4693  3A  BA  46      13.0  TRK:   LDA   INDAT           ;GET INPUT DATA
4696  E6  0F           7.0        ANI   $0F             ;REMOVE 4 BITS
4698  FE  08           7.0        CPI   $8              ;PARITY TRACK?
469A  CA  9F  46      10.0        JZ    TRKA            ;YES
469D  E6  07           7.0        ANI   $7              ;NO-IGNORE BIT3
469F  3C               4.0  TRKA: INR   A              ;INCREMENT FOR SUBROUTINE
46A0  32  55  46      13.0        STA   BAD1            ;SAVE IT.
46A3  3D               4.0        DCR   A              ;PREPARE TO USE AS TABLE INDEX
46A4  21  B0  46      10.0        LXI   H,TIMTBL       ;INCORRECT TIE BUS TIME TABLE START
46A7  5F               4.0        MOV   E,A            ;LOW HALF OF TABLE INDEX
46A8  16  00           7.0        MVI   D,0            ;ZERO HIGH HALF
46AA  19              10.0        DAD   D              ;PRODUCE TABLE ENTRY ADDRESS
46AB  7E               7.0        MOV   A,M            ;GET INCORRECT TIE BUS TIME FROM TABLE
```

```
2238  46AC  32  B9  46        13.0          STA     TIETIM          ;SAVE VALUE
2239  46AF  C9                10.0          RET
2240  46B0  03                        TIMTBL: .BYTE   3               ;INCORRECT TIE BUS TIME FOR BIT 0
2241  46B1  00                                .BYTE   0               ;INCORRECT TIE BUS TIME FOR BIT 1
2242  46B2  01                                .BYTE   1               ;INCORRECT TIE BUS TIME FOR BIT 2
2243  46B3  04                                .BYTE   4               ;INCORRECT TIE BUS TIME FOR BIT 3
2244  46B4  08                                .BYTE   8               ;INCORRECT TIE BUS TIME FOR BIT 4
2245  46B5  07                                .BYTE   7               ;INCORRECT TIE BUS TIME FOR BIT 5
2246  46B6  05                                .BYTE   5               ;INCORRECT TIE BUS TIME FOR BIT 6
2247  46B7  06                                .BYTE   6               ;INCORRECT TIE BUS TIME FOR BIT 7
2248  46B8  02                                .BYTE   2               ;INCORRECT TIE BUS TIME FOR BIT P
2249
2250
```

```
                                        .SBTTL  PROGRAM VARIABLES

                                TIETIM: .BYTE   0               ;TIE BUS JAM TIME
                                INDAT:  .BYTE   0               ;DATA BYTE TC ECC
                                INDATP: .BYTE   0               ;DATA BYTE PARITY
                                UNITMP: .BYTE   0               ;UNIT MAP
                                AMTMSK: .WORD   0               ;AMTIE MASK WORD
                                DATAA:  .BYTE   0               ;ACTUAL DATA AFTER TRANSLATION
                                DATAAP: .BYTE   0               ;ACTUAL DATA PARITY AFTER TRANSLATION
                                DATAE:  .BYTE   0               ;EXPECTED DATA AFTER TRANSLATION
                                DATAEP: .BYTE   0               ;EXPECTED DATA PARITY AFTER TRANSLATION
                                        .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
AMTMSK  46BD           ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000          BAD     457D          BADST  = 0090
BADT4X  456E           BAD1    4655           BAD1C   45CA          BAD2    4656
BAD2C   45D5           BAD3    4657           BAD3C   45E7          BIT0   = 0001
BIT1   = 0002          BIT15  = 8000          BIT2   = 0004         BIT3   = 0008
BIT4   = 0010          BIT5   = 0020          BIT6   = 0040         BIT7   = 0080
BIT8   = 0100          BIT9   = 0200          BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60           BRKXCT= 4F00          BSAVE   4F9C          BYTCNT= 00D4
BYTEH   4F24           BYTEL   4F23          B4X5    458E          C      =%0001
CASCT   4F21           CASCTL= 00A0          CASSTA= 00A0          CATTH  = 0089
CATTL  = 0088          CBUSST= 00A1          CBYTH  = 008B         CBYTL  = 008A
CDG1H  = 0087          CDG1L  = 0086         CDG2H  = 0093         CDG2L  = 0092
CDG3H  = 0095          CDG3L  = 0094         CDVTH  = 008D         CDVTL  = 008C
CHPTIE= 0028           CH0TIE= 0020          CH1TIE= 0021          CH2TIE= 0022
CH3TIE= 0023           CH4TIE= 0024          CH5TIE= 0025          CH6TIE= 0026
CH7TIE= 0027           CKLOP  = 2815         CLEAR   4543          CLECC   4658
CLKCTL= 00F0           CLOCK   4F26          CLRLP   4547          CMC0H  = 0099
CMC0L  = 0098          CMC1H  = 009B         CMC1L  = 009A         CMC2H  = 009D
CMC2L  = 009C          CMC3H  = 009F         CMC3L  = 009E         CMINH  = 0097
CMINL  = 0096          CNTCTL= 00D7          CRCWRD= 0018          CSAVE   4F9D
CSRLH  = 0091          CSRLL  = 0090         CTCH   = 0085         CTCL   = 0084
CTSTH  = 008F          CTSTL  = 008E         CXCTH  = 0081         CXCTL  = 0080
CXINH  = 0083          CXINL  = 0082         C.     = 0001         C.AVAI= 0080
C.DP   = 0008          C.DSE  = 0010         C.DTU  = 0003         C.DVA  = 0008
C.FAIL= 00FC           C.FMT  = 0070         C.FNCT= 003E          C.GO   = 0001
C.INTC= 00FE           C.MAIN= 0020          C.NSA  = 0080         C.RCT  = 00FC
C.SER  = 0080          C.SHR  = 0040         C.SKPC= 000F          C.TAPE= 0040
C.WCS  = 0002          C4X5    4597          D      =%0002         DATAA   46BF
DATAAP  46C0           DATACT= 00D0          DATAE   46C1          DATAEP  46C2
DBUS    4F28           DBUSCT= 00C0          DBUSST= 00C0          DDRA   = 00D8
DDRAIN= 0010           DDRB   = 00D9         DDRBIN= 0002          DDRC   = 00DA
DDRCIN= 0001           DDRCO  = 0088         DDRCTL= 00DB          DIAFLG  4F22
DIAGPG= 4300           DIAGRM= 4F90          DIARD  = 000B         DONE1  = 0045
DONINT= 0010           DSAVE   4F9E          DSE    = 0006         DUMMU   4416  G
DUMMV   445C  G        DUMMY   431E  G       D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010           D.LAGC= 0020          D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008           D.WR4  = 0080         D4X5    4586          E      =%0003
ECC     465D           ECCBAD= 0042          ECCCHR  4680          ECCCOR= 0019
ECCOK  - 0041          ECCSTA= 001A          ECCTBL  4678          ECCTST= 000E
ECC1    466A           EDATA  = 0095         EOTCLR= 0003          ERFLG   4F93
ERLP   = 2809          ERLPA  = 280F         ERLPB  = 2812         ERLPE  = 280C
ERNUM   4F90           ERRCNT= 00D6          ESAVE   4F9F          EXIT    4540
E.ACRC= 0010           E.AMT  = 0020         E.CDP  = 0080         E.CRC  = 0080
E.PNTR= 0008           E.RPE  = 0040         E.STEC= 0001          E.TTEC= 0002
E.UNC  = 0004          E4X5    45B0          FIFORD= 006A          FORMAT  4F25
FOUND   4344           FWDTST= 0061          F4X5    45FC          GCRID  = 0089
GCRSET= 0002           GOODTM= 0092          GP4X5   4653          H      =%0004
HLSAVE  4FA0           IE     = 0008         INDAT   46BA          INDATP  46BB
INTSTA= 00E0           INV4X5  45E0          ITERA   4F9A          I.PWR  = 0020
I.RMPE= 0040           I5.5   = 0010         I6.5   = 0020         I7.5   = 0040
KCALL  = 005F          KCLR   = 007B         KDEP   = 003F         KENAB  = 0078
KEXAM  = 003E          KEYBRD= 00C8          KEY1   = 0078         KEY10  = 006D
KEY11  = 006E          KEY12  = 006F         KEY13  = 005C         KEY14  = 005D
```

```
KEY15 = 005E            KEY16 = 005F            KEY17 = 003C            KEY18 = 003D
KEY19 = 003E            KEY2  = 0079            KEY20 = 003F            KEY3  = 007A
KEY4  = 007B            KEY5  = 0074            KEY6  = 0075            KEY7  = 0076
KEY8  = 0077            KEY9  = 006C            KINTA = 006F            KLDAD = 003D
KN0   = 003C            KN1   = 005C            KN2   = 005D            KN3   = 005E
KN4   = 006C            KN5   = 006D            KN6   = 006E            KN7   = 0074
KN8   = 0075            KN9   = 0076            KU2   = 0079            KU3   = 007A
KU8   = 0077            L     =%0005            LBLANK= 000F            LCE   = 000B
LCH   = 000C            LCL   = 000D            LCP   = 000E            LC0   = 0000
LC1   = 0001            LC2   = 0002            LC3   = 0003            LC4   = 0004
LC5   = 0005            LC6   = 0006            LC7   = 0007            LC8   = 0008
LC9   = 0009            LDLEDA= 00CA            LDLEDB= 00CB            LDLEDC= 00CC
LDLEDD= 00CD            LDLEDE= 00CE            LDLEDF= 00CF            LKDIAG= 2800
LKKBD = 004C            LKKEY = 0049            LKLWMG= 0058            LKLWMP= 0055
LKLWPG= 0052            LKLWPP= 004F            LKMOD7= 0046            LKOPR = 0046
LPFLG   4F94            LPNUM   4F92            M     =%0006            MBSEL = 00E0
MB.A  = 0008            MB.B  = 0004             EMTOP= 4FFF            MINUS = 000A
MM    = 8000            MSE   = 0008            MSGN  = 0008            MTACLR= 0000
MT.ARA= 0020            MT.CPE= 0080            MT.DSE= 0001            MT.FWD= 0040
MT.INH= 0008            MT.LWR= 0004            MT.MOT= 0002            MT.NWT= 0080
MT.PEC= 0040            MT.PSB= 0004            MT.PS0= 0001            MT.PS1= 0002
MT.REV= 0020            MT.WRT= 0010            MT.Z  = 0008            M.ATA = 0080
M.CAPE= 0020            M.CONT= 0080            M.DEM = 0020            M.EBL = 0010
M.EXC = 0008            M.FAIL= 0008            M.ILR = 0010            M.INIT= 0010
M.OCC = 0020            M.ONLI= 0001            M.PE  = 0040            M.PORT= 0080
M.RDEN= 0002            M.RDPE= 0008            M.RUN = 0004            M.SCLK= 0001
M.TRA = 0040            M.UNIT= 0007            M.WCLK= 0040            M.WCLN= 0080
M.WREN= 0080            M5.5  = 0001            M6.5  = 0002            M7.5  = 0004
NOTCAP= 0088            OKAY  = 00FF            OPRRAM= 4300            OPSTRT= 0058
OPVER = 0040            PADCNT= 00D5            PADCRC= 0080            PDIAG = 0048
PEID  = 008A            PENAB = 004C            PESET = 0001            PL    = 00B1
POINTB  4631            POINTE  4619            PO4X5   459D            PRDD  = 004C
PRENT = 009C            PS    = 00B2            PSTAT = 0048            PSW   =%0009
P.AMTP= 0001            P.BCTC= 0040            P.CMDP= 0020            P.INTE= 0080
P.LCS = 0040            P.LWR = 0020            P.RDP = 0002            P.RPEN= 0004
P.RPST= 0002            P.RP0E= 0020            P.RP1E= 0010            P.RP2E= 0020
P.RP3E= 0010            P.SING= 0080            P.STAT= 0002            P.STPE= 0080
P.TACH= 0008            P.TUPR= 0010            P.WCSP= 0004            P.WDS = 0040
P.WFLP= 0001            P.WPEN= 0010            P.WP0E= 0008            P.WP1E= 0004
P.WP2E= 0008            P.WP3E= 0004            P.5VOK= 0002            QUE   = 281B
QUEM  = 281E            RAMT  = 0010            RARA  = 0006            RARAI = 0004
RCHBDO= 0048            RCHBD1= 0047            RCHOK = 0046            RCHTST= 000C
RCLRT = 000D            RCMD  = 000B            RCMLP = 0003            RCONT = 0080
RDATA = 0017            RDCLK = 0010            RDON  = 0011            READG = 0007
REND  = 0014            REQST = 2806            RESCHR= 00D1            REVTST= 0064
REWIND= 0004            RFIFOL= 0008            RGCLK = 0002            RGCRI = 0003
RIBG  = 0001            RILL  = 0012            RINST = 000C            RMCTST= 0008
RMK2  = 0013            RNOP  = 0000            RPATH = 0001            RPBAD = 0044
RPCHI = 0001            RPCLK = 0003            RPCTL = 0009            RPEI  = 0002
RPFAIL= 0000            RPF1  = 009D            RPF2  = 009E            RPOK  = 0043
RPOSTN= 0016            RPSTA = 0015            RRCMT = 000A            RSTAT = 0002
RTIEB = 000A            RTIER = 0030            RTM   = 0005            RUNKI = 0009
RUPTST= 005E            RWDUNL= 0005            R.AMT = 0001            R.BOP = 0008
R.DATA= 0040            R.DON = 0002            R.DRDY= 0010            R.END = 0010
```

```
R.ILL  = 0004          R.JVOK= 0004          R.MK2  = 0008          R.PLOD= 0008
R.PL00= 0010          R.PL01= 0020          R.POST= 0020          R.STNM= 0002
R.STOP= 0004          R.STPC= 0001          R.TBJN= 0080          R.TSTD= 0040
R.VOK  = 0080          R00H   = 0081          R00L   = 0080          R01H   = 0083
R01L   = 0082          R02H   = 0085          R02L   = 0084          R03H   = 0087
R03L   = 0086          R04H   = 0089          R04L   = 0088          R05H   = 008B
R05L   = 008A          R06H   = 008D          R06L   = 008C          R07H   = 008F
R07L   = 008E          R10H   = 0091          R10L   = 0090          R11H   = 0093
R11L   = 0092          R12H   = 0095          R12L   = 0094          R13H   = 0097
R13L   = 0096          R14H   = 0099          R14L   = 0098          R15H   = 009B
R15L   = 009A          R16H   = 009D          R16L   = 009C          R17H   = 009F
R17L   = 009E          R7.5   = 0010          SELCLR= 0000          SETATA= 00A1
SID    = 0080          SOD    = 0080          SOE    = 0040          SP     =%0008
SSCLK  = 0040          SSTEP  = 0005          STACK  = 4FFF          STATRM= 4F20
STPCT    4F20          STRSP  = 5000          TAB4X5   4635          TADR00= 0080
TADR01= 0081          TADR02= 0082          TADR03= 0083          TADR04= 0084
TADR05= 0085          TADR06= 0086          TADR07= 0087          TADR10= 0088
TADR11= 0089          TADR12= 008A          TADR13= 008B          TAMT   = 0044
TASEL  = 0080          TCMD   = 0040          TC.FWD= 0040          TC.INH= 0008
TC.LWR= 0004          TC.REV= 0020          TC.WRT= 0010          TEMP     4F99
TEST1    4300          TIETIM   46B9          TIMTBL   46B0          TMF    = 0099
TMRDY  = 0040          TRANS    4681          TRANS1   4683          TRANS2   468B
TRK      4693          TRKA     469F          TRKENA= 00D2          TRNIN    4645
TRNOUT   4649          TRNTMP   4654          TSET   = 2803          TSTEND= 2818
TSTS   = 0040          TST01A   441F          TST01B   4423          TST01C   4376
TST01F   438E          TST01G   43B7          TST01H   43E5          TST01I   439A
TST01K   4465          TST01L   447B          TST01M   4493  G       TST01N   44DD
TST01Q   44BE          TST01R   44B5  G       TST01S   44D2          TST01T   44CC  G
TST01U   44FA  G       TST01V   451C  G       TST01X   434E          TST01Y   4467
TUSEL0= 00D1          TUSEL1= 00D2          TU78   = 0010          T.ATH0= 0001
T.ATH1= 0002          T.BOT  = 0004          T.EOT  = 0002          T.FPT  = 0001
T.NTHR= 0004          T.ONL  = 0020          T.PES  = 0008          T.PSBJ= 0020
T.PSUJ= 0008          T.PS1J= 0010          T.RDY  = 0080          T.RDYO= 0040
T.RWD  = 0010          T.SCLK= 0002          T4X5     4572          UIBG   = 00A1
UNITMP   46BC          UNITSL   4327          VALFC    4F98          VALTB    4F95
VELTST= 005B          WDR.P  = 0010          WMCCTL= 00D3          WMCERR= 00DA
WMCSTA= 00D0          WRTCLK= 0000          WRTDAT= 00D3          W.ACRC= 0004
W.CRC  = 0008          W.DIAG= 0002          W.DONN= 0040          W.ECC  = 0010
W.ENAB= 0080          W.ERR  = 0020          W.FMT  = 0070          W.GCR  = 0010
W.LEFT= 0004          W.ONES= 0020          W.RESI= 0002          W.REV  = 0004
W.ROME= 0010          W.RST  = 0001          W.SKIP= 000F          W.WRIT= 0008
W.XFER= 0020          X      =%000A          X.DONN= 0080          X.ENAB= 0040
X.PEPE= 0002          X.ROME= 0001          X.WCLK= 0001          Y      =%000B
.      = 46C3
```

ERRORS DETECTED: 0

*ECC3.A78/PTP,ECC3=NLIST,PARAM,MACRO,LIST,ECC3
RUN-TIME: 4 5 0 SECONDS
CORE USED: 10K

     3   "MEMORY MAP" EXTENDED ADDRESS BIT
    19   READ MICRO CONTROLLER CONTROL & STATUS WORDS
   139   TRANSPORT/PORT COMMAND + STATUS WORDS
   380   COMMON ADDRESS SPACE REGISTERS
   444   MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
   485   MASSBUS CONTROL WORD DEFINITIONS
   557   WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
   709   KEYBOARD/DISPLAY CONTROL + STATUS WORD
   797   MISCELLANEOUS VARIABLE DEFINITIONS
   801   RAM ADDRESS DEFINITIONS
   819   LINKAGE TABLE ADDRESSES
   842   BIT DEFINITIONS
   856   RIM AND SIM INSTRUCTION BIT DEFINITIONS
   877   INTERRUPTABLE INSTRUCTION MACROS
   924   DIAGNOSTIC VARIABLES
   945   COMMON VARIABLE STORAGE
   959   MACRO CALL DEFINITIONS
  1047   TM78 MICRO-DIAGNOSTIC MACRO CALLS
  1101   8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
  1333   TEST 1 - ECC BAD DATA TWO TRACKS CORRECT POINTERS
  1830   TEST 2 - ECC GOOD DATA THREE HARD POINTERS
  2339   SUBROUTINE CLEAR ALL TU PORTS
  2383   SUBROUTINE 4X5 TRANSLATE A SUBGROUP
  2527   SUBROUTINE CALCULATE POINTER
  2567   TABLE 4 X 5 TRANSLATION
  2608   SUBROUTINE VARIABLES
  2622   SUBROUTINE CLEAR ECC
  2634   SUBROUTINE CALCUALTE ECC CHARACTER
  2676   SUBROUTINE POLYNOMIAL BIT TRANSLATION
  2699   SUBROUTINE TRK
  2787   PROGRAM VARIABLES

```
                ; ************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;                "BYTE/SCLK COUNT NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                MM = DATA FORMAT FROM CAS REGISTER 2
                ;                NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;                BYTE-SCLK COUNT = LLL
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                LLL = AS ABOVE
                ;                MM  = AS ABOVE
                ;                NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;                TRANSITION COUNT = LLL
                ;                WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;                EXPECTED 18 BITS =      E EEEEEE
                ;                ACTUAL 18 BITS =        A AAAAAA
                ;
                ;                WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;                BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;                OF CAS REG 15 LOW BYTE.
                ;
                ;                ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;                AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;                17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;                TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;                AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;                "SUBGROUP NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       10 -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ; ************************************************************************
```

```
1016                    ;*************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ;*************************************************************************
1035                    ;*************************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAK (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ;*************************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                        .TITLE  ECC4 - ECC CONTROLLER PART #4
                        .SBTTL  TEST 1 - ECC BAD DATA TWO TRACKS CORRECT POINTERS
                        ;ID     ECC4-ERROR CORRECTION CONTROLLER PART #4
4300                    ST
                        ; ***************************************************************************
                        ;*TEST TITLE
                        ;*---- -----
                        ;*ECC BAD DATA TWO TRACKS CORRECT POINTERS
4300                    SD
                        ; ***************************************************************************
                        ;*DESCRIPTION
                        ;*-----------
                        ;*THIS TEST CHECKS THE ABILITY OF THE ECC CONTROLLER TO PERFORM TWO
                        ;*TRACK ERROR CORRECTION WITH INCORRECT DATA IN TWO TRACKS, AND TWO
                        ;*CORRECT POINTERS ALSO PROVIDED.  THE PROGRAM CHECKS THAT THE DATA IS
                        ;*CORRECTED, AND THAT THE TWO TRACK ERROR CORRECT STATUS BIT IS SET.
                        ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
                        ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
                        ;*MESSAGE, AND THE DATA BIT POSITIONS OF THE INCORRECT DATA
                        ;*CAN BE DETERMINED FROM THE FOLLOWING TABLE.
                        ;*
                        ;*      BYTE COUNT AND
                        ;*      INPUT/EXPECTED          DATA BIT POSITIONS
                        ;*          DATA                WITH ECC CORRECTION
                        ;*      --------------          -------------------
                        ;*
                        ;*          XXXX0000                    0,1
                        ;*          XXXX0001                    1,2
                        ;*          XXXX0010                    2,3
                        ;*          XXXX0011                    3,4
                        ;*          XXXX0100                    4,5
                        ;*          XXXX0101                    5,6
                        ;*          XXXX0110                    6,7
                        ;*          XXXX0111                    7,P
                        ;*          XXXX1000                    P,0
                        ;*          XXXX1001                    1,2
                        ;*          XXXX1010                    2,3
                        ;*          XXXX1011                    3,4
                        ;*          XXXX1100                    4,5
                        ;*          XXXX1101                    5,6
                        ;*          XXXX1110                    6,7
                        ;*          XXXX1111                    7,P
                        ;*
                        ;*      X = DON'T CARE BITS
                        ;*
                        ;*THE POINTERS PROVIDED FOR THE TRACK IN ERROR ARE ALSO DERIVED FROM THE
                        ;*INPUT/EXPECTED DATA AS PER THE FOLLOWING TABLE.
                        ;*
                        ;*      INPUT/EXPECTED
                        ;*          DATA                    POINTER GENERATED
                        ;*      --------------              -----------------
                        ;*
                        ;*          XX00XXXX                HISTORY
                        ;*          XX01XXXX                PHASE TIE
```

```
                                ;*         XX10XXXX                    AMTIE
                                ;*         XX11XXXX                    ILLEGAL 5X4
                                ;*
                                ;*      X = DON'T CARE BITS
 4300                           SP
  (1)                           ; ***************************************************************************
  (1)                           ;*PROCEDURE
  (1)                           ;*---------
                                ;*BGNTST
                                ;*    SET NORMAL READ PATH CLOCK
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
                                ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
                                ;*    DELAY 50 MICROSECONDS
                                ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
                                ;*    CLEAR THE INPUT DATA
                                ;*    CLEAR BAD TRANSLATION POINTERS
                                ;*  BGNDO
                                ;*  : ENABLE THE READ PATH CLOCK
                                ;*  : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
                                ;*  : CLOCK THE FIFO'S
                                ;*  : ISSUE CLEAR ALL COMMAND
                                ;*  : WAIT
                                ;*  : STOP THE READ PATH, SET TIE BUS JAM AND SET PLO DISABLE
                                ;*  : CALL SUBROUTINE CLECC
                                ;*  : SET THE LOOP COUNT TO 7
                                ;*  : BGNDO
                                ;*  : :  GET THE INPUT DATA BYTE
                                ;*  : :  CALL SUBROUTINE ECC
                                ;*  : :  DECREMENT THE LOOP COUNT
                                ;*  : :  DO UNTIL THE LOOP COUNT=0
                                ;*  : ENDDO
                                ;*  : SET THE LOOP COUNT TO 3
                                ;*  : BGNDO
                                ;*  : :  FILL THE TRANSLATOR SUBROUTINE BUFFER WITH THE DATA BYTE
                                ;*  : :  CALL SUBROUTINE TRK
                                ;*  : :  CALL SUBROUTINE BADT4X5
                                ;*  : :  INIT THE LOOP COUNT TO 5
                                ;*  : :  BGNDO
                                ;*  : :  :  MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
                                ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
                                ;*  : :  :  MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
                                ;*  : :  :  CLOCK THE DATA INTO THE FIFO
                                ;*  : :  :  DECREMENT THE LOOP COUNT
                                ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
                                ;*  : :  :  DO UNTIL LOOP COUNT = 0
                                ;*  : :  ENDDO
                                ;*  : :  FILL THREE BYTES OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE DATA BYTE
                                ;*  : :  FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
                                ;*  : :  CALL SUBROUTINE TRK
                                ;*  : :  CALL SUBROUTINE BADT4X5
                                ;*  : :  INIT THE LOOP COUNT TO 5
                                ;*  : :  BGNDO
                                ;*  : :  :  MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
```

```
1431                    ;*  : : :  MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
1432                    ;*  : : :  CLOCK THE DATA INTO THE FIFO
1433                    ;*  : : :  DECREMENT THE LOOP COUNT
1434                    ;*  : : :  DO UNTIL THE LOOP COUNT = 0
1435                    ;*  : : ENDDO
1436                    ;*  : : DECREMENT THE LOOP COUNT
1437                    ;*  : : DO UNTIL THE LOOP COUNT = 0
1438                    ;*  : ENDDO
1439                    ;*  : ISSUE DIAGNOSTIC READ COMMAND
1440                    ;*  : SET UP WATCHDOG TIMER COUNT
1441                    ;*  : BGNDO
1442                    ;*  : :    SINGLE STEP THE READ PATH
1443                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1444                    ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1445                    ;*  : ENDDO
1446                    ;*  : IF WATCHDOG TIMER=0
1447                    ;*  : :    THEN-ERROR
1448                    ;*  : :    ELSE-CONTINUE
1449                    ;*  : ENDIF
1450                    ;*  : GET THE TIE BUS TABLE ADDRESS
1451                    ;*  : SINGLE STEP THE READ PATH
1452                    ;*  : INIT THE LOOP COUNT TO 0
1453                    ;*  : BGNDO
1454                    ;*  : :    IF LOOP COUNT=TIE BUS JAM COUNT 1 OR JAM COUNT 2
1455                    ;*  : :    :  THEN-CALL SUBROUTINE POINTER
1456                    ;*  : :    :  ELSE-CLEAR TIE BUS POINTER
1457                    ;*  : :    ENDIF
1458                    ;*  : :    SINGLE STEP THE READ PATH
1459                    ;*  : :    DECREMENT THE LOOP COUNT
1460                    ;*  : :    DO UNTIL LOOP COUNT = 9
1461                    ;*  : ENDDO
1462                    ;*  : SET UP WATCHDOG TIMER COUNT
1463                    ;*  : BGNDO
1464                    ;*  : :  SINGLE STEP THE READ PATH
1465                    ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
1466                    ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
1467                    ;*  : ENDDO
1468                    ;*  : IF WATCHDOG TIMER=0
1469                    ;*  : :  THEN-ERROR-EXIT TEST
1470                    ;*  : :  ELSE-CONTINUE
1471                    ;*  : ENDIF
1472                    ;*  : SINGLE STEP THE REA  PATH
1473                    ;*  : INIT THE  LOOP COUNT TO 7
1474                    ;*  : BGNDO
1475                    ;*  : :  SINGLE STEP THE READ PATH
1476                    ;*  : :  COMPARE CORRECTED DATA WITH INPUT DATA
1477                    ;*  : :  IF NOT EQUAL
1478                    ;*  : :  :  THEN-ECC ERROR
1479                    ;*  : :  :  ELSE-CONTINUE
1480                    ;*  : :  ENDIF
1481                    ;*  : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
1482                    ;*  : :  IF NOT EQUAL
1483                    ;*  : :  :  THEN-ECC ERROR
1484                    ;*  : :  :  ELSE-CONTINUE
```

```
1485                                ;*  : :  ENDIF
1486                                ;*  : :  DECREMENT THE LOOP COUNT
1487                                ;*  : :  DO UNTIL THE LOOP COUNT = 0
1488                                ;*  : ENDDO
1489                                ;*  : SINGLE STEP THE READ PATH
1490                                ;*  : IF EXPECTED NOT EQUAL
1491                                ;*  : :  THEN-ECC ERROR
1492                                ;*  : :  ELSE-CONTINUE
1493                                ;*  : ENDIF
1494                                ;*  : INPUT THE ECC STATUS
1495                                ;*  : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
1496                                ;*  : IF THE RESULTING STATUS=SINGLE TRACK ERROR CORRECT + POINTER ERROR
1497                                ;*  : :  THEN-CONTINUE
1498                                ;*  : :  ELSE-ERROR
1499                                ;*  : ENDIF
1500                                ;*  : INCREMENT THE INPUT DATA
1501                                ;*  : DO UNTIL INPUT DATA = 0
1502                                ;*  ENDDO
1503                                ;*ENDTST
1504  4300                          SE
 (1)                                ; ***************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1505                                ;*ECC4 MICRO TEST 01
1506                                ;*ECC4 MICRO ERROR 01
1507                                ;*ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1508                                ;*M8951, M8950'S
1509                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1510                                ;*FATAL ERROR - TEST ABORTED
1511                                ;*
1512                                ;*ECC4 MICRO TEST 01
1513                                ;*ECC4 MICRO ERROR 02
1514                                ;*ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1515                                ;*M8951, M8950'S
1516                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
1517                                ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1518                                ;*BYTE/SCLK COUNT NUMBER = LLL
1519                                ;*FATAL ERROR - MICRO TEST ABORTED
1520                                ;*
1521                                ;*ECC4 MICRO TEST 01
1522                                ;*ECC4 MICRO ERROR 03
1523                                ;*ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1524                                ;*M8951, M8950'S
1525                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
1526                                ;*SENT TO THE ECC
1527                                ;*BYTE/SCLK COUNT NUMBER = LLL
1528                                ;*FATAL ERROR - MICRO TEST ABORTED
1529                                ;*
1530                                ;*ECC4 MICRO TEST 01
1531                                ;*ECC4 MICRO ERROR 04
1532                                ;*ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1533                                ;*M8951, M8950'S
1534                                ;*CORRECTED DATA FROM THE ECC-INCORRECT
1535                                ;*BYTE/SCLK COUNT NUMBER = LLL
```

```
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*ECC4 MICRO TEST 01
                                        ;*ECC4 MICRO ERROR 05
                                        ;*ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*
                                        ;*ECC4 MICRO TEST 01
                                        ;*ECC4 MICRO ERROR 06
                                        ;*ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                        ;*M8951, M8950'S
                                        ;*CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*
                                        ;*ECC4 MICRO TEST 01
                                        ;*ECC4 MICRO ERROR 07
                                        ;*ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTER
                                        ;*M8951, M8950'S
                                        ;*CORRECTED ECC CHARACTER INCORRECT
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
                                        ;*
                                        ;*ECC4 MICRO TEST 01
                                        ;*ECC4 MICRO ERROR 10
                                        ;*ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
                                        ;*M8951, M8950'S
                                        ;*ECC STATUS INCORRECT AFTER A DATA GROUP
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*BYTE/SCLK COUNT NUMBER = LLL
1568 4300                               S
 (1)                                    ; ************************************************************************************
1569
1570 4300                               TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1) 4300  3E  01          7.0                   MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1) 4302  CD  03  28      3.0                   CALL    TSET            ;SETUP THE TEST
1571                                    ;%ECC4-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1572                                    ;&M8951, M8950'S
1573
1574 4305                                       REQ     @7,0,0,0,0
 (1) 4305  CD  06  28     18.0                   CALL    REQST
 (1) 4308  00                                    .BYTE   0               ;DATA PATTERN NUMBER
 (1) 4309  00  00                                .WORD   0               ;SYSTEM "0" COUNT
 (1) 430B  00  00                                .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1) 430D  00                                    .BYTE   0               ;DATA COMPARE FLAG IF  1
 (1) 430E  07                                    .BYTE   @7              ;REQUEST CODE
1575 430F                                       RIN     R12L
 (1) 430F  DB  94         10.0                   IN      R12L            ;READ R12L INTO AC
 (1) 4311  7F              4.0                   MOV     A,A             ;RETRY LINK
1576 4312  32  38  49     13.0          STA     UNITMP
1577 4315  A7              4.0          ANA     A
1578 4316  C2  27  43     10.0          JNZ     UNITSL
```

```
1579  4319                                        ERR     EXIT,DUMMY
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28        18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0001                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                          .BYTE
 (1)  431E  CD  15  28        18.0                DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  74  47        10.0                        JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1580                                    ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1581                                    ;>FATAL ERROR - TEST ABORTED
1582  4324  C3  74  47        10.0                JMP     EXIT
1583
1584  4327  06  00              7.0   UNITSL: MVI     B,0
1585  4329  3A  38  49         13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1586  432C  E6  01               7.0           ANI     @01             ;PORT 0?
1587  432E  C2  44  43         10.0           JNZ     FOUND           ;YES-GO USE PORT #0
1588  4331  04                   4.0           INR     B               ;NO-UPDATE POINTER TO PORT #1
1589  4332  3A  38  49         13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1590  4335  E6  02               7.0           ANI     @02             ;PORT 1?
1591  4337  C2  44  43         10.0           JNZ     FOUND           ;YES-GO USE PORT #1
1592  433A  04                   4.0           INR     B               ;NO-UPDATE POINTER TO PORT #2
1593  433B  3A  38  49         13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1594  433E  E6  04               7.0           ANI     @04             ;PORT 2?
1595  4340  C2  44  43         10.0           JNZ     FOUND           ;YES-GO USE PORT #2
1596  4343  04                   4.0           INR     B               ;NO-ASSUME PORT #3
1597  4344  CD  82  47         18.0   FOUND:  CALL    CLEAR           ;CLEAR ALL TU PORTS
1598  4347  DB  E0              10.0           IN      INTSTA
1599  4349  E6  80               7.0           ANI     BIT7
1600  434B  B0                   4.0           ORA     B
1601  434C  D3  E0              10.0           OUT     MBSEL
1602  434E  32  35  49         13.0           STA     SUNIT           ;SAVE FOR NEXT TEST
1603
1604  4351  3E  10               7.0   TST01X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
1605  4353  D3  F0              10.0           OUT     CLKCTL
1606  4355  3E  10               7.0           MVI     A,W.GCR
1607  4357  D3  D3              10.0           OUT     WMCCTL
1608  4359  3E  04               7.0           MVI     A,P.RPEN
1609  435B  D3  4C              10.0           OUT     PENAB
1610
1611  435D  3E  60               7.0           MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
1612  435F  D3  48              10.0           OUT     PDIAG           ;LCS, LWR
1613  4361  AF                   4.0           XRA     A
1614  4362  D3  44              10.0           OUT     TAMT
1615  4364  3E  0A               7.0           MVI     A,10            ;DELAY
1616  4366  3D                   4.0   1$:     DCR     A               ;50
1617  4367  C2  66  43         10.0           JNZ     1$              ;MICROSECONDS
1618
1619  436A  D3  D2              10.0           OUT     TRKENA          ;CLEAR ALL TRACKS FROM
1620  436C  D3  0A              10.0           OUT     RTIEB           ;CLEAR THE TIE BUS REGISTER
1621  436E  D3  D2              10.0           OUT     TRKENA
1622  4370  32  94  48         13.0           STA     BAD1            ;CLEAR THE INVALID
1623  4373  32  95  48         13.0           STA     BAD2            ;TRACK TRANSLATIONS
1624  4376  32  96  48         13.0           STA     BAD3            ;POINTERS
1625  4379  32  36  49         13.0           STA     INDAT           ;CLEAR THE INPUT DATA BYTE
```

```
1626  437C                                    ROUT    R05H
 (1)  437C  D3  8B          10.0                 OUT     R05H            ;WRITE AC INTO R05H
 (1)  437E  7F               4.0                 MOV     A,A             ;RETRY LINK
1627
1628  437F  3E  A8           7.0  TST01C: MVI     A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1629                                                                      ;SET TIE BUS JAM AND PLO DISABLE
1630  4381  D3  09          10.0          OUT     RPCTL
1631  4383  D3  08          10.0          OUT     RFIFOL                  ;CLOCK THE FIFO'S
1632  4385  3E  0D           7.0          MVI     A,RCLRT                 ;ISSUE CLEAR ALL COMMAND
1633  4387  D3  0B          10.0          OUT     RCMD
1634
1635  4389  00               4.0          NOP                             ;WAIT
1636  438A  00               4.0          NOP
1637  438B  00               4.0          NOP
1638  438C  00               4.0          NOP
1639  438D  00               4.0          NOP
1640
1641  438E  3E  A9           7.0          MVI     A,R.PLO1!R.STPC.R.PLOD!R.TBJN   ;STOP THE READ PATH
1642                                                                      ;SET TIE BUS JAM AND PLO DISABLE
1643  4390  D3  09          10.0          OUT     RPCTL
1644  4392  CD  97  48       18.0          CALL    CLECC                   ;CLEAR THE ECC CHARACTER
1645  4395  16  07           7.0          MVI     D,@7                    ;SET UP THE LOOP COUNT
1646  4397  3A  36  49       13.0  TST01F: LDA     INDAT                   ;GET THE INPUT DATA
1647  439A  CD  9C  48       18.0          CALL    ECC                     ;CALCULATE ECC FOR
1648  439D  15                4.0          DCR     D                       ;THE 7 INPUT BYTES
1649  439E  C2  97  43       10.0          JNZ     TST01F
1650
1651  43A1  0E  03           7.0          MVI     C,@3                    ;INIT THE LOOP COUNT
1652  43A3  3A  36  49       13.0  TST01I: LDA     INDAT                   ;GET THE INPUT DATA
1653  43A6                                ROUT    R05L
 (1)  43A6  D3  8A          10.0                 OUT     R05L            ;WRITE AC INTO R05L
 (1)  43A8  7F               4.0                 MOV     A,A             ;RETRY LINK
1654  43A9  32  84  48       13.0          STA     TRNIN                   ;FILL THE TRANSLATOR
1655  43AC  32  85  48       13.0          STA     TRNIN+1                 ;SUBROUTINE INPUT
1656  43AF  32  86  48       13.0          STA     TRNIN+2                 ;BUFFER
1657  43B2  32  87  48       13.0          STA     TRNIN+3                 ;
1658  43B5  CD  D8  48       18.0          CALL    TRK
1659  43B8  AF                4.0          XRA     A                       ;CLEAR THE ACCUMULATOR
1660  43B9  32  96  48       13.0          STA     BAD3                    ;CLEAR THE THIRD POINTER
1661  43BC  CD  AD  47       18.0          CALL    BADT4X5                 ;TRANSLATE THE SUBGROUP
1662
1663  43BF  21  88  48       10.0          LXI     H,TRNOUT                ;GET POINTER TO TRANSLATE DATA TABLE
1664  43C2  06  05           7.0          MVI     B,@5                    ;SET UP LOOP COUNT
1665  43C4  7E                7.0  TST01G: MOV     A,M                     ;GET A DATA BYTE
1666  43C5  D3  40           10.0          OUT     TCMD                    ;STORE DATA IN COMMAND ADDRESS
1667  43C7  23                6.0          INX     H                       ;POINT TO DATA PARITY
1668  43C8  7E                7.0          MOV     A,M                     ;GET THE DATA PARITY
1669
1670  43C9  07                4.0          RLC                             ;POSITION FOR OUTPUT
1671  43CA  F6  60            7.0          ORI     P.LWR!P.LCS             ;OR IN CONTROL BITS
1672  43CC  D3  48           10.0          OUT     PDIAG                   ;OUTPUT THE DATA PARITY
1673
1674  43CE  D3  08           10.0          OUT     RFIFOL                  ;CLOCK DATA INTO THE FIFO'S
1675
```

```
1676  43D0  23              6.0          INX     H               ;UPDATE THE TABLE POINTER
1677  43D1  05              4.0          DCR     B               ;DECREMENT LOOP COUNT
1678  43D2  C2  C4  43     10.0          JNZ     TST01G          ;DO UNTIL LOOP COUNT = 0
1679
1680  43D5  3A  36  49     13.0          LDA     INDAT           ;GET THE INPUT DATA
1681  43D8  32  84  48     13.0          STA     TRNIN           ;FILL THE FIRST THREE
1682  43DB  32  85  48     13.0          STA     TRNIN+1         ;BYTES OF THE TRANSLATOR
1683  43DE  32  86  48     13.0          STA     TRNIN+2         ;SUBROUTINE WITH THE INPUT DATA
1684  43E1  3A  C5  48     13.0          LDA     ECCCHR          ;STORE THE CALCULATED
1685  43E4  32  87  48     13.0          STA     TRNIN+3         ;ECC CHARACTER AS THE LAST CHARACTER
1686  43E7  CD  D8  48     18.0          CALL    TRK
1687  43EA  AF              4.0          XRA     A               ;CLEAR THE ACCUMULATOR
1688  43EB  32  96  48     13.0          STA     BAD3            ;CLEAR THE THIRD POINTER
1689  43EE  CD  AD  47     18.0          CALL    BADT4X5         ;TRANSLATE THE SECOND SUBGROUP
1690
1691  43F1  21  88  48     10.0          LXI     H,TRNOUT        ;GET THE POINTER TO INPUT DATA TABLE
1692  43F4  06  05          7.0          MVI     B,@5            ;SET UP THE LOOP COUNT
1693  43F6  7E              7.0  TST01H: MOV     A,M             ;GET A DATA BYTE
1694  43F7  D3  40         10.0          OUT     TCMD            ;STORE DATA IN CMD ADDRESS
1695  43F9  23              6.0          INX     H               ;POINT TO DATA PARITY BIT
1696  43FA  7E              7.0          MOV     A,M             ;GET THE DATA PARITY
1697  43FB  07              4.0          RLC                     ;POSITION FOR OUTPUT
1698  43FC  F6  60          7.0          ORI     P.LWR!P.LCS     ;OR IN CONTROL BITS
1699  43FE  D3  48         10.0          OUT     PDIAG           ;OUTPUT THE DATA PARITY
1700  4400  D3  08         10.0          OUT     RFIFOL          ;CLOCK DATA INTO THE FIFOS
1701  4402  23              6.0          INX     H               ;UPDATE THE TABLE POINTER
1702  4403  05              4.0          DCR     B               ;DECREMENT THE LOOP COUNT
1703  4404  C2  F6  43     10.0          JNZ     TST01H          ;DO UNTIL LOOP COUNT=0
1704
1705  4407  0D              4.0          DCR     C               ;PUT THE GROUP IN
1706  4408  C2  A3  43     10.0          JNZ     TST01I          ;THE FIFO TWICE
1707
1708  440B  3E  0B          7.0          MVI     A,DIARD         ;LOAD THE DIAGNOSTIC READ
1709  440D  D3  0B         10.0          OUT     RCMD            ;COMMAND
1710  440F  11  01  00     10.0          LXI     D,1             ;SET WATCH DOG INCREMENT
1711  4412  21  A8  FD     10.0          LXI     H,-600          ;SET WATCH DOG COUNT TO 600
1712  4415  D3  0C         10.0  2$:     OUT     RINST           ;STEP THE READ PATH
1713  4417  DB  01         10.0          IN      RPCHI           ;DATA READY SET?
1714  4419  E6  10          7.0          ANI     R.DRDY
1715  441B  C2  30  44     10.0          JNZ     TST01A          ;YES-GO PROCESS
1716  441E  19              10.0         DAD     D               ;WATCH DOG TIMEOUT?
1717  441F  D2  15  44     10.0          JNC     2$              ;NO-CONTINUE
1718  4422                               ERR     TST01C,DUMMW,1
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4422  CD  09  28     18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4425  02                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4426  01                                      .BYTE   1
 (1)  4427  CD  15  28     18.0          DUMMW:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  442A  DA  7F  43     10.0                  JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1719                                    ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
1720                                    ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1721                                    ;<FATAL ERROR - MICRO TEST ABORTED
1722  442D  C3  74  47     10.0          JMP     EXIT
```

```
1723
1724  4430  D3  0C          10.0  TST01A: OUT   RINST
1725  4432  06  00           7.0          MVI   B,$0          ;INIT THE BIT TIME COUNT (LOOP COUNT)
1726  4434  00               4.0  TST01B: NOP
1727  4435  3A  32  49       13.0          LDA   TIET1         ;GET FIRST TIE BUS JAM TIME
1728  4438  B8               4.0          CMP   B             ;IS IT TIME TO JAM THE TIE BUS?
1729  4439  C2  42  44       10.0          JNZ   4$            ;NO - CONTINUE
1730  443C  CD  58  48       18.0          CALL  POINTER       ;YES - GO GET A REAL POINTER
1731  443F  C3  52  44       10.0          JMP   5$            ;
1732  4442  3A  33  49       13.0  4$:     LDA   TIET2         ;GET SECOND TIE BUS JAM TIME
1733  4445  B8               4.0          CMP   B             ;IS IT TIME TO JAM THE TIE BUS?
1734  4446  C2  4F  44       10.0          JNZ   6$            ;NO-CONTINUE
1735  4449  CD  58  48       18.0          CALL  POINTER       ;YES- GO GET A REAL POINTER
1736  444C  C3  52  44       10.0          JMP   5$            ;
1737  444F  AF               4.0  6$:     XRA   A             ;IF NOT THE BAD TRACK, PROVIDE A GOOD POINTER
1738  4450  D3  0A          10.0          OUT   RTIEB         ;LOAD THE POINTER
1739  4452  D3  0C          10.0  5$:     OUT   RINST         ;STEP CLOCK
1740  4454  04               4.0          INR   B             ;DECREMENT THE LOOP COUNT
1741  4455  78               4.0          MOV   A,B
1742  4456  FE  09           7.0          CPI   $9            ;DONE
1743  4458  C2  34  44       10.0          JNZ   TST01B        ;DO UNTIL LOOP COUNT = ZERO
1744  445B  AF               4.0          XRA   A
1745  445C  D3  0A          10.0          OUT   RTIEB         ;CLEAR THE TIE BUS REGISTER
1746  445E  D3  0C          10.0          OUT   RINST         ;CLOCK THE READ PATH
1747  4460  D3  0C          10.0          OUT   RINST         ;CLOCK THE READ PATH
1748  4462  11  01  00       10.0          LXI   D,1           ;SET WATCHDOG TIMER INCREMENT
1749  4465  21  A8  FD       10.0          LXI   H,-600        ;SET WATCHDOG COUNT TO 600
1750  4468  D3  0C          10.0  3$:     OUT   RINST         ;SINGLE STEP THE READ PATH
1751  446A  DB  01          10.0          IN    RPCHI         ;DATA READY SET?
1752  446C  E6  10           7.0          ANI   R.DRDY
1753  446E  C2  83  44       10.0          JNZ   TST01K        ;YES-GO PROCESS
1754  4471  19               10.0          DAD   D             ;NO-WATCHDOG TIMEOUT?
1755  4472  D2  68  44       10.0          JNC   3$            ;NO-CONTINUE
1756  4475                                 ERR   TST01C,DUMMX,1
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4475  CD  09  28       18.0                CALL  ERLP          ;PROCESS ERROR - DO 2.3
 (1)       0003                          MSGN  =     MSGN+1        ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4478  03                          .BYTE MSGN          ;MESSAGE NUMBER ID
 (1)  4479  01                          .BYTE 1
 (1)  447A  CD  15  28       18.0  DUMMX:: CALL  CKLOP         ;CHECK LOOP FUNCTION - DO 2.3
 (1)  447D  DA  7F  43       10.0          JC    TST01C        ;LOOP ADDRESS IF LOOP SPECIFIED
1757                             ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
1758                             ;>SENT TO THE ECC
1759                             ;<FATAL ERROR - MICRO TEST ABORTED
1760  4480  C3  74  47       10.0          JMP   EXIT
1761
1762  4483  16  01           7.0  TST01K: MVI   D,@1          ;INIT THE LOOP COUNT
1763  4485  00               4.0  TST01Y: NOP
1764  4486  3A  36  49       13.0          LDA   INDAT         ;GET THE INPUT DATA
1765  4489                                 ROUT  EDATA
 (1)  4489  D3  95          10.0                OUT   EDATA           ;WRITE AC INTO EDATA
 (1)  448B  7F               4.0                MOV   A,A             ;RETRY LINK
1766  448C  A7               4.0          ANA   A             ;SET CONDITION BITS
1767  448D  3E  00           7.0          MVI   A,0           ;CLEAR THE ACC.
```

```
1768  448F  E2  93  44   10.0          JPO     1$                ;ODD # OF ONES - CONTINUE
1769  4492  3C               4.0          INR     A                 ;EVEN # OF ONES - SET PARITY
1770  4493  32  37  49   13.0  1$:     STA     INDATP            ;STORE THE PARITY BIT
1771  4496  D3  0C           10.0  TST01L: OUT     RINST             ;SINGLE STEP THE READ LOGIC
1772  4498  3A  36  49   13.0          LDA     INDAT
1773  449B  47               4.0          MOV     B,A
1774  449C  DB  19           10.0          IN      ECCCOR
1775  449E  B8               4.0          CMP     B
1776  449F  CA  AA  44   10.0          JZ      TST01M
1777  44A2                             ROUT    ADATA
 (1)  44A2  D3  94           10.0                  OUT     ADATA             ;WRITE AC INTO ADATA
 (1)  44A4  7F               4.0                   MOV     A,A               ;RETRY LINK
1778  44A5                             ERRB    TST01C,TST01M,@1
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44A5  CD  12  28   18.0                        CALL    ERLPB             ;PROCESS ERROR - DO 2.3
 (1)       0004                                    MSGN    =       MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44A8  04                                     .BYTE   MSGN              ;MESSAGE NUMBER ID
 (1)  44A9  01                                     .BYTE   @1                ;PRINT ROUTINE NUMBER
 (1)  44AA  CD  15  28   18.0                  TST01M::        CALL    CKLOP             ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44AD  DA  7F  43   10.0                          JC      TST01C            ;LOOP ADDRESS IF LOOP SPECIFIED
1779                         ;>CORRECTED DATA FROM THE ECC-INCORRECT
1780  44B0  DB  15           10.0          IN      RPSTA             ;GET THE ECC PARITY BIT
1781  44B2  E6  80            7.0          ANI     E.CDP             ;GET THE CORRECTED PARITY BIT
1782  44B4  07               4.0          RLC                       ;POSITION BIT FOR COMPARE
1783  44B5  4F               4.0          MOV     C,A               ;SAVE IN REG. C
1784  44B6  3A  37  49   13.0          LDA     INDATP            ;GET THE EXPECTED PARITY BIT
1785  44B9  A7               4.0          ANA     A                 ;SET THE CONDITION BITS
1786  44BA  CA  D1  44   10.0          JZ      TST01QO           ;GO CHECK FOR PARITY=0
1787  44BD  79               4.0          MOV     A,C               ;ELSE CHECK FOR PARITY=1
1788  44BE  FE  01            7.0          CPI     @1
1789  44C0  CA  C8  44   10.0          JZ      TST01RG
1790  44C3                             ERR     TST01C,TSTU1RG,@1
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44C3  CD  09  28   18.0                        CALL    ERLP              ;PROCESS ERROR - DO 2.3
 (1)       0005                                    MSGN    =       MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44C6  05                                     .BYTE   MSGN              ;MESSAGE NUMBER ID
 (1)  44C7  01                                     .BYTE   @1
 (1)  44C8  CD  15  28   18.0                  TST01RG::       CALL    CKLOP             ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44CB  DA  7F  43   10.0                          JC      TST01C            ;LOOP ADDRESS IF LOOP SPECIFIED
1791                         ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
1792  44CE  C3  E1  44   10.0          JMP     TST01SZ           ;CONTINUE WITH TEST
1793  44D1  79               4.0  TST01QO: MOV     A,C               ;CHECK FOR PARITY = ZERO
1794  44D2  A7               4.0          ANA     A                 ;SET CONDITION BITS
1795  44D3  CA  DB  44   10.0          JZ      TST01TP           ;CONTINUE IF ZERO
1796  44D6                             ERR     TST01C,TST01TP,@1
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44D6  CD  09  28   18.0                        CALL    ERLP              ;PROCESS ERROR - DO 2.3
 (1)       0006                                    MSGN    =       MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44D9  06                                     .BYTE   MSGN              ;MESSAGE NUMBER ID
 (1)  44DA  01                                     .BYTE   @1
 (1)  44DB  CD  15  28   18.0                  TST01TP::       CALL    CKLOP             ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44DE  DA  7F  43   10.0                          JC      TST01C            ;LOOP ADDRESS IF LOOP SPECIFIED
1797                         ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
1798  44E1  7A               4.0  TST01SZ: MOV     A,D               ;GET THE CHARACTER COUNT
```

```
1799  44E2  3C                    4.0          INR     A                ;INCREMENT
1800  44E3  FE  08                 7.0          CPI     $8               ;DONE?
1801  44E5  CA  EC  44            10.0          JZ      TST01N           ;YES-GO CHECK ECC
1802  44E8  57                    4.0          MOV     D,A              ;NO-CONTINUE
1803  44E9  C3  85  44            10.0          JMP     TST01Y
1804  44EC  D3  0C                10.0  TST01N: OUT     RINST            ;SINGLE STEP THE READ PATH
1805  44EE  3A  C5  48            13.0          LDA     ECCCHR           ;GET THE EXPECTED ECC CHARACTER
1806  44F1                                      ROUT    EDATA            ;SAVE IN CAS
 (1)  44F1  D3  95                10.0                  OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  44F3  7F                    4.0                   MOV     A,A              ;RETRY LINK
1807  44F4  47                    4.0          MOV     B,A              ;SAVE IN REGISTER B
1808  44F5  DB  19                10.0          IN      ECCCOR
1809  44F7                                      ROUT    ADATA
 (1)  44F7  D3  94                10.0                  OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  44F9  7F                    4.0                   MOV     A,A              ;RETRY LINK
1810  44FA  B8                    4.0          CMP     B
1811  44FB  CA  03  45            10.0          JZ      TST01U
1812  44FE                                      ERR     TST01C,TST01U,1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44FE  CD  09  28            18.0                  CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)        0007                                        MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4501  07                                          .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  4502  01                                          .BYTE   1
 (1)  4503  CD  15  28            18.0          TST01U::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4506  DA  7F  43            10.0                  JC      TST01C           ;LOOP ADDRESS IF LOOP SPECIFIED
1813                                        ;>CORRECTED ECC CHARACTER INCORRECT
1814  4509  DB  1A                10.0          IN      ECCSTA           ;GET THE ECC STATUS
1815  450B  E6  0F                 7.0          ANI     @17              ;REMOVE UNWANTED BITS
1816  450D  FE  02                 7.0          CPI     E.TTEC           ;IS TWO TRACK CORRECT BIT SET?
1817  450F  CA  1F  45            10.0          JZ      TST01V
1818  4512                                      ROUT    ADATA
 (1)  4512  D3  94                10.0                  OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  4514  7F                    4.0                   MOV     A,A              ;RETRY LINK
1819  4515  3E  02                 7.0          MVI     A,E.TTEC
1820  4517                                      ROUT    EDATA
 (1)  4517  D3  95                10.0                  OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  4519  7F                    4.0                   MOV     A,A              ;RETRY LINK
1821  451A                                      ERRB    TST01C,TST01V,1
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  451A  CD  12  28            18.0                  CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)        0008                                        MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  451D  08                                          .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  451E  01                                          .BYTE   1                ;PRINT ROUTINE NUMBER
 (1)  451F  CD  15  28            18.0          TST01V::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4522  DA  7F  43            10.0                  JC      TST01C           ;LOOP ADDRESS IF LOOP SPECIFIED
1822                                      ;>ECC STATUS INCORRECT AFTER A DATA GROUP
1823  4525  3A  36  49            13.0          LDA     INDAT
1824  4528  3C                    4.0          INR     A
1825  4529  32  36  49            13.0          STA     INDAT
1826  452C  C2  7F  43            10.0          JNZ     TST01C
1827  452F                                      ENDTST  TST01X
 (1)                                          ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  452F                                      REQ     7                ;FAKE CALL TO KEEP TEST ALIVE
 (2)  452F  CD  06  28            18.0                  CALL    REQST
```

```
(2)  4532  00                                                  .BYTE                    ;DATA PATTERN NUMBER
(2)  4533  00  00                                              .WORD                    ;SYSTEM '"' COUNT
(2)  4535  00  00                                              .WORD                    ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
(2)  4537  00                                                  .BYTE                    ;DATA COMPARE FLAG IF =1
(2)  4538  07                                                  .BYTE   7                ;REQUEST CODE
(1)  4539  3A  9A  4F      13.0          LDA     ITERA         ;GET ITERATION COUNT
(1)  453C  3D                4.0          DCR     A             ;DOWNCOUNT
(1)  453D  32  9A  4F      13.0          STA     ITERA         ;SAVE COUNT
(1)  4540  F2  51  43      10.0          JP      TST01X        ;DO TEST UNTIL 'ILL = 0
```

```
1830
1831  4543                      .SBTTL  TEST 2 - ECC GOOD DATA THREE HARD POINTERS
 (1)                            ST
 (1)                            ;*****************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1832                            ;*ECC GOOD DATA THREE HARD POINTERS
1833  4543                      SD
 (1)                            ;*****************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1834                            ;*THIS TEST CHECKS THE ABILITY OF THE ECC TO DETECT THE UNCORRECTABLE
1835                            ;*SITUATION AND PASS THE DATA AS IS (I.E., NOT PERFORM ANY CORRECTION)
1836                            ;*TO THE CHECK CHARACTER LOGIC WITH THE UNCORRECTABLE STATUS BIT SET.
1837                            ;*THE PROGRAM LOADS CORRECT DATA TO THE FIFO AND JAMS THREE HARD POINTERS
1838                            ;*TO THE ECC BASED ON THE LAST DATA BYTE IN THE GROUP.  SEE TABLE.
1839                            ;*
1840                            ;*                                                  DATA BIT POSITION
1841                            ;*      BYTE COUNT              DATA                WITH HARD POINTERS
1842                            ;*      ----------              ----                ------------------
1843                            ;*
1844                            ;*         001          000,001,002,003,004,005,006        7,P,0
1845                            ;*         002          007,010,011,012,013,014,015        6,7,P
1846                            ;*         003          016,017,020,021,022,023,024        5,6,7
1847                            ;*         004          025,026,027,030,031,032,033        4,5,6
1848                            ;*         005          034,035,036,037,040,041,042        3,4,5
1849                            ;*         006          043,044,045,046,047,050,051        2,3,4
1850                            ;*         007          052,053,054,055,056,057,060        1,2,3
1851                            ;*         010          061,062,063,064,065,066,067        0,0,1
1852                            ;*         011          070,071,072,073,074,075,076        7,P,0
1853                            ;*         012          077,100,101,102,103,104,105        6,7,P
1854                            ;*         013          106,107,110,111,112,113,114        5,6,7
1855                            ;*         014          115,116,117,120,121,122,123        4,5,6
1856                            ;*         015          124,125,126,127,130,131,132        3,4,5
1857                            ;*         016          133,134,135,136,137,140,141        2,3,4
1858                            ;*         017          142,143,144,145,146,147,150        1,2,3
1859                            ;*         020          151,152,153,154,155,156,157        0,1,2
1860                            ;*         021          160,161,162,163,164,165,166        7,P,0
1861                            ;*         022          167,170,171,172,173,174,175        6,7,P
1862                            ;*         023          176,177,200,201,202,203,204        5,6,7
1863                            ;*         024          205,206,207,210,211,212,213        4,5,6
1864                            ;*         025          214,215,216,217,220,221,222        3,4,5
1865                            ;*         026          223,224,225,226,227,230,231        2,3,4
1866                            ;*         027          232,233,234,235,236,237,240        1,2,3
1867                            ;*         030          241,242,243,244,245,246,247        P,0,1
1868                            ;*         031          250,251,252,253,254,255,256        7,P,0
1869                            ;*         032          257,260,261,262,263,264,265        6,7,P
1870                            ;*         033          266,267,270,271,272,273,274        5,6,7
1871                            ;*         034          275,276,277,300,301,302,303        4,5,6
1872                            ;*         035          304,305,306,307,310,311,312        3,4,5
1873                            ;*         036          313,314,315,316,317,320,321        2,3,4
1874                            ;*         037          322,323,324,325,326,327,330        1,2,3
1875                            ;*         040          331,332,333,334,335,336,337        0,1,2
1876                            ;*         041          340,341,342,343,344,345,346        7,P,0
1877                            ;*         042          347,350,351,352,353,354,355        6,7,P
```

```
                        ;*          043          356,357,360,361,362,363,364          5,6,7
                        ;*          044          365,366,367,370,371,372,373          4,5,6
                        ;*          045          374,375,376,377,000,001,002          3,4,5
                        ;*
                        ;*THE POINTERS PROVIDED FOR THE TRACK IN ERROR ARE ALWAYS BAD 4X5.
4543                    SP
 (1)                    ; ***************************************************************************
 (1)                    ;*PROCEDURE
 (1)                    ;*---------
                        ;*BGNTST
                        ;*    SET NORMAL READ PATH CLOCK
                        ;*    CALL SUBROUTINE CLEAR
                        ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
                        ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
                        ;*    DELAY 50 MICROSECONDS
                        ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
                        ;*    CLEAR THE INPUT DATA
                        ;*    CLEAR BAD TRANSLATION POINTERS
                        ;*  BGNDO
                        ;*  : ENABLE THE READ PATH CLOCK
                        ;*  : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
                        ;*  : CLOCK THE FIFO'S
                        ;*  : ISSUE CLEAR ALL COMMAND
                        ;*  : WAIT
                        ;*  : STOP THE READ PATH, SET TIE BUS JAM AND SET PLO DISABLE
                        ;*  : CALL SUBROUTINE CLECC
                        ;*  : SET THE LOOP COUNT TO 7
                        ;*  : BGNDO
                        ;*  : :  GET THE INPUT DATA BYTE
                        ;*  : :  STORE IN BUFFER
                        ;*  : :  CALL SUBROUTINE ECC
                        ;*  : :  DECREMENT THE LOOP COUNT
                        ;*  : :  INCREMENT THE INPUT DATA
                        ;*  : :  DO UNTIL THE LOOP COUNT=0
                        ;*  : ENDDO
                        ;*  : SET THE LOOP COUNT TO 3
                        ;*  : BGNDO
                        ;*  : :  FILL THE TRANSLATOR SUBROUTINE BUFFER FROM THE DATA BUFFER
                        ;*  : :  CALL SUBROUTINE T4X5
                        ;*  : :  INIT THE LOOP COUNT TO 5
                        ;*  : :  BGNDO
                        ;*  : :  :  MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
                        ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
                        ;*  : :  :  MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
                        ;*  : :  :  CLOCK THE DATA INTO THE FIFO
                        ;*  : :  :  DECREMENT THE LOOP COUNT
                        ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
                        ;*  : :  :  DO UNTIL LOOP COUNT = 0
                        ;*  : :  ENDDO
                        ;*  : :  FILL THREE BYTES OF TRANSLATOR SUBROUTINE INPUT BUFFER FROM THE DATA BUFFER
                        ;*  : :  FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
                        ;*  : :  CALL SUBROUTINE T4X5
                        ;*  : :  INIT THE LOOP COUNT TO 5
                        ;*  : :  BGNDO
```

```
1929                    ;*  : : :  MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
1930                    ;*  : : :  MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
1931                    ;*  : : :  CLOCK THE DATA INTO THE FIFO
1932                    ;*  : : :  DECREMENT THE LOOP COUNT
1933                    ;*  : : :  DO UNTIL THE LOOP COUNT = 0
1934                    ;*  : : ENDDO
1935                    ;*  : : DECREMENT THE LOOP COUNT
1936                    ;*  : : DO UNTIL THE LOOP COUNT = 0
1937                    ;*  : ENDDO
1938                    ;*  : ISSUE DIAGNOSTIC READ COMMAND
1939                    ;*  : SET UP WATCHDOG TIMER COUNT
1940                    ;*  : BGNDO
1941                    ;*  : :    SINGLE STEP THE READ PATH
1942                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1943                    ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1944                    ;*  : ENDDO
1945                    ;*  : IF WATCHDOG TIMER=0
1946                    ;*  : :    THEN-ERROR
1947                    ;*  : :    ELSE-CONTINUE
1948                    ;*  : ENDIF
1949                    ;*  : GET THE TIE BUS TABLE ADDRESS
1950                    ;*  : SINGLE STEP THE READ PATH
1951                    ;*  : INIT THE LOOP COUNT TO 0
1952                    ;*  : CALL SUBROUTINE TRK
1953                    ;*  : BGNDO
1954                    ;*  : :    IF LOOP COUNT=TIE BUS JAM COUNT 1, 2 OR 3
1955                    ;*  : :    : THEN-LOAD TIE BUS WITH 10(8)
1956                    ;*  : :    : ELSE-CLEAR TIE BUS POINTER
1957                    ;*  : :    ENDIF
1958                    ;*  : :    SINGLE STEP THE READ PATH
1959                    ;*  : :    DECREMENT THE LOOP COUNT
1960                    ;*  : :    DO UNTIL LOOP COUNT = 9
1961                    ;*  : ENDDO
1962                    ;*  : SET UP WATCHDOG TIMER COUNT
1963                    ;*  : BGNDO
1964                    ;*  : :  SINGLE STEP THE READ PATH
1965                    ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
1966                    ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
1967                    ;*  : ENDDO
1968                    ;*  : IF WATCHDOG TIMER=0
1969                    ;*  : :  THEN-ERROR-EXIT TEST
1970                    ;*  : :  ELSE-CONTINUE
1971                    ;*  : ENDIF
1972                    ;*  : SINGLE STEP THE READ PATH
1973                    ;*  : INIT THE  LOOP COUNT TO 7
1974                    ;*  : BGNDO
1975                    ;*  : :  SINGLE STEP THE READ PATH
1976                    ;*  : :  COMPARE CORRECTED DATA WITH INPUT DATA
1977                    ;*  : :  IF NOT EQUAL
1978                    ;*  : :  : THEN-ECC ERROR
1979                    ;*  : :  : ELSE-CONTINUE
1980                    ;*  : :  ENDIF
1981                    ;*  : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
1982                    ;*  : :  IF NOT EQUAL
```

```
1983                                ;*  : :  :  THEN-ECC ERROR
1984                                ;*  : :  :  ELSE-CONTINUE
1985                                ;*  : :  ENDIF
1986                                ;*  : :  DECREMENT THE LOOP COUNT
1987                                ;*  : :  DO UNTIL THE LOOP COUNT = 0
1988                                ;*  : ENDDO
1989                                ;*  : SINGLE STEP THE READ PATH
1990                                ;*  : IF EXPECTED NOT EQUAL
1991                                ;*  : :  THEN-ECC ERROR
1992                                ;*  : :  ELSE-CONTINUE
1993                                ;*  : ENDIF
1994                                ;*  : INPUT THE ECC STATUS
1995                                ;*  : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
1996                                ;*  : IF THE RESULTING STATUS=SINGLE TRACK ERROR CORRECT + POINTER ERROR
1997                                ;*  : :  THEN-CONTINUE
1998                                ;*  : :  ELSE-ERROR
1999                                ;*  : ENDIF
2000                                ;*  : INCREMENT THE INPUT DATA
2001                                ;*  : DO UNTIL INPUT DATA = 0
2002                                ;*  ENDDO
2003                                ;*ENDTST
2004  4543                          SE
 (1)                                ; ************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
2005                                ;*ECC4 MICRO TEST 02
2006                                ;*ECC4 MICRO ERROR 11
2007                                ;*ECC4-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2008                                ;*M8951, M8950'S
2009                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
2010                                ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
2011                                ;*BYTE/SCLK COUNT NUMBER = LLL
2012                                ;*FATAL ERROR - MICRO TEST ABORTED
2013                                ;*
2014                                ;*ECC4 MICRO TEST 02
2015                                ;*ECC4 MICRO ERROR 12
2016                                ;*ECC4-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2017                                ;*M8951, M8950'S
2018                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
2019                                ;*SENT TO THE ECC
2020                                ;*BYTE/SCLK COUNT NUMBER = LLL
2021                                ;*FATAL ERROR - MICRO TEST ABORTED
2022                                ;*
2023                                ;*ECC4 MICRO TEST 02
2024                                ;*ECC4 MICRO ERROR 13
2025                                ;*ECC4-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2026                                ;*M8951, M8950'S
2027                                ;*DATA FROM THE ECC-INCORRECT
2028                                ;*DATA SHOULD NOT BE CHANGED
2029                                ;*BYTE/SCLK COUNT NUMBER = LLL
2030                                ;*ACTUAL = NNNN
2031                                ;*EXPECTED = NNNN
2032                                ;*
2033                                ;*ECC4 MICRO TEST 02
```

```
2034                                    ;*ECC4 MICRO ERROR 14
2035                                    ;*ECC4-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2036                                    ;*M8951, M8950'S
2037                                    ;*CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
2038                                    ;*PARITY SHOULD NOT BE CHANGED
2039                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2040                                    ;*
2041                                    ;*ECC4 MICRO TEST 02
2042                                    ;*ECC4 MICRO ERROR 15
2043                                    ;*ECC4-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2044                                    ;*M8951, M8950'S
2045                                    ;*CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
2046                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2047                                    ;*
2048                                    ;*ECC4 MICRO TEST 02
2049                                    ;*ECC4 MICRO ERROR 16
2050                                    ;*ECC4-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2051                                    ;*M8951, M8950'S
2052                                    ;*CORRECTED ECC CHARACTER INCORRECT
2053                                    ;*ECC CHARACTER SHOULD NOT BE CHANGED
2054                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2055                                    ;*
2056                                    ;*ECC4 MICRO TEST 02
2057                                    ;*ECC4 MICRO ERROR 17
2058                                    ;*ECC4-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2059                                    ;*M8951, M8950'S
2060                                    ;*ECC STATUS INCORRECT AFTER A DATA GROUP
2061                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2062                                    ;*ACTUAL = NNNN
2063                                    ;*EXPECTED = NNNN
2064  4543                              S
 (1)                                    ; ***************************************************************************
2065
2066  4543                              TEST2:  TESTX   @2                ;INITIALIZE THE TEST
 (1)  4543  3E  02          7.0                        MVI     A,@2              ;DEFINE THE TEST NUMBER
 (1)  4545  CD  03  28      18.0                       CALL    TSET              ;SETUP THE TEST
2067                                    ;%ECC4-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2068                                    ;&M8951, M8950'S
2069
2070  4548  3A  35  49      13.0              LDA     SUNIT
2071  454B  D3  E0          10.0              OUT     MBSEL
2072  454D  3E  10           7.0  TST02X:      MVI     A,RDCLK           ;SET NORMAL READ PATH CLOCKS
2073  454F  D3  F0          10.0              OUT     CLKCTL
2074  4551  3E  10           7.0              MVI     A,W.GCR
2075  4553  D3  D3          10.0              OUT     WMCCTL
2076  4555  3E  04           7.0              MVI     A,P.RPEN
2077  4557  D3  4C          10.0              OUT     PENAB
2078
2079  4559  3E  60           7.0              MVI     A,P.LWR!P.LCS     ;SET THE PORT CONTROL TO
2080  455B  D3  48          10.0              OUT     PDIAG             ;LCS, LWR
2081  455D  AF               4.0              XRA     A
2082  455E  D3  44          10.0              OUT     TAMT
2083  4560  3E  0A           7.0              MVI     A,10              ;DELAY
2084  4 ,2  3D               4.0  1$:          DCR     A                 ;50
```

```
2085  4563  C2  62  45   10.0          JNZ   1$                ;MICROSECONDS
2086  4566  AF               4.0          XRA   A                 ;CLEAR THE ACCUMULATOR
2087  4567  D3  D2          10.0          OUT   TRKENA            ;CLEAR ALL TRACKS FROM
2088  4569  D3  0A          10.0          OUT   RTIEB             ;CLEAR THE TIE BUS REGISTER
2089  456B  D3  D2          10.0          OUT   TRKENA
2090  456D  32  94  48      13.0          STA   BAD1              ;CLEAR THE INVALID
2091  4570  32  95  48      13.0          STA   BAD2              ;TRACK TRANSLATIONS
2092  4573  32  96  48      13.0          STA   BAD3              ;POINTERS
2093  4576  32  36  49      13.0          STA   INDAT             ;CLEAR THE INPUT DATA BYTE
2094  4579                                ROUT  R05H
 (1)  4579  D3  8B          10.0                OUT   R05H              ;WRITE AC INTO R05H
 (1)  457B  7F               4.0                MOV   A,A               ;RETRY LINK
2095
2096  457C  3E  A8           7.0  TST02C: MVI   A,R.PLO1!R.PLOD.R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
2097                                                                     ;SET TIE BUS JAM AND PLO DISABLE
2098  457E  D3  09          10.0          OUT   RPCTL
2099  4580  D3  08          10.0          OUT   RFIFOL            ;CLOCK THE FIFO'S
2100  4582  3E  0D           7.0          MVI   A,RCLRT           ;ISSUE CLEAR ALL COMMAND
2101  4584  D3  0B          10.0          OUT   RCMD
2102
2103  4586  00               4.0          NOP                     ;WAIT
2104  4587  00               4.0          NOP
2105  4588  00               4.0          NOP
2106  4589  00               4.0          NOP
2107  458A  00               4.0          NOP
2108
2109  458B  3E  A9           7.0          MVI   A,R.PLO1!R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
2110                                                                     ;SET TIE BUS JAM AND PLO DISABLE
2111  458D  D3  09          10.0          OUT   RPCTL
2112  458F  CD  97  48      18.0          CALL  CLECC             ;CLEAR THE ECC CHARACTER
2113  4592  21  3F  49      10.0          LXI   H,BUFFER          ;SET UP THE BUFFER POINTER
2114  4595  16  07           7.0          MVI   D,@7              ;SET UP THE LOOP COUNT
2115  4597  3A  36  49      13.0  TST02F: LDA   INDAT             ;GET THE INPUT DATA
2116  459A  77               7.0          MOV   M,A               ;SAVE CHARACTER IN THE BUFFER
2117  459B  3C               4.0          INR   A                 ;UPDATE THE DATA
2118  459C  32  36  49      13.0          STA   INDAT             ;SAVE IT
2119  459F  23               6.0          INX   H                 ;UPDATE THE BUFFER POINTER
2120  45A0  3D               4.0          DCR   A                 ;RECREATE THE DATA FOR THE ECC SUBROUTINE
2121  45A1  CD  9C  48      18.0          CALL  ECC               ;CALCULATE ECC FOR
2122  45A4  15               4.0          DCR   D                 ;THE 7 INPUT BYTES
2123  45A5  C2  97  45      10.0          JNZ   TST02F
2124
2125  45A8  0E  03           7.0          MVI   C,@3              ;INIT THE LOOP COUNT
2126  45AA  3A  36  49      13.0  TST02I: LDA   INDAT             ;GET THE INPUT DATA
2127  45AD  06  25           7.0          MVI   B,@45             ;CHECK FOR SPECIAL CONDITION
2128  45AF  FE  03           7.0          CPI   @3                ;
2129  45B1  CA  BD  45      10.0          JZ    TST02Z            ;
2130  45B4  06  00           7.0          MVI   B,0               ;INITIALIZE THE QUOTION
2131  45B6  04               4.0  TST02W: INR   B                 ;
2132  45B7  D6  07           7.0          SUI   @7                ;DIVIDE BY 7
2133  45B9  A7               4.0          ANA   A                 ;SET THE CONDITION BITS
2134  45BA  C2  B6  45       0.0          JNZ   TST02W            ;
2135  45BD  78               4.0  TST02Z: MOV   A,B
2136  45BE                                ROUT  R05L              ;SET THE COUNTER IN THE CAS
```

```
 (1)  45BE  D3  8A        10.0                OUT     R05L          ;WRITE AC INTO R05L
 (1)  45C0  7F             4.0                MOV     A,A           ;RETRY LINK
2137  45C1  3A  3F  49     13.0          LDA   BUFFER
2138  45C4  32  84  48     13.0          STA   TRNIN             ;FILL THE TRANSLATOR
2139  45C7  3A  40  49     13.0          LDA   BUFFER+1
2140  45CA  32  85  48     13.0          STA   TRNIN+1           ;SUBROUTINE INPUT
2141  45CD  3A  41  49     13.0          LDA   BUFFER+2
2142  45D0  32  86  48     13.0          STA   TRNIN+2           ;BUFFER
2143  45D3  3A  42  49     13.0          LDA   BUFFER+3
2144  45D6  32  87  48     13.0          STA   TRNIN+3           ;
2145  45D9  CD  B1  47     18.0          CALL  T4X5              ;TRANSLATE THE SUBGROUP
2146
2147  45DC  21  88  48     10.0          LXI   H,TRNOUT          ;GET POINTER TO TRANSLATED DATA TABLE
2148  45DF  06  05          7.0          MVI   B,@5              ;SET UP LOOP COUNT
2149  45E1  7E              7.0  TST02G: MOV   A,M               ;GET A DATA BYTE
2150  45E2  D3  40         10.0          OUT   TCMD              ;STORE DATA IN COMMAND ADDRESS
2151  45E4  23              6.0          INX   H                 ;POINT TO DATA PARITY
2152  45E5  7E              7.0          MOV   A,M               ;GET THE DATA PARITY
2153
2154  45E6  07              4.0          RLC                     ;POSITION FOR OUTPUT
2155  45E7  F6  60          7.0          ORI   P.LWR!P.LCS       ;OR IN CONTROL BITS
2156  45E9  D3  48         10.0          OUT   PDIAG             ;OUTPUT THE DATA PARITY
2157
2158  45EB  D3  08         10.0          OUT   RFIFOL            ;CLOCK DATA INTO THE FIFO'S
2159
2160  45ED  23              6.0          INX   H                 ;UPDATE THE TABLE POINTER
2161  45EE  05              4.0          DCR   B                 ;DECREMENT LOOP COUNT
2162  45EF  C2  E1  45     10.0          JNZ   TST02G            ;DO UNTIL LOOP COUNT = 0
2163
2164  45F2  3A  43  49     13.0          LDA   BUFFER+4
2165  45F5  32  84  48     13.0          STA   TRNIN             ;FILL THE FIRST THREE
2166  45F8  3A  44  49     13.0          LDA   BUFFER+5
2167  45FB  32  85  48     13.0          STA   TRNIN+1           ;BYTES OF THE TRANSLATOR
2168  45FE  3A  45  49     13.0          LDA   BUFFER+6
2169  4601  32  86  48     13.0          STA   TRNIN+2           ;SUBROUTINE WITH THE INPUT DATA
2170  4604  3A  C5  48     13.0          LDA   ECCCHR            ;STORE THE CALCULATED
2171  4607  32  87  48     13.0          STA   TRNIN+3           ;ECC CHARACTER AS THE LAST CHARACTER
2172
2173  460A  CD  B1  47     18.0          CALL  T4X5              ;TRANSLATE THE SECOND SUBGROUP
2174
2175  460D  21  88  48     10.0          LXI   H,TRNOUT          ;GET THE POINTER TO INPUT DATA TABLE
2176  4610  06  05          7.0          MVI   B,@5              ;SET UP THE LOOP COUNT
2177  4612  7E              7.0  TST02H: MOV   A,M               ;GET A DATA BYTE
2178  4613  D3  40         10.0          OUT   TCMD              ;STORE DATA IN CMD ADDRESS
2179  4615  23              6.0          INX   H                 ;POINT TO DATA PARITY BIT
2180  4616  7E              7.0          MOV   A,M               ;GET THE DATA PARITY
2181  4617  07              4.0          RLC                     ;POSITION FOR OUTPUT
2182  4618  F6  60          7.0          ORI   P.LWR!P.LCS       ;OR IN CONTROL BITS
2183  461A  D3  48         10.0          OUT   PDIAG             ;OUTPUT THE DATA PARITY
2184  461C  D3  08         10.0          OUT   RFIFOL            ;CLOCK DATA INTO THE FIFOS
2185  461E  23              6.0          INX   H                 ;UPDATE THE TABLE POINTER
2186  461F  05              4.0          DCR   B                 ;DECREMENT THE LOOP COUNT
2187  4620  C2  12  46     10.0          JNZ   TST02H            ;DO UNTIL LOOP COUNT=0
2188
```

```
2189  4623  0D                    4.0          DCR     C               ;PUT THE GROUP IN
2190  4624  C2  AA  45            10.0         JNZ     TST02I          ;THE FIFO TWICE
2191
2192  4627  3E  0B                7.0          MVI     A,DIARD         ;LOAD THE DIAGNOSTIC READ
2193  4629  D3  0B                10.0         OUT     RCMD            ;COMMAND
2194  462B  11  01  00            10.0         LXI     D,1             ;SET WATCH DOG INCREMENT
2195  462E  21  A8  FD            10.0         LXI     H,-600          ;SET WATCH DOG COUNT TO 600
2196  4631  D3  0C                10.0  2$:    OUT     RINST           ;STEP THE READ PATH
2197  4633  DB  01                10.0         IN      RPCHI           ;DATA READY SET?
2198  4635  E6  10                7.0          ANI     R.DRDY
2199  4637  C2  4C  46            10.0         JNZ     TST02A          ;YES-GO PROCESS
2200  463A  19                    10.0         DAD     D               ;WATCH DOG TIMEOUT?
2201  463B  D2  31  46            10.0         JNC     2$              ;NO-CONTINUE
2202  463E                                     ERR     RETRY,DUMMU,1
 (1)                                         ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  463E  CD  09  28            18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0009                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4641  09                               .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4642  01                               .BYTE   1
 (1)  4643  CD  15  28            18.0       DUMMU:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4646  DA  77  47            10.0               JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2203                                     ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
2204                                     ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
2205                                     ;<FATAL ERROR - MICRO TEST ABORTED
2206  4649  C3  74  47            10.0         JMP     EXIT
2207
2208  464C  CD  D8  48            18.0  TST02A: CALL    TRK             ;SET UP HARD POINTER TRACK NUMBERS
2209  464F  D3  0C                10.0         OUT     RINST
2210  4651  06  00                7.0          MVI     B,$0            ;INIT THE BIT TIME COUNT (LOOP COUNT)
2211  4653  00                    4.0   TST02B: NOP
2212  4654  3A  32  49            13.0         LDA     TIET1           ;GET FIRST TIE BUS JAM TIME
2213  4657  B8                    4.0          CMP     B               ;IS IT TIME TO JAM THE TIE BUS?
2214  4658  C2  62  46            10.0         JNZ     4$              ;NO - CONTINUE
2215  465B  3E  08                7.0          MVI     A,$8            ;LOAD ILLEGAL
2216  465D  D3  0A                10.0         OUT     RTIEB           ;5X4 POINTER
2217  465F  C3  81  46            10.0         JMP     5$              ;
2218  4662  3A  33  49            13.0  4$:    LDA     TIET2           ;GET SECOND TIE BUS JAM TIME
2219  4665  B8                    4.0          CMP     B               ;IS IT TIME TO JAM THE TIE BUS?
2220  4666  C2  70  46            10.0         JNZ     2$              ;NO-CONTINUE
2221  4669  3E  08                7.0          MVI     A,$8            ;LOAD ILLEGAL
2222  466B  D3  0A                10.0         OUT     RTIEB           ;5X4 POINTER
2223  466D  C3  81  46            10.0         JMP     5$              ;
2224  4670  3A  34  49            13.0  2$:    LDA     TIET3           ;GET THIRD TIE BUS JAM TIME
2225  4673  B8                    4.0          CMP     B               ;IS IT TIME TO JAM THE BUS?
2226  4674  C2  7E  46            10.0         JNZ     6$              ;NO-CONTINUE
2227  4677  3E  08                7.0          MVI     A,$8            ;LOAD ILLEGAL
2228  4679  D3  0A                10.0         OUT     RTIEB           ;5X4 POINTER
2229  467B  C3  81  46            10.0         JMP     5$              ;CONTINUE
2230  467E  AF                    4.0   6$:    XRA     A               ;IF NOT THE BAD TRACK, PROVIDE A GOOD POINTER
2231  467F  D3  0A                10.0         OUT     RTIEB           ;LOAD THE POINTER
2232  4681  D3  0C                10.0  5$:    OUT     RINST           ;STEP CLOCK
2233  4683  04                    4.0          INR     B               ;DECREMENT THE LOOP COUNT
2234  4684  78                    4.0          MOV     A,B
2235  4685  FE  09                7.0          CPI     $9              ;DONE
```

```
2236  4687  C2  53  46   10.0          JNZ   TST02B          ;DO UNTIL LOOP COUNT = ZERO
2237  468A  AF             4.0          XRA   A
2238  468B  D3  0A        10.0          OUT   RTIEB           ;CLEAR THE TIE BUS REGISTER
2239  468D  D3  0C        10.0          OUT   RINST           ;CLOCK THE READ PATH
2240  468F  D3  0C        10.0          OUT   RINST           ;CLOCK THE READ PATH
2241  4691  11  01  00    10.0          LXI   D,1             ;SET WATCHDOG TIMER INCREMENT
2242  4694  21  A8  FD    10.0          LXI   H,-600          ;SET WATCHDOG COUNT TO 600
2243  4697  D3  0C        10.0  3$:     OUT   RINST           ;SINGLE STEP THE READ PATH
2244  4699  DB  01        10.0          IN    RPCHI           ;DATA READY SET?
2245  469B  E6  10         7.0          ANI   R.DRDY
2246  469D  C2  B2  46    10.0          JNZ   TST02K          ;YES-GO PROCESS
2247  46A0  19            10.0          DAD   D               ;NO-WATCHDOG TIMEOUT?
2248  46A1  D2  97  46    10.0          JNC   3$              ;NO-CONTINUE
2249  46A4                               ERR   RETRY,DUMMV,1
 (1)                                   ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  46A4  CD  09  28    18.0                CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000A                             MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46A7  0A                              .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  46A8  01                              .BYTE 1
 (1)  46A9  CD  15  28    18.0          DUMMV:: CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  46AC  DA  77  47    10.0                JC    RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2250                                   ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
2251                                   ;>SENT TO THE ECC
2252                                   ;<FATAL ERROR - MICRO TEST ABORTED
2253  46AF  C3  74  47    10.0          JMP   EXIT
2254
2255  46B2  16  01         7.0  TST02K: MVI   D,@1            ;INIT THE LOOP COUNT
2256  46B4  21  3F  49    10.0          LXI   H,BUFFER
2257  46B7  00             4.0  TST02Y: NOP
2258  46B8  7E             7.0          MOV   A,M             ;GET THE INPUT DATA
2259  46B9                               ROUT  EDATA
 (1)  46B9  D3  95        10.0                OUT   EDATA           ;WRITE AC INTO EDATA
 (1)  46BB  7F             4.0                MOV   A,A             ;RETRY LINK
2260  46BC  A7             4.0          ANA   A               ;SET CONDITION BITS
2261  46BD  3E  00          7.0          MVI   A,0             ;CLEAR THE ACC.
2262  46BF  E2  C3  46    10.0          JPO   1$              ;ODD # OF ONES - CONTINUE
2263  46C2  3C             4.0          INR   A               ;EVEN # OF ONES - SET PARITY
2264  46C3  32  37  49    13.0  1$:     STA   INDATP          ;STORE THE PARITY BIT
2265  46C6  D3  0C        10.0  TST02L: OUT   RINST           ;SINGLE STEP THE READ LOGIC
2266  46C8  7E             7.0          MOV   A,M
2267  46C9  23             6.0          INX   H
2268  46CA  47             4.0          MOV   B,A
2269  46CB  DB  19        10.0          IN    ECCCOR
2270  46CD  B8             4.0          CMP   B
2271  46CE  CA  D9  46    10.0          JZ    TST02M
2272  46D1                               ROUT  ADATA
 (1)  46D1  D3  94        10.0                OUT   ADATA           ;WRITE AC INTO ADATA
 (1)  46D3  7F             4.0                MOV   A,A             ;RETRY LINK
```

```
2274  46D4                                        ERRB    RETRY,TST02M,@1
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46D4   CD   12   28        18.0                    CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000B                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46D7   0B                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46D8   01                                          .BYTE   @1              ;PRINT ROUTINE NUMBER
 (1)  46D9   CD   15   28        18.0            TST02M.:        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46DC   DA   77   47        10.0                    JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2275                                        ;>DATA FROM THE ECC-INCORRECT
2276                                        ;>DATA SHOULD NOT BE CHANGED
2277  46DF   DB   15                10.0          IN      RPSTA           ;GET THE ECC PARITY BIT
2278  46E1   E6   80                 7.0          ANI     E.CDP           ;GET THE CORRECTED PARITY BIT
2279  46E3   07                     4.0          RLC                     ;POSITION BIT FOR COMPARE
2280  46E4   4F                     4.0          MOV     C,A             ;SAVE IN REG. C
2281  46E5   3A   37   49           13.0          LDA     INDATP          ;GET THE EXPECTED PARITY BIT
2282  46E8   A7                     4.0          ANA     A               ;SET THE CONDITION BITS
2283  46E9   CA   00   47           10.0          JZ      TST02QO         ;GO CHECK FOR PARITY=0
2284  46EC   79                     4.0          MOV     A,C             ;ELSE CHECK FOR PARITY=1
2285  46ED   FE   01                7.0          CPI     @1
2286  46EF   CA   F7   46           10.0          JZ      TST02RG
2287  46F2                                       ERR     RETRY,TST02RG,@1
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  46F2   CD   09   28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000C                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46F5   0C                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  46F6   01                                          .BYTE   @1
 (1)  46F7   CD   15   28        18.0            TST02RG::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  46FA   DA   77   47        10.0                    JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2288                                        ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
2289                                        ;>PARITY SHOULD NOT BE CHANGED
2290                                        ;<
2291  46FD   C3   14   47        10.0          JMP     TST02SZ         ;CONTINUE WITH TEST
2292  4700   79                     4.0  TST02QO: MOV    A,C             ;CHECK FOR PARITY = ZERO
2293  4701   A7                     4.0          ANA     A               ;SET CONDITION BITS
2294  4702   CA   0E   47           10.0          JZ      TST02TP         ;CONTINUE IF ZERO
2295  4705   78                     4.0          MOV     A,B
2296  4706                                       ROUT    R05L
 (1)  4706   D3   8A                10.0                    OUT     R05L            ;WRITE AC INTO R05L
 (1)  4708   7F                     4.0                    MOV     A,A             ;RETRY LINK
2297  4709                                       ERR     RETRY,TST02TP,@1
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4709   CD   09   28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000D                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  470C   0D                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  470D   01                                          .BYTE   @1
 (1)  470E   CD   15   28        18.0            TST02TP::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4711   DA   77   47        10.0                    JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2298                                        ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
2299                                        ;<
2300  4714   7A                     4.0  TST02SZ: MOV    A,D             ;GET THE CHARACTER COUNT
2301  4715   3C                     4.0          INR     A               ;INCREMENT
2302  4716   FE   08                7.0          CPI     $8              ;DONE?
2303  4718   CA   1F   47           10.0          JZ      TST02N          ;YES-GO CHECK ECC
2304  471B   57                     4.0          MOV     D,A             ;NO-CONTINUE
```

```
2305  471C  C3  B7  46      10.0          JMP     TST02Y
2306
2307  471F  D3  0C          10.0  TST02N: OUT     RINST           ;SINGLE STEP THE READ PATH
2308  4721  3A  C5  48      13.0          LDA     ECCCHR          ;GET THE EXPECTED ECC CHARACTER
2309  4724                                ROUT    EDATA           ;SAVE IN CAS
 (1)  4724  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4726  7F              4.0                   MOV     A,A             ;RETRY LINK
2310  4727  47              4.0           MOV     B,A             ;SAVE IN REGISTER B
2311  4728  DB  19          10.0          IN      ECCCOR
2312  472A                                ROUT    ADATA
 (1)  472A  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  472C  7F              4.0                   MOV     A,A             ;RETRY LINK
2313  472D  B8              4.0           CMP     B
2314  472E  CA  36  47      10.0          JZ      TST02U
2315  4731                                ERR     RETRY,TST02U,1
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4731  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       000E                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4734  0E                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4735  01                                            .BYTE   1
 (1)  4736  CD  15  28      18.0                  TST02U::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4739  DA  77  47      10.0                          JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2316                                    ;>CORRECTED ECC CHARACTER INCORRECT
2317                                    ;>ECC CHARACTER SHOULD NOT BE CHANGED
2318  473C  DB  1A          10.0                  IN      ECCSTA          ;GET THE ECC STATUS
2319  473E  E6  0F              7.0               ANI     @17             ;REMOVE UNWANTED BITS
2320  4740  FE  04              7.0               CPI     E.UNC           ;IS UNCORRECTABLE BIT SET?
2321  4742  CA  52  47      10.0                  JZ      TST02V
2322  4745                                        ROUT    ADATA
 (1)  4745  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4747  7F              4.0                   MOV     A,A             ;RETRY LINK
2323  4748  3E  04              7.0               MVI     A,E.UNC
2324  474A                                        ROUT    EDATA
 (1)  474A  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  474C  7F              4.0                   MOV     A,A             ;RETRY LINK
2325  474D                                        ERRB    RETRY,TST02V,1
 (1)                                            ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  474D  CD  12  28      18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000F                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4750  0F                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4751  01                                            .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4752  CD  15  28      18.0                  TST02V::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4755  DA  77  47      10.0                          JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2326                                    ;>ECC STATUS INCORRECT AFTER A DATA GROUP
2327  4758  3A  36  49      13.0          LDA     INDAT
2328  475B  FE  03              7.0               CPI     @3              ;DONE
2329  475D  C2  7C  45      10.0          JNZ     TST02C
```

```
2331  4760                                        ENDTST  TST02X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4760                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4760  CD  06  28        18.0                                  CALL    FREQST
 (2)  4763  00                                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  4764  00  00                                                  .WORD                   ;SYSTEM '"' COUNT
 (2)  4766  00  00                                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4768  00                                                      .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4769  07                                                      .BYTE   7               ;REQUEST CODE
 (1)  476A  3A  9A  4F        13.0                  LDA     ITERA           ;GET ITERATION COUNT
 (1)  476D  3D                 4.0                  DCR     A               ;DOWNCOUNT
 (1)  476E  32  9A  4F        13.0                  STA     ITERA           ;SAVE COUNT
 (1)  4771  F2  4D  45        10.0                  JP      TST02X          ;DO TEST UNTIL TILL = 0
2332
2333  4774  C3  18  28        10.0  EXIT:   JMP     TSTEND
2334  4777  3A  36  49        13.0  RETRY:  LDA     INDAT
2335  477A  D6  07              7.0                  SUI     @7
2336  477C  32  36  49        13.0                  STA     INDAT
2337  477F  C3  7C  45        10.0                  JMP     TST02C
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ;*****************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ;*****************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ;*****************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*    CLEAR PORT SELECT FOR TRANSPORT
                                        ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*    CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ;*****************************************************************************
4782  F5                    12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
4783  C5                    12.0          PUSH    B
4784  06  00                 7.0           MVI     B,0             ;START TO CLEAR AT PORT #0
4786  DB  E0                10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
4788  E6  80                7.0           ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
478A  B0                    4.0           ORA     B               ;ADD IN THE SELECTED PORT #
478B  D3  E0                10.0          OUT     MBSEL           ;RESET TO THIS PORT
478D  3E  80                7.0           MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
478F  D3  40                10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
4791  AF                    4.0           XRA     A               ;CLEAR TU COMMAND A
4792  D3  40                10.0          OUT     TCMD            ;
4794  3E  81                7.0           MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
4796  D3  40                10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
4798  3E  00                7.0           MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
479A  D3  40                10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
479C  AF                    4.0           XRA     A
479D  D3  44                10.0          OUT     TAMT            ;CLEAR AMTIE WORD
479F  D3  48                10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
47A1  D3  4C                10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
47A3  04                    4.0           INR     B               ;POINT TO THE NEXT PORT TO CLEAR
47A4  78                    4.0           MOV     A,B
47A5  FE  04                7.0           CPI     4               ;DONE?
47A7  C2  86  47            10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
47AA  C1                    10.0          POP     B               ;RESET B & C
47AB  F1                    10.0          POP     PSW             ;ALL DONE
47AC  C9                    10.0          RET                     ;EXIT
```

```
                                .SBTTL  SUBROUTINE 4X5 TRANSLATE A SUBGROUP
47AD                            SSUB
                                ; ***************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*4X5 TRANSLATE A SUBGROUP
47AD                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO 4X5 TRANSLATE A 4 BYTE DATA SUBGROUP INTO
                                ;*A 5 BYTE SUBGROUP AS RECORDED ON TAPE OR READ FROM TAPE. AT THE ENTRY
                                ;*T4X5.  AT ENTRY BADT4X5 THE VARIABLES BAD 1, 2, 3 CAN CONTAIN NUMBERS
                                ;*FROM 1-9 TO INDICATE TRACKS THAT WILL BE TRANSLATED TO A VALID VALUE,
                                ;*BUT NOT CORRECT FOR THE INPUT DATA SIMULATING BAD DATA.
47AD                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*--- -----
                                ;*BGNSUB
                                ;*    CLEAR THE BAD TRANSLATION POINTERS
                                ;*    GET THE 4 (INPUT) DATA BYTES TO BE TRANSLATED
                                ;*    CLEAR THE TRANSLATED (OUTPUT) BUFFER (OUTPUT BUFFER IS 9X5)
                                ;*    SET LOOP COUNT TO 1
                                ;*    BGNDO
                                ;*    :    CALCULATE ODD PARITY FOR THE FOUR (INPUT) DATA BYTES (INPUT BUFFER IS NOW 9X4
                                ;*    :    ROTATE EACH (5) ENTRY OF THE TRANSLATED (OUTPUT) BUFFER RIGHT
                                ;*    :    GET THE LEAST SIGNIFICANT BIT OF EACH (INPUT) DATA BYTE AND
                                ;*    :        FORM A TABLE INDEX (MS BIT OF INDEX = LS BIT OF THE FIRST
                                ;*    :        BYTE OF THE INPUT DATA, ETC.)
                                ;*    :    REMOVE THE LS BIT OF EACH INPUT DATA BYTE (THE ONE JUST USED
                                ;*    :        TO BUILD THE INDEX)
                                ;*    :    USE THE 4 BIT INDEX TO GET THE TRANSLATION VALUE
                                ;*    :    PUT ONE BIT OF THE TRANSLATION VALUE, AS THE MOST SIGNIFICANT
                                ;*    :        BIT EACH OF THE 5 TRANSLATED DATA BYTES (LS BIT OF THE
                                ;*    :        TRANSLATION VALUE REPLACES MS BIT OF THE FIRST TRANSLATED
                                ;*    :        DATA BYTE.
                                ;*    :    INCREMENT THE LOOP COUNT
                                ;*    :    DO UNTIL THE LOOP COUNT = 10
                                ;*    ENDDO
                                ;*ENDSUB
47AD                            S
                                ; ***************************************************************************

47AD  F5              12.0  BADT4X5: PUSH   PSW             ;SAVE REGISTER A + PSW
47AF  C3  BC  47      10.0           JMP    BAD
47B1  F5              12.0  T4X5:    PUSH   PSW             ;SAVE REGISTER A + PSW
47B2  AF               4.0           XRA    A               ;CLEAR A
47B3  32  94  48      13.0           STA    BAD1            ;CLEAR BAD TRANSLATION 1
47B6  32  95  48      13.0           STA    BAD2            ;CLEAR BAD TRANSLATION 2
47B9  32  96  48      13.0           STA    BAD3            ;CLEAR BAD TRANSLATION 3
47BC  C5              12.0  BAD:     PUSH   B               ;SAVE REGISTER B + C
47BD  D5              12.0           PUSH   D               ;SAVE REGISTER D + E
47BE  E5              12.0           PUSH   H               ;SAVE REGISTER H + L
```

```
2427                                        ;FOUR DATA BYTES ARE NOW IS THE BUFFER TABLE
2428                                        ;CLEAR THE TRANSLATED DATA TABLE.
2429  47BF  06  0A          7.0            MVI   B,10            ;SET UP LOOP COUNT
2430  47C1  11  88  48      10.0           LXI   D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
2431  47C4  AF               4.0            XRA   A              ;CLEAR A
2432  47C5  12               7.0  D4X5:     STAX  D              ;STORE IN THE TABLE
2433  47C6  13               6.0            INX   D              ;UPDATE TABLE POINTER
2434  47C7  05               4.0            DCR   B              ;DECREMENT LOOP COUNT
2435  47C8  C2  C5  47       10.0           JNZ   D4X5           ;DO UNTIL LOOP COUNT=0
2436  47CB  0E  01           7.0            MVI   C,1            ;SET UP TRACK COUNT
2437  47CD  06  04           7.0  B4X5:     MVI   B,4            ;SET UP BIT COUNT
2438  47CF  11  84  48       10.0           LXI   D,TRNIN
2439  47D2  AF               4.0            XRA   A              ;CLEAR THE GROUP POSITION COUNT
2440  47D3  32  92  48       13.0           STA   GP4X5
2441  47D6  1A               7.0  C4X5:     LDAX  D              ;GET A DATA BYTE
2442  47D7  A7               4.0            ANA   A              ;SET CONDITION BITS
2443  47D8  E2  DC  47       10.0           JPO   PO4X5          ;ODD PARITY LEAVE CARRY CLEAR
2444  47DB  37               4.0            STC                  ;EVEN PARITY SET CARRY
2445  47DC  1F               4.0  PO4X5:     RAR                  ;SHIFT OUT DESIRED BIT
2446  47DD  12               7.0            STAX  D              ;STORE RESULT BACK IN TEMP TABLE
2447  47DE  3A  92  48       13.0           LDA   GP4X5          ;GET THE GROUP POSITION BYTE
2448  47E1  17               4.0            RAL                  ;PUT IN THIS DATA BIT
2449  47E2  32  92  48       13.0           STA   GP4X5          ;SAVE THE UPDATED GROUP POSITION BYTE
2450  47E5  13               6.0            INX   D              ;UPDATE THE TABLE POINTER
2451  47E6  05               4.0            DCR   B              ;DECREMENT THE BIT COUNT
2452  47E7  C2  D6  47       10.0           JNZ   C4X5           ;DO UNTIL ALL 4 BITS ARE RECEIVED
2453
2454                                        ;AT THIS POINT GP4X5 CONTAINS 4 BITS - ONE FROM EACH OF THE FOUR BYTES
2455
2456                                        ,PERFORM A DOUBLE PRECISION SHIFT RIGHT ON  4E TRANSLATED DATA
2457  47EA  06  05           7.0            MVI   B,5            ;SET UP LOOP COUNT
2458  47EC  11  88  48       10.0           LXI   D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
2459  47EF  13               6.0  E4X5:     INX   D              ;POINT TO PARITY BIT
2460  47F0  1A               7.0            LDAX  D              ;GET PARITY BIT
2461  47F1  A7               4.0            ANA   A              ;CLEAR THE CARRY BIT
2462  47F2  1F               4.0            RAR                  ;SHIFT TO THE CARRY BIT
2463  47F3  12               7.0            STAX  D              ;STORE IT BACK
2464  47F4  1B               6.0            DCX   D              ;DECREMENT TO DATA BITS
2465  47F5  1A               7.0            LDAX  D              ;GET THE DATA BITS
2466  47F6  1F               4.0            RAR                  ;SHIFT IN CARRY BITS
2467  47F7  12               7.0            STAX  D              ;STORE BACK IN TABLE
2468  47F8  13               6.0            INX   D              ;POINT TO NEXT TABLE ENTRY
2469  47F9  13               6.0            INX   D              ;
2470  47FA  05               4.0            DCR   B              ;DECREMENT LOOP COUNT
2471  47FB  C2  EF  47       10.0           JNZ   E4X5           ;DO UNTIL LOOP COUNT=0
2472  47FE  3A  94  48       13.0           LDA   BAD1           ;IS A BAD POINTER SPECIFIED?
2473  4801  A7               4.0            ANA   A              ;SET CONDITION BITS
2474  4802  CA  09  48       10.0           JZ    BAD1C          ;NO-CONTINUE
2475  4805  B9               4.0            CMP   C              ;YES-IS IT THIS DATA POSITION?
2476  4806  CA  1F  48       10.0           JZ    INV4X5         ;YES-GO PERFORM INCORRECT TRANSLATION
2477  4809  3A  95  48       13.0 BAD1C:    LDA   BAD2           ;IS A BAD POINTER SPECIFIED
2478  480C  A7               4.0            ANA   A              ;SET CONDITION BITS
2479  480D  CA  14  48       10.0           JZ    BAD2C          ;NO-CONTINUE
2480  4810  B9               4.0            CMP   C              ;YES-IS IT THIS DATA POSITION
```

```
2481  4811  CA  1F  48   10.0          JZ    INV4X5          ;YES-GO PERFORM INCORRECT TRANSLATION
2482  4814  3A  96  48   13.0  BAD2C:  LDA   BAD3            ;IS A BAD POINTER SPECIFIED
2483  4817  A7               4.0          ANA   A               ;SET CONDITION BITS
2484  4818  CA  26  48   10.0          JZ    BAD3C           ;NO-PERFORM PROPER TRANSLATION
2485  481B  B9               4.0          CMP   C               ;YES-IS IT THIS DATA POSITION
2486  481C  C2  26  48   10.0          JNZ   BAD3C           ;NO-GO PERFORM PROPER TRANSLATION
2487  481F  3A  92  48   13.0  INV4X5: LDA   GP4X5           ;GET THE GROUP POSITION COUNT
2488  4822  3C               4.0          INR   A               ;INCREMENT-TO INCORRECT TRANSLATION
2489  4823  32  92  48   13.0          STA   GP4X5           ;STORE IT
2490  4826  11  74  48   10.0  BAD3C:  LXI   D,TAB4X5        ;LOAD ADDRESS OF TRANSLATION TABLE
2491  4829  26  00            7.0          MVI   H,0             ;CLEAR REGISTER H
2492  482B  3A  92  48   13.0          LDA   GP4X5           ;GET GROUP POSITION COUNT
2493  482E  E6  0F            7.0          ANI   $0F
2494  4830  6F               4.0          MOV   L,A
2495  4831  19              10.0          DAD   D               ;ADD GROUP POSITION COUNT TO TABLE
2496  4832  7E               7.0          MOV   A,M             ;GET TRANSLATED DATA
2497  4833  32  93  48   13.0          STA   TRNTMP
2498                                ;"TRNTMP" = THE TRANSLATED DATA
2499
2500  4836  11  88  48   10.0          LXI   D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
2501  4839  06  05            7.0          MVI   B,5             ;SET UP LOOP COUNT
2502  483B  3A  93  48   13.0  F4X5:   LDA   TRNTMP          ;GET TRANSLATED DATA
2503  483E  13               6.0          INX   D               ;POINT TO PARITY POSITION
2504  483F  A7               4.0          ANA   A               ;CLEAR CARRY BIT
2505  4840  1F               4.0          RAR                   ;SHIFT OUT A BIT
2506  4841  32  93  48   13.0          STA   TRNTMP          ;STORE TRANSLATED DATA
2507  4844  1A               7.0          LDAX  D               ;GET THE PARITY BIT (ALWAYS ZERO)
2508  4845  17               4.0          RAL                   ;ROLL IN THE CARRY BIT
2509  4846  12               7.0          STAX  D               ;STORE AS THE PARITY BIT
2510  4847  13               6.0          INX   D               ;UPDATE TABLE POINTER
2511  4848  05               4.0          DCR   B               ;DECREMENT THE LOOP COUNT
2512  4849  C2  3B  48   10.0          JNZ   F4X5            ;DO UNTIL LOOP COUNT=0
2513
2514                          ;NOW THE TRANSLATED DATA TABLE HAS BEEN UPDATED
2515
2516  484C  0C               4.0          INR   C               ;DECREMENT THE TRACK COUNT
2517  484D  79               4.0          MOV   A,C
2518  484E  FE  0A            7.0          CPI   10
2519  4850  C2  CD  47   10.0          JNZ   B4X5            ;DO UNTIL ALL TRACKS TRANSLATED
2520  4853  E1              10.0          POP   H               ;RESTORE REGISTER H + L
2521  4854  D1              10.0          POP   D               ;RESTORE REGISTER D + E
2522  4855  C1              10.0          POP   B               ;RESTORE REGISTER B + C
2523  4856  F1              10.0          POP   PSW             ;RESTORE REGISTER A + PSW
2524  4857  C9              10.0          RET                   ;RETURN TO USER
2525
```

```
2527                                    .SBTTL  SUBROUTINE CALCULATE POINTER
2528                                    ;*THE POINTERS PROVIDED FOR THE TRACK IN ERROR ARE ALSO DERIVED FROM THE
2529                                    ;*INPUT/EXPECTED DATA AS PER THE FOLLOWING TABLE.
2530                                    ;*
2531                                    ;*      INPUT/EXPECTED
2532                                    ;*          DATA                        POINTER GENERATED
2533                                    ;*      -------------                   -----------------
2534                                    ;*
2535                                    ;*         XX00XXXX                     HISTORY
2536                                    ;*         XX01XXXX                     PHASE TIE
2537                                    ;*         XX10XXXX                     AMTIE
2538                                    ;*         XX11XXXX                     ILLEGAL 5X4
2539                                    ;*
2540                                    ;*      X = DON'T CARE BITS
2541  4858                              SP
 (1)                                    ;*****************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
2542
2543  4858  F5              12.0  POINTER: PUSH   PSW             ;SAVE THE ACCUMULATOR
2544  4859  C5              12.0         PUSH   B               ;SAVE B + C
2545  485A  3A  36  49        13.0         LDA    INDAT           ;GET INPUT DATA
2546  485D  E6  30             7.0         ANI    $30             ;REMOVE THE WORKING BITS
2547  485F  07                 4.0         RLC                    ;POSITION THE DATA
2548  4860  07                 4.0         RLC
2549  4861  07                 4.0         RLC
2550  4862  07                 4.0         RLC
2551  4863  21  70  48          10.0         LXI    H,POINTBL       ;GET BASE ADDRESS OF TABLE
2552  4866  4F                 4.0         MOV    C,A
2553  4867  06  00             7.0         MVI    B,0
2554  4869  09                10.0         DAD    B
2555  486A  7E                 7.0         MOV    A,M             ;GET THE ECC POINTER
2556  486B  D3  0A            10.0         OUT    RTIEB           ;SET THE TIE BUS VALUE
2557  486D  C1                10.0         POP    B               ;RESTORE B + C
2558  486E  F1                10.0         POP    PSW             ;RESTORE ACCUMULATOR
2559  486F  C9                10.0         RET
2560
2561  4870  01                       POINTBL: .BYTE  1
2562  4871  02                                .BYTE  2
2563  4872  04                                .BYTE  4
2564  4873  08                                .BYTE  8
2565
```

```
                                .SBTTL  TABLE 4 X 5 TRANSLATION

                                ;THIS TABLE TRANSLATED DATA FOR AS FOLLOWS
                                ;  INPUT GROUP POSITIONS              OUTPUT GROUP POSITIONS
                                ;   1 2 3 4 / 5 6 7 8                 10 9 8 7 6 / 5 4 3 2 1
                                ;
                                ;       0000                              10011
                                ;       0001                              11011
                                ;       0010                              01001
                                ;       0011                              11001
                                ;       0100                              10111
                                ;       0101                              10101
                                ;       0110                              01101
                                ;       0111                              11101
                                ;       1000                              01011
                                ;       1001                              10010
                                ;       1010                              01010
                                ;       1011                              11010
                                ;       1100                              01111
                                ;       1101                              10110
                                ;       1110                              01110
                                ;       1111                              11110

4874  13        TAB4X5: .BYTE   @23
4875  1B                .BYTE   @33
4876  09                .BYTE   @11
4877  19                .BYTE   @31
4878  17                .BYTE   @27
4879  15                .BYTE   @25
487A  0D                .BYTE   @15
487B  1D                .BYTE   @35
487C  0B                .BYTE   @13
487D  12                .BYTE   @22
487E  0A                .BYTE   @12
487F  1A                .BYTE   @32
4880  0F                .BYTE   @17
4881  16                .BYTE   @26
4882  0E                .BYTE   @16
4883  1E                .BYTE   @36
```

```
2608                                    .SBTTL  SUBROUTINE VARIABLES
2609
2610  4884 0004                         TRNIN:  .BLKB   4               ;TABLE CONTAINING THE 4 DATA BYTES TO
2611                                                                    ;BE TRANSLATED
2612  4888 000A                         TRNOUT: .BLKB   10              ;TRANSLATED DATA TABLE DATA AFTER 4X5
2613                                                                    ;TRANSLATION
2614  4892   00                         GP4X5:  .BYTE   0               ;ACCUMULATOR FOR THE 4 BIT SUBGROUP TO
2615                                                                    ;BE TRANSLATED
2616  4893   00                         TRNTMP: .BYTE   0               ;TEMP. STORAGE FOR THE TRANSLATED SUBGROUP.
2617  4894   00                         BAD1:   .BYTE   0               ;BAD TRANSLATION POINTER 1
2618  4895   00                         BAD2:   .BYTE   0               ;BAD TRANSLATION POINTER 2
2619  4896   00                         BAD3:   .BYTE   0               ;BAD TRANSLATION POINTER 3
2620
```

```
2622                                    .SBTTL  SUBROUTINE CLEAR ECC
2623  4897                              S
 (1)                                    ; ************************************************************************
2624                                    ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION "ECCCHR" -
2625                                    ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
2626                                    ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
2627  4897                              S
 (1)                                    ; ************************************************************************
2628
2629  4897  AF                   4.0  CLECC:  XRA     A              ;CLEAR THE ACCUMULATOR
2630  4898  32  C5  48           13.0         STA     ECCCHR         ;CLEAR THE ECC CHARACTER
2631  489B  C9                   10.0         RET                    ;RETURN TO USER
2632
```

```
                                        .SBTTL  SUBROUTINE CALCUALTE ECC CHARACTER
489C                                    $
 (1)                                    ; ************************************************************************
                                        ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
                                        ;THE CONTENTS OF "ECCCHR" TO UPDATE THE CONTENTS OF "ECCCHR" ACCORDING
                                        ;TO THE ANSI STANDARD ECC POLYNOMIAL.
489C                                    $
 (1)                                    ; ************************************************************************

489C  E5              12.0  ECC:    PUSH    H               ;SAVE H&L
489D  D5              12.0          PUSH    D               ;SAVE D&E
489E  F5              12.0          PUSH    PSW             ;SAVE THE ACCUMULATOR
489F  21  C5  48      10.0          LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
48A2  AE               7.0          XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
48A3  5F               4.0          MOV     E,A             ;SAVE XOR RESULT IN E
48A4  E6  10           7.0          ANI     $10             ;IS BIT #4 OF RESULT SET
48A6  7B               4.0          MOV     A,E             ;RESTORE XOR RESULT FROM B
48A7  CA  AC  48      10.0          JZ      ECC1            ;CONTINUE IF BIT #4 RESET
48AA  EE  23           7.0          XRI     $23             ;ELSE-XOR WITH 23
48AC  5F               4.0  ECC1:    MOV     E,A             ;STORE THE ECC RESULT IN E
                                                            ;
48AD  AF               4.0          XRA     A               ;CLEAR A
48AE  4F               4.0          MOV     C,A             ;CLEAR THE TRANSLATE RESULT
48AF  21  BD  48      10.0          LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
48B2  CD  C6  48      18.0          CALL    TRANS           ;TRANSLATE THE BITS
48B5  79               4.0          MOV     A,C             ;GET THE TRANSLATED RESULT
48B6  32  C5  48      13.0          STA     ECCCHR          ;STORE RESULT
48B9  F1              10.0          POP     PSW             ;RESTORE THE ACCUMULATOR
48BA  D1              10.0          POP     D               ;RESTORE D&E
48BB  E1              10.0          POP     H               ;RESTORE H&L
48BC  C9              10.0          RET

48BD  08                    ECCTBL: $08                     ;BIT 0 = POSITION 3
48BE  20                            $20                     ;BIT 1 = POSITION 5
48BF  02                            $02                     ;BIT 2 = POSITION 1
48C0  40                            $40                     ;BIT 3 = POSITION 6
48C1  80                            $80                     ;BIT 4 = POSITION 7
48C2  01                            $01                     ;BIT 5 = POSITION 0
48C3  10                            $10                     ;BIT 6 = POSITION 4
48C4  04                            $04                     ;BIT 7 = POSITION 2

48C5  00                    ECCCHR: .BYTE   0
```

```
2676                                    .SBTTL  SUBROUTINE POLYNOMIAL BIT TRANSLATION
2677
2678                                    ;
2679                                    ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
2680                                    ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
2681                                    ;
2682
2683  48C6  06  01            7.0  TRANS:  MVI   B,1          ;INIT 'B' TO BIT POSITION 0
2684  48C8  7B                 4.0  TRANS1: MOV   A,E          ;GET CHAR TO BE TRANSLATED
2685  48C9  A0                 4.0          ANA   B          ;SEE IF BIT POSITION IN 'B' IS SET
2686  48CA  CA  D0  48        10.0          JZ    TRANS2     ;DO NEXT BIT POSITION IF NOT SET
2687  48CD  79                 4.0          MOV   A,C        ;GET PREVIOUS RESULT OF 'OR'
2688  48CE  B6                 7.0          ORA   M          ;'OR' IN NEW POSITION
2689  48CF  4F                 4.0          MOV   C,A        ;SAVE RESULT
2690
2691  48D0  78                 4.0  TRANS2: MOV   A,B
2692  48D1  07                 4.0          RLC              ;POSITION MASK TO NEXT BIT
2693  48D2  47                 4.0          MOV   B,A
2694  48D3  D8                12.0          RC               ;EXIT WHEN ALL POSITIONS DONE
2695  48D4  23                 6.0          INX   H          ;POINT TO NEXT TABLE ENTRY
2696  48D5  C3  C8  48        10.0          JMP   TRANS1     ;PROCESS NEXT BIT
2697
```

```
                                .SBTTL  SUBROUTINE TRK
48D8                            SSUB
                                ; ***************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*GENERATE BAD TRACK TRANSLATION
48D8                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE GENRATES THREE TRACK NUMBERS FOR THE BADT4X5 SUBROUTINE
                                ;*BASED ON THE VALUE OF INDAT AND THREE TIE BUS TIMES
                                ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
                                ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
                                ;*MESSAGE, AND BITS SIMULATED IN ERROR CAN BE DETERMINED FROM THE
                                ;*FOLLOWING TABLE.
                                ;*
                                ;*      INPUT/EXPECTED          DATA BIT POSITIONS
                                ;*          DATA                WITH ECC CORRECTION
                                ;*      -------------           -------------------
                                ;*
                                ;*         XXXX0000                    0,1,2
                                ;*         XXXX0001                    1,2,3
                                ;*         XXXX0010                    2,3,4
                                ;*         XXXX0011                    3,4,5
                                ;*         XXXX0100                    4,5,6
                                ;*         XXXX0101                    5,6,7
                                ;*         XXXX0110                    6,7,P
                                ;*         XXXX0111                    7,P,0
                                ;*         XXXX1000                    P,0,1
                                ;*         XXXX1001                    1,2,3
                                ;*         XXXX1010                    2,3,4
                                ;*         XXXX1011                    3,4,5
                                ;*         XXXX1100                    4,5,6
                                ;*         XXXX1101                    5,6,7
                                ;*         XXXX1110                    6,7,P
                                ;*         XXXX1111                    7,P,0
                                ;*
                                ;*      X = DON'T CARE BITS
48D8                            S
                                ; ***************************************************************************

48D8  3A 36 49      13.0  TRK:  LDA   INDAT         ;GET INPUT DATA
48DB  E6 0F          7.0        ANI   $0F           ;REMOVE 4 BITS
48DD  FE 08          7.0        CPI   $8            ;PARITY TRACK?
48DF  CA E4 48      10.0        JZ    TRKA          ;YES
48E2  E6 07          7.0        ANI   $7            ;NO-IGNORE BIT3
48E4  3C             4.0  TRKA: INR   A             ;INCREMENT FOR SUBROUTINE
48E5  32 94 48      13.0        STA   BAD1          ;SAVE IT.
48E8  3D             4.0        DCR   A             ;PREPARE TO USE AS TABLE INDEX
48E9  21 29 49      10.0        LXI   H,TIMTBL      ;INCORRECT TIE BUS TIME TABLE START
48EC  5F             4.0        MOV   E,A           ;LOW HALF OF TABLE INDEX
48ED  16 00          7.0        MVI   D,0           ;ZERO HIGH HALF
48EF  19            10.0        DAD   D             ;PRODUCE TABLE ENTRY ADDRESS
```

```
2746  48F0  7E          7.0          MOV   A,M          ;GET INCORRECT TIE BUS TIME FROM TABLE
2747  48F1  32 32 49    13.0         STA   TIET1        ;SAVE FIRST TIE BUS JAM TIME
2748  48F4  3A 94 48    13.0         LDA   BAD1
2749  48F7  3C          4.0          INR   A            ;GENERATE VALUE FOR BAD2
2750  48F8  FE 0A       7.0          CPI   $A           ;TOO BIG?
2751  48FA  C2 FF 48    10.0         JNZ   TRKB         ;JUMP IF OK
2752  48FD  3E 01       7.0          MVI   A,@1         ;YES
2753  48FF  32 95 48    13.0  TRKB:  STA   BAD2         ;SAVE IT
2754  4902  3D          4.0          DCR   A            ;PREPARETO USE AS TABLE INDEX
2755  4903  21 29 49    10.0         LXI   H,TIMTBL     ;TIE BUS TIME TABLE START
2756  4906  5F          4.0          MOV   E,A          ;LOW HALF OF TABLE INDEX
2757  4907  16 00       7.0          MVI   D,0          ;CLEAR UPPER HALF
2758  4909  19          10.0         DAD   D            ;PRODUCE TABLE ENTRY ADDRESS
2759  490A  7E          7.0          MOV   A,M          ;GET SECOND TIE BUS JAM TIME
2760  490B  32 33 49    13.0         STA   TIET2        ;SAVE IT
2761  490E  3A 95 48    13.0         LDA   BAD2
2762  4911  3C          4.0          INR   A
2763  4912  FE 0A       7.0          CPI   $A
2764  4914  C2 19 49    10.0         JNZ   TRKC
2765  4917  3E 01       7.0          MVI   A,@1
2766  4919  32 96 48    13.0  TRKC:  STA   BAD3
2767  491C  3D          4.0          DCR   A
2768  491D  21 29 49    10.0         LXI   H,TIMTBL
2769  4920  5F          4.0          MOV   E,A
2770  4921  16 00       7.0          MVI   D,0
2771  4923  19          10.0         DAD   D
2772  4924  7E          7.0          MOV   A,M
2773  4925  32 34 49    13.0         STA   TIET3
2774  4928  C9          10.0         RET
2775  4929  02          TIMTBL: .BYTE  2          ; TIE BUS TIME FOR BIT 0
2776  492A  03                  .BYTE  3          ; TIE BUS TIME FOR BIT 1
2777  492B  00                  .BYTE  0          ; TIE BUS TIME FOR BIT 2
2778  492C  01                  .BYTE  1          ; TIE BUS TIME FOR BIT 3
2779  492D  04                  .BYTE  4          ; TIE BUS TIME FOR BIT 4
2780  492E  08                  .BYTE  8          ; TIE BUS TIME FOR BIT 5
2781  492F  07                  .BYTE  7          ; TIE BUS TIME FOR BIT 6
2782  4930  05                  .BYTE  5          ; TIE BUS TIME FOR BIT 7
2783  4931  C6                  .BYTE  6          ; TIE BUS TIME FOR BIT P
2784
2785
```

```
                                                    .SBTTL  PROGRAM VARIABLES

                                                    TIET1:  .BYTE   0               ;FIRST TIE JAM TIME
                                                    TIET2:  .BYTE   0               ;SECOND TIE BUS JAM TIME
                                                    TIET3:  .BYTE   0               ;THIRD TIE BUS JAM TIME
                                                    SUNIT:  .BYTE   0               ;TU PORT #
                                                    INDAT:  .BYTE   0               ;DATA BYTE TO ECC
                                                    INDATP: .BYTE   0               ;DATA BYTE PARITY
                                                    UNITMP: .BYTE   0               ;UNIT MAP
                                                    AMTMSK: .WORD   0               ;AMTIE MASK WORD
                                                    DATAA:  .BYTE   0               ;ACTUAL DATA AFTER TRANSLATION
                                                    DATAAP: .BYTE   0               ;ACTUAL DATA PARITY AFTER TRANSLATION
                                                    DATAE:  .BYTE   0               ;EXPECTED DATA AFTER TRANSLATION
                                                    DATAEP: .BYTE   0               ;EXPECTED DATA PARITY AFTER TRANSLATION
                                                    BUFFER: .BLKB   7               ;DATA BYTE STORAGE
                                                            .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
AMTMSK  4939           ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000         BAD     47BC          BADST  = 0090
BADT4X  47AD           BAD1    4894          BAD1C   4809          BAD2    4895
BAD2C   4814           BAD3    4896          BAD3C   4826          BIT0   = 0001
BIT1   = 0002          BIT15  = 8000         BIT2   = 0004         BIT3   = 0008
BIT4   = 0010          BIT5   = 0020         BIT6   = 0040         BIT7   = 0080
BIT8   = 0100          BIT9   = 0200         BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60           BRKXCT= 4F00          BSAVE   4F9C          BUFFER  493F
BYTCNT= 00D4           BYTEH   4F24          BYTEL   4F23          B4X5    47CD
C      =%0001          CASCT   4F21          CASCTL= 00A0          CASSTA= 00A0
CATTH  = 0089          CATTL  = 0088         CBUSST= 00A1          CBYTH  = 008B
CBYTL  = 008A          CDG1H  = 0087         CDG1L  = 0086         CDG2H  = 0093
CDG2L  = 0092          CDG3H  = 0095         CDG3L  = 0094         CDVTH  = 008D
CDVTL  = 008C          CHPTIE= 0028          CHOTIE= 0020          CH1TIE= 0021
CH2TIE= 0022           CH3TIE= 0023          CH4TIE= 0024          CH5TIE= 0025
CH6TIE= 0026           CH7TIE= 0027          CKLOP  = 2815         CLEAR   4782
CLECC   4897           CLKCTL= 00F0          CLOCK   4F26          CLRLP   4786
CMCOH  = 0099          CMCOL  = 0098         CMC1H  = 009B         CMC1L  = 009A
CMC2H  = 009D          CMC2L  = 009C         CMC3H  = 009F         CMC3L  = 009E
CMINH  = 0097          CMINL  = 0096         CNTCTL= 00D7          CRCWRD= 0018
CSAVE   4F9D           CSRLH  = 0091         CSRLL  = 0090         CTCH   = 0085
CTCL   = 0084          CTSTH  = 008F         CTSTL  = 008E         CXCTH  = 0081
CXCTL  = 0080          CXINH  = 0083         CXINL  = 0082         C.     = 0001
C.AVAI= 0080           C.DP   = 0008         C.DSE  = 0010         C.DTU  = 0003
C.DVA  = 0008          C.FAIL= 00FC          C.FMT  = 0070         C.FNCT= 003E
C.GO   = 0001          C.INTC= 00FE          C.MAIN= 0020          C.NSA  = 0080
C.RCT  = 00FC          C.SER  = 0080         C.SHR  = 0040         C.SKPC= 000F
C.TAPE= 0040           C.WCS  = 0002         C4X5    47D6          D      =%0002
DATAA   493B           DATAAP  493C          DATACT= 00D0          DATAE   493D
DATAEP  493E           DBUS    4F28          DBUSCT= 00C0          DBUSST= 00C0
DDRA   = 00D8          DDRAIN= 0010          DDRB   = 00D9         DDRBIN= 0002
DDRC   = 00DA          DDRCIN= 0001          DDRCO  = 0088         DDRCTL= 00DB
DIAFLG  4F22           DIAGPG= 4300          DIAGRM= 4F90          DIARD  = 000B
DONE1  = 0045          DONINT= 0010          DSAVE   4F9E          DSE    = 0006
DUMMU   4643  G        DUMMV   46A9  G       DUMMW   4427  G       DUMMX   447A  G
DUMMY   431E  G        D.ATH0= 0001          D.ATH1= 0002          D.EOTD= 0010
D.LAGC= 0020           D.NOTW= 0040          D.NTHR= 0004          D.TACH= 0008
D.WR4  = 0080          D4X5    47C5          E      =%0003         ECC     489C
ECCBAD= 0042           ECCCHR  48C5          ECCCOR= 0019          ECCOK  = 0041
ECCSTA= 001A           ECCTBL  48BD          ECCTST= 000E          ECC1    48AC
EDATA  = 0095          EOTCLR= 0003          ERFLG   4F93          ERLP   = 2809
ERLPA  = 280F          ERLPB  = 2812         ERLPE  = 280C         ERNUM   4F90
ERRCNT= 00D6           ESAVE   4F9F          EXIT    4774          E.ACRC= 0010
E.AMT  = 0020          E.CDP  = 0080         E.CRC  = 0080         E.PNTR= 0008
E.RPE  = 0040          E.STEC= 0001          E.TTEC= 0002          E.UNC  = 0004
E4X5    47EF           FIFORD= 006A          FORMAT  4F25          FOUND   4344
FWDTST= 0061           F4X5    483B          GCRID  = 0089         GCRSET= 0002
GOODTM= 0092           GP4X5   4892          H      =%0004         HLSAVE  4FA0
IE     = 0008          INDAT   4936          INDATP  4937          INTSTA= 00E0
INV4X5  481F           ITERA   4F9A          I.PWR  = 0020         I.RMPE= 0040
I5.5   = 0010          I6.5   = 0020         I7.5   = 0040         KCALL  = 005F
KCLR   = 007B          KDEP   = 003F         KENAB  = 0078         KEXAM  = 003E
KLOBRD= 00C8           KEY1   = 0078         KEY10  = 006D         KEY11  = 006E
```

```
KEY12 = 006F    KEY13 = 005C    KEY14 = 005D    KEY15 = 005E
KEY16 = 005F    KEY17 = 003C    KEY18 = 003D    KEY19 = 003E
KEY2  = 0079    KEY20 = 003F    KEY3  = 007A    KEY4  = 007B
KEY5  = 0074    KEY6  = 0075    KEY7  = 0076    KEY8  = 0077
KEY9  = 006C    KINTA = 006F    KLDAD = 003D    KN0   = 003C
KN1   = 005C    KN2   = 005D    KN3   = 005E    KN4   = 006C
KN5   = 006D    KN6   = 006E    KN7   = 0074    KN8   = 0075
KN9   = 0076    KU2   = 0079    KU3   = 007A    KU8   = 0077
L     =%0005    LBLANK= 000F    LCE   = 000B    LCH   = 000C
LCL   = 000D    LCP   = 000E    LC0   = 0000    LC1   = 0001
LC2   = 0002    LC3   = 0003    LC4   = 0004    LC5   = 0005
LC6   = 0006    LC7   = 0007    LC8   = 0008    LC9   = 0009
LDLEDA= 00CA    LDLEDB= 00CB    LDLEDC= 00CC    LDLEDD= 00CD
LDLEDE= 00CE    LDLEDF= 00CF    LKDIAG= 2800    LKKBD = 004C
LKKEY = 0049    LKLWMG= 0058    LKLWMP= 0055    LKLWPG= 0052
LKLWPP= 004F    LKMOD7= 0046    LKOPR = 0046    LPFLG   4F94
LPNUM   4F92    M     =%0006    MBSEL = 00E0    MB.A  = 0008
MB.B  = 0004    MEMTOP= 4FFF    MINUS = 000A    MM    = 8000
MSE   = 0008    MSGN  = 000F    MTACLR= 0000    MT.ARA= 0020
MT.CPE= 0080    MT.DSE= 0001    MT.FWD= 0040    MT.INH= 0008
MT.LWR= 0004    MT.MOT= 0002    MT.NWT= 0080    MT.PEC= 0040
MT.PSB= 0004    MT.PS0= 0001    MT.PS1= 0002    MT.REV= 0020
MT.WRT= 0010    MT.Z  = 0008    M.ATA = 0080    M.CAPE= 0020
M.CONT= 0080    M.DEM = 0020    M.EBL = 0010    M.EXC = 0008
M.FAIL= 0008    M.ILR = 0010    M.INIT= 0010    M.OCC = 0020
M.ONLI= 0001    M.PE  = 0040    M.PGMI= 0080    M.RDEN= 0002
M.RDPE= 0008    M.RUN = 0004    M.SCLK= 0001    M.TRA = 0040
M.UNIT= 0007    M.WCLK= 0040    M.WCLN= 0080    M.WREN= 0080
M5.5  = 0001    M6.5  = 0002    M7.5  = 0004    NOTCAP= 0088
OKAY  = 00FF    OPRRAM= 4300    OPSTRT= 0058    OPVER = 0040
PADCNT= 00D5    PADCRC= 0080    PDIAG = 0048    PEID  = 008A
PENAB = 004C    PESET = 0001    PL    = 00B1    POINTB  4870
POINIE  4858    PO4X5   47DC    PRDD  = 004C    PRENF = 009C
PS    = 00B2    PSTAT = 0048    PSW   =%0009    P.AMTP= 0001
P.BCTC= 0040    P.CMDP= 0020    P.INTE= 0080    P.LCS = 0040
P.LWR = 0020    P.RDP = 0002    P.RPEN= 0004    P.RPST= 0002
P.RPOE= 0020    P.RP1E= 0010    P.RP2E= 0020    P.RP3E= 0010
P.SING= 0080    P.STAT= 0002    P.STPE= 0080    P.TACH= 0008
P.TUPR= 0010    P.WCSP= 0004    P.WDS = 0040    P.WFLP= 0001
P.WPEN= 0010    P.WPOE= 0008    P.WP1E= 0004    P.WP2E= 0008
P.WP3E= 0004    P.5VOK= 0002    QUE   = 281B    QUEM  = 281E
RAMT  = 0010    RARA  = 0006    RARAI = 0004    RCHBD0= 0048
RCHBD1= 0047    RCHOK = 0046    RCHTST= 000C    RCLRT = 000D
RCMD  = 000B    RCMLP = 0003    RCONT = 0080    RDATA = 0017
RDCLK = 0010    RDON  = 0011    READG = 0007    REND  = 0014
REQST = 2806    RESCHR= 00D1    RETRY   4777    REVTST= 0064
REWIND= 0004    RFIFOL= 0008    RGCLK = 0002    RGCRI = 0003
RIBG  = 0001    RILL  = 0012    RINST = 000C    RMCTST= 0008
RMK2  = 0013    RNOP  = 0000    RPATH = 0001    RPBAD = 0044
RPCHI = 0001    RPCLK = 0003    RPCTL = 0009    RPEI  = 0002
RPFAIL= 0000    RPF1  = 009D    RPF2  = 009E    RPOK  = 0043
RPOSTN= 0016    RPSTA = 0015    RRCMT = 000A    RSTAT = 0002
RTIEB = 000A    RTIER = 0030    RTM   = 0005    RUNKI = 0009
RUPTST= 005E    RWDUNL= 0005    R.AMT = 0001    R.BOP = 0008
```

```
R.DATA= 0040        R.DON = 0002        R.DRDY= 0010        R.END = 0010
R.ILL = 0004        R.JVOK= 0004        R.MK2 = 0008        R.PLOD= 0008
R.PLO0= 0010        R.PLO1= 0020        R.POST= 0020        R.STNM= 0002
R.STOP= 0004        R.STPC= 0001        R.TBJN= 0080        R.TSTD= 0040
R.VOK = 0080        R00H  = 0081        R00L  = 0080        R01H  = 0083
R01L  = 0082        R02H  = 0085        R02L  = 0084        R03H  = 0087
R03L  = 0086        R04H  = 0089        R04L  = 0088        R05H  = 008B
R05L  = 008A        R06H  = 008D        R06L  = 008C        R07H  = 008F
R07L  = 008E        R10H  = 0091        R10L  = 0090        R11H  = 0093
R11L  = 0092        R12H  = 0095        R12L  = 0094        R13H  = 0097
R13L  = 0096        R14H  = 0099        R14L  = 0098        R15H  = 009B
R15L  = 009A        R16H  = 009D        R16L  = 009C        R17H  = 009F
R17L  = 009E        R7.5  = 0010        SELCLR= 0000        SETATA= 00A1
SID   = 0080        SOD   = 0080        SOE   = 0040        SP    =%0008
SSCLK = 0040        SSTEP = 0005        STACK = 4FFF        STATRM= 4F20
STPCT   4F20        STRSP = 5000        SUNIT   4935        TAB4X5  4874
TADR00= 0080        TADR01= 0081        TADR02= 0082        TADR03= 0083
TADR04= 0084        TADR05= 0085        TADR06= 0086        TADR07= 0087
TADR10= 0088        TADR11= 0089        TADR12= 008A        TADR13= 008B
TAMT  = 0044        TASEL = 0080        TCMD  = 0040        TC.FWD= 0040
TC.INH= 0008        TC.LWR= 0004        TC.REV= 0020        TC.WRT= 0010
TEMP    4F99        TEST1   4300        TEST2   4543        TIET1   4932
TIET2   4933        TIET3   4934        TIMTBL  4929        TMF   = 0099
TMRDY = 0040        TRANS   48C6        TRANS1  48C8        TRANS2  48D0
TRK     48D8        TRKA    48E4        TRKB    48FF        TRKC    4919
TRKENA= 00D2        TRNIN   4884        TRNOUT  4888        TRNTMP  4893
TSET  = 2803        TSTEND= 2818        TSTS  = 0040        TST01A  4430
TST01B  4434        TST01C  437F        TST01F  4397        TST01G  43C4
TST01H  43F6        TST01I  43A3        TST01K  4483        TST01L  4496
TST01M  44AA  G     TST01N  44EC        TST01Q  44D1        TST01R  44C8  G
TST01S  44E1        TST01T  44DB  G     TST01U  4503  G     TST01V  451F  G
TST01X  4351        TST01Y  4485        TST02A  464C        TST02B  4653
TST02C  457C        TST02F  4597        TST02G  45E1        TST02H  4612
TST02I  45AA        TST02K  46B2        TST02L  46C6        TST02M  46D9  G
TST02N  471F        TST02Q  4700        TST02R  46F7  G     TST02S  4714
TST02T  470E  G     TST02U  4736  G     TST02V  4752  G     TST02W  45B6
TST02X  454D        TST02Y  46B7        TST02Z  45BD        TUSEL0= 00D1
TUSEL1= 00D2        TU78  = 0010        T.ATH0= 0001        T.ATH1= 0002
T.BOT = 0004        T.EOT = 0002        T.FPT = 0001        T.NTHR= 0004
T.ONL = 0020        T.PES = 0008        T.PSBJ= 0020        T.PSOJ= 0008
T.PS1J= 0010        T.RDY = 0080        T.RDYO= 0040        T.RWD = 0010
T.SCLK= 0002        T4X5    47B1        UIBG  = 00A1        UNITMP  4938
UNITSL  4327        VALFC   4F98        VALTB   4F95        VELTST= 005B
WDR.P = 0010        WMCCTL= 00D3        WMCERR= 00DA        WMCSTA= 00D0
WRTCLK= 0000        WRTDAT= 00D3        W.ACRC= 0004        W.CRC = 0008
W.DIAG= 0002        W.DONN= 0040        W.ECC = 0010        W.ENAB= 0080
W.ERR = 0020        W.FMT = 0070        W.GCR = 0010        W.LEFT= 0004
W.ONES= 0020        W.RESI= 0002        W.REV = 0004        W.ROME= 0010
W.RST = 0001        W.SKIP= 000F        W.WRIT= 0008        W.XFER= 0020
X     =%000A        X.DONN= 0080        X.ENAB= 0040        X.PEPE= 0002
X.ROME= 0001        X.WCLK= 0001        Y     =%000B        .     = 4946
```

ERRORS DETECTED: 0

*ECC4.A78/PTP,ECC4=NLIST,PARAM,MACRO,LIST,ECC4
RUN-TIME: 5 7 0 SECONDS

     3   "MEMORY MAP" EXTENDED ADDRESS BIT
    19   READ MICRO CONTROLLER CONTROL & STATUS WORDS
   139   TRANSPORT/PORT COMMAND + STATUS WORDS
   380   COMMON ADDRESS SPACE REGISTERS
   444   MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
   485   MASSBUS CONTROL WORD DEFINITIONS
   557   WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
   709   KEYBOARD/DISPLAY CONTROL + STATUS WORD
   797   MISCELLANEOUS VARIABLE DEFINITIONS
   801   RAM ADDRESS DEFINITIONS
   819   LINKAGE TABLE ADDRESSES
   842   BIT DEFINITIONS
   856   RIM AND SIM INSTRUCTION BIT DEFINITIONS
   877   INTERRUPTABLE INSTRUCTION MACROS
   924   DIAGNOSTIC VARIABLES
   945   COMMON VARIABLE STORAGE
   959   MACRO CALL DEFINITIONS
  1047   TM78 MICRO-DIAGNOSTIC MACRO CALLS
  1101   8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
  1333   TEST 1 - ECC BAD DATA TWO TRACKS TWO HARD AND SEVEN SOFT POINTERS
  1855   TEST 2 - ECC BAD DATA TRACKS 1 AND 9 NINE SOFT POINTERS
  2358   SUBROUTINE CLEAR ALL TU PORTS
  2402   SUBROUTINE 4X5 TRANSLATE A SUBGROUP
  2546   SUBROUTINE CALCULATE POINTER
  2581   TABLE 4 X 5 TRANSLATION
  2622   SUBROUTINE VARIABLES
  2636   SUBROUTINE CLEAR ECC
  2648   SUBROUTINE CALCUALTE ECC CHARACTER
  2690   SUBROUTINE POLYNOMIAL BIT TRANSLATION
  2713   SUBROUTINE TRK
  2801   PROGRAM VARIAB_ES

```
                ;*************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;                "BYTE/SCLK COUNT NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                MM = DATA FORMAT FROM CAS REGISTER 2
                ;                NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;                BYTE-SCLK COUNT = LLL
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                LLL = AS ABOVE
                ;                MM  = AS ABOVE
                ;                NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;                TRANSITION COUNT = LLL
                ;                WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;                EXPECTED 18 BITS =      E EEEEEE
                ;                ACTUAL 18 BITS =        A AAAAAA
                ;
                ;                WHERF EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;                BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;                OF CAS REG 15 LOW BYTE.
                ;
                ;                ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;                AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;                17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;                TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;                AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;                "SUBGROUP NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;*************************************************************************
```

```
                        ;*************************************************************************
                        ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
                        ;TO CAUSE SOME HOST CPU ACTION.

                        ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
                        ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
                        ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
                        ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
                        ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
                        ;               - HOST RESPONSE CODE 31 OR 33
                        ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
                        ;               - HOST RESPONSE CODE 31 OR 33
                        ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
                        ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
                        ;                       BIT0 = 1 TEST PORT 0
                        ;                       BIT1 = 1 TEST PORT 1
                        ;                       BIT2 = 1 TEST PORT 2
                        ;                       BIT3 = 1 TEST PORT 3
                        ;*************************************************************************
                        ;*************************************************************************
                        ;DATA PATTERN CODES

                        ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
                        ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
                        ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
                        ;               18 BITS OF ALL 1'S
                        ;               18 BITS OF ALL 0'S
                        ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
                        ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
                        ;*************************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  ECC5 - ECC CONTROLLER PART #5
                                .SBTTL  TEST 1 - ECC BAD DATA TWO TRACKS TWO HARD AND SEVEN SOFT POINTERS
                                ;ID     ECC5-ERROR CORRECTION CONTROLLER PART #5
4300                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*ECC BAD DATA TWO TRACKS TWO HARD AND SEVEN SOFT POINTERS
4300                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CHECKS THE ABILITY OF THE ECC CONTROLLER TO PERFORM TWO
                                ;*TRACK ERROR CORRECTION WITH INCORRECT DATA IN TWO TRACKS, AND TWO
                                ;*CORRECT POINTERS ALSO PROVIDED. HOWEVER, AN ADDITIONAL 7 SOFT POINTERS
                                ;*ARE PROVIDED TO TEST THE ABILITY OF THE ECC TO DESCRIMINATE
                                ;*BETWEEN HARD AND SOFT POINTERS.  THE PROGRAM CHECKS THAT THE DATA IS
                                ;*CORRECTED, AND THAT THE TWO TRACK ERROR CORRECT STATUS BIT IS SET.
                                ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
                                ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
                                ;*MESSAGE, AND THE DATA BIT POSITIONS OF THE INCORRECT DATA
                                ;*CAN BE DETERMINED FROM THE FOLLOWING TABLE.
                                ;*
                                ;*                                                  DATA BIT POSITION
                                ;*      BYTE COUNT              DATA                WITH HARD POINTERS
                                ;*      ----------              ----                ------------------
                                ;*
                                ;*         001          000,001,002,003,004,005,006        7,P
                                ;*         002          007,010,011,012,013,014,015        6,7
                                ;*         003          016,017,020,021,022,023,024        5,6
                                ;*         004          025,026,027,030,031,032,033        4,5
                                ;*         005          034,035,036,037,040,041,042        3,4
                                ;*         006          043,044,045,046,047,050,051        2,3
                                ;*         007          052,053,054,055,056,057,060        1,2
                                ;*         010          061,062,063,064,065,066,067        0,0
                                ;*         011          070,071,072,073,074,075,076        7,P
                                ;*         012          077,100,101,102,103,104,105        6,7
                                ;*         013          106,107,110,111,112,113,114        5,6
                                ;*         014          115,116,117,120,121,122,123        4,5
                                ;*         015          124,125,126,127,130,131,132        3,4
                                ;*         016          133,134,135,136,137,140,141        2,3
                                ;*         017          142,143,144,145,146,147,150        1,2
                                ;*         020          151,152,153,154,155,156,157        0,1
                                ;*         021          160,161,162,163,164,165,166        7,P
                                ;*         022          167,170,171,172,173,174,175        6,7
                                ;*         023          176,177,200,201,202,203,204        5,6
                                ;*         024          205,206,207,210,211,212,213        4,5
                                ;*         025          214,215,216,217,220,221,222        3,4
                                ;*         026          223,224,225,226,227,230,231        2,3
                                ;*         027          232,233,234,235,236,237,240        1,2
                                ;*         030          241,242,243,244,245,246,247        P,0
                                ;*         031          250,251,252,253,254,255,256        7,P
                                ;*         032          257,260,261,262,263,264,265        6,7
                                ;*         033          266,267,270,271,272,273,274        5,6
```

```
1380                                    ;*          034          275,276,277,300,301,302,303          4,5
1381                                    ;*          035          304,305,306,307,310,311,312          3,4
1382                                    ;*          036          313,314,315,316,317,320,321          2,3
1383                                    ;*          037          322,323,324,325,326,327,330          1,2
1384                                    ;*          040          331,332,333,334,335,336,337          0,1
1385                                    ;*          041          340,341,342,343,344,345,346          7,P
1386                                    ;*          042          347,350,351,352,353,354,355          6,7
1387                                    ;*          043          356,357,360,361,362,363,364          5,6
1388                                    ;*          044          365,366,367,370,371,372,373          4,5
1389                                    ;*          045          374,375,376,377,000,001,002          3,4
1390                                    ;*
1391                                    ;*THE POINTERS PROVIDED FOR THE TWO TRACKS IN ERROR ARE BAD 4x5.  THE
1392                                    ;*POINTERS PROVIDED FOR THE OTHER POSITIONS ARE PHASE TIE.
1393  4300                              SP
 (1)                                    ; ************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
1394                                    ;*BGNTST
1395                                    ;*    SET NORMAL READ PATH CLOCK
1396                                    ;*    CALL SUBROUTINE CLEAR
1397                                    ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
1398                                    ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
1399                                    ;*    DELAY 50 MICROSECONDS
1400                                    ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
1401                                    ;*    CLEAR THE INPUT DATA
1402                                    ;*    CLEAR BAD TRANSLATION POINTERS
1403                                    ;*  BGNDO
1404                                    ;*  : ENABLE THE READ PATH CLOCK
1405                                    ;*  : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
1406                                    ;*  : CLOCK THE FIFO'S
1407                                    ;*  : ISSUE CLEAR ALL COMMAND
1408                                    ;*  : WAIT
1409                                    ;*  : STOP THE READ PATH, SET TIE BUS JAM AND SET PLO DISABLE
1410                                    ;*  : CALL SUBROUTINE CLECC
1411                                    ;*  : SET THE LOOP COUNT TO 7
1412                                    ;*  : BGNDO
1413                                    ;*  : :  GET THE INPUT DATA BYTE
1414                                    ;*  : :  STORE IN THE BUFFER
1415                                    ;*  : :  CALL SUBROUTINE ECC
1416                                    ;*  : :  DECREMENT THE LOOP COUNT
1417                                    ;*  : :  INCREMENT THE INPUT DATA
1418                                    ;*  : :  DO UNTIL THE LOOP COUNT=0
1419                                    ;*  : ENDDO
1420                                    ;*  : SET THE LOOP COUNT TO 3
1421                                    ;*  : BGNDO
1422                                    ;*  : :  FILL THE TPANSLATOR INPUT BUFFER WITH THE DATA FROM THE BUFFER
1423                                    ;*  : :  CALL SUBROUTINE TRK
1424                                    ;*  : :  CALL SUBROUTINE BADT4X5
1425                                    ;*  : :  INIT THE LOOP COUNT TO 5
1426                                    ;*  : :  BGNDO
1427                                    ;*  : :  :  MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
1428                                    ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1429                                    ;*  : :  :  MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
1430                                    ;*  : :  :  CLOCK THE DATA INTO THE FIFO
```

```
1431                    ;*  : : :  DECREMENT THE LOOP COUNT
1432                    ;*  : : :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1433                    ;*  : : :  DO UNTIL LOOP COUNT = 0
1434                    ;*  : : ENDDO
1435                    ;*  : : FILL THREE BYTES OF TRANSLATOR INPUT BUFFER WITH THE BUFFER DATA
1436                    ;*  : : FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
1437                    ;*  : : CALL SUBROUTINE TRK
1438                    ;*  : : CALL SUBROUTINE BADT4X5
1439                    ;*  : : INIT THE LOOP COUNT TO 5
1440                    ;*  : : BGNDO
1441                    ;*  : : :  MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
1442                    ;*  : : :  MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
1443                    ;*  : : :  CLOCK THE DATA INTO THE FIFO
1444                    ;*  : : :  DECREMENT THE LOOP COUNT
1445                    ;*  : : :  DO UNTIL THE LOOP COUNT = 0
1446                    ;*  : : ENDDO
1447                    ;*  : : DECREMENT THE LOOP COUNT
1448                    ;*  : : DO UNTIL THE LOOP COUNT = 0
1449                    ;*  : ENDDO
1450                    ;*  : ISSUE DIAGNOSTIC READ COMMAND
1451                    ;*  : SET UP WATCHDOG TIMER COUNT
1452                    ;*  : BGNDO
1453                    ;*  : :    SINGLE STEP THE READ PATH
1454                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1455                    ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1456                    ;*  : ENDDO
1457                    ;*  : IF WATCHDOG TIMER=0
1458                    ;*  : :    THEN-ERROR
1459                    ;*  : :    ELSE-CONTINUE
1460                    ;*  : ENDIF
1461                    ;*  : GET THE TIE BUS TABLE ADDRESS
1462                    ;*  : SINGLE STEP THE READ PATH
1463                    ;*  : INIT THE LOOP COUNT TO 0
1464                    ;*  : BGNDO
1465                    ;*  : :    IF LOOP COUNT=TIE BUS JAM COUNT 1 OR JAM COUNT 2
1466                    ;*  : :    :  THEN-LOAD A BAD 4X5 POINTER
1467                    ;*  : :    :  ELSE-LOAD A PHASE TIE POINTER
1468                    ;*  : :    ENDIF
1469                    ;*  : :    SINGLE STEP THE READ PATH
1470                    ;*  : :    DECREMENT THE LOOP COUNT
1471                    ;*  : :    DO UNTIL LOOP COUNT = 9
1472                    ;*  : ENDDO
1473                    ;*  : SET UP WATCHDOG TIMER COUNT
1474                    ;*  : BGNDO
1475                    ;*  : :  SINGLE STEP THE READ PATH
1476                    ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
1477                    ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
1478                    ;*  : ENDDO
1479                    ;*  : IF WATCHDOG TIMER=0
1480                    ;*  : :  THEN-ERROR-EXIT TEST
1481                    ;*  : :  ELSE-CONTINUE
1482                    ;*  : ENDIF
1483                    ;*  : SINGLE STEP THE READ PATH
1484                    ;*  : INIT THE  LOOP COUNT TO 7
```

```
                                ;*   : BGNDO
                                ;*   : :  SINGLE STEP THE READ PATH
                                ;*   : :  COMPARE CORRECTED DATA WITH INPUT DATA
                                ;*   : :  IF NOT EQUAL
                                ;*   : :  :  THEN-ECC ERROR
                                ;*   : :  :  ELSE-CONTINUE
                                ;*   : :  ENDIF
                                ;*   : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
                                ;*   : :  IF NOT EQUAL
                                ;*   : :  :  THEN-ECC ERROR
                                ;*   : :  :  ELSE-CONTINUE
                                ;*   : :  ENDIF
                                ;*   : :  DECREMENT THE LOOP COUNT
                                ;*   : :  DO UNTIL THE LOOP COUNT = 0
                                ;*   : ENDDO
                                ;*   : SINGLE STEP THE READ PATH
                                ;*   : IF EXPECTED NOT EQUAL
                                ;*   : :  THEN-ECC ERROR
                                ;*   : :  ELSE-CONTINUE
                                ;*   : ENDIF
                                ;*   : INPUT THE ECC STATUS
                                ;*   : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
                                ;*   : IF THE RESULTING STATUS=SINGLE TRACK ERROR CORRECT + POINTER ERROR
                                ;*   : :  THEN-CONTINUE
                                ;*   : :  ELSE-ERROR
                                ;*   : ENDIF
                                ;*   : INCREMENT THE INPUT DATA
                                ;*   : DO UNTIL INPUT DATA = 0
                                ;*  ENDDO
                                ;*ENDTST
1515  4300                      SE
 (1)                            ; *****************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1516                            ;*ECC5 MICRO TEST 01
1517                            ;*ECC5 MICRO ERROR 01
1518                            ;*ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1519                            ;*M8951, M8950'S
1520                            ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1521                            ;*FATAL ERROR - TEST ABORTED
1522                            ;*
1523                            ;*ECC5 MICRO TEST 01
1524                            ;*ECC5 MICRO ERROR 02
1525                            ;*ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1526                            ;*M8951, M8950'S
1527                            ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
1528                            ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1529                            ;*'BYTE/SCLK COUNT NUMBER = LLL''
1530                            ;*FATAL ERROR - MICRO TEST ABORTED
1531                            ;*
1532                            ;*ECC5 MICRO TEST 01
1533                            ;*ECC5 MICRO ERROR 03
1534                            ;*ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1535                            ;*M8951, M8950'S
```

```
1536                                    ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
1537                                    ;*SENT TO THE ECC
1538                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1539                                    ;*FATAL ERROR - MICRO TEST ABORTED
1540                                    ;*
1541                                    ;*ECC5 MICRO TEST 01
1542                                    ;*ECC5 MICRO ERROR 04
1543                                    ;*ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1544                                    ;*M8951, M8950'S
1545                                    ;*CORRECTED DATA FROM THE ECC-INCORRECT
1546                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1547                                    ;*ACTUAL = NNNN
1548                                    ;*EXPECTED = NNNN
1549                                    ;*
1550                                    ;*ECC5 MICRO TEST 01
1551                                    ;*ECC5 MICRO ERROR 05
1552                                    ;*ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1553                                    ;*M8951, M8950'S
1554                                    ;*CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
1555                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1556                                    ;*
1557                                    ;*ECC5 MICRO TEST 01
1558                                    ;*ECC5 MICRO ERROR 06
1559                                    ;*ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1560                                    ;*M8951, M8950'S
1561                                    ;*CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
1562                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1563                                    ;*
1564                                    ;*ECC5 MICRO TEST 01
1565                                    ;*ECC5 MICRO ERROR 07
1566                                    ;*ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1567                                    ;*M8951, M8950'S
1568                                    ;*CORRECTED ECC CHARACTER INCORRECT
1569                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1570                                    ;*
1571                                    ;*ECC5 MICRO TEST 01
1572                                    ;*ECC5 MICRO ERROR 10
1573                                    ;*ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1574                                    ;*M8951, M8950'S
1575                                    ;*ECC STATUS INCORRECT AFTER A DATA GROUP
1576                                    ;*ACTUAL = NNNN
1577                                    ;*EXPECTED = NNNN
1578                                    ;*BYTE/SCLK COUNT NUMBER = LLL
1579  4300                              S
 (1)                                    ; ***************************************************************************
1580
1581  4300                              TEST1:  TESTX   @1              ;INITIALIZE THE TEST
 (1)  4300  3E  01              7.0                     MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28         18.0                     CALL    TSET            ;SETUP THE TEST
1582                                    ;%ECC5-ECC CONTROLLER-BAD DATA TWO TRACKS CORRECT POINTERS
1583                                    ;&M8951, M8950'S
1584
1585  4305                                      REQ     @7,0,0,0,0
 (1)  4305  CD  06  28         18.0                     CALL    REQST
```

```
  (1)  4308  00                                        .BYTE   0               ;DATA PATTERN NUMBER
  (1)  4309  00  00                                    .WORD   0               ;SYSTEM "0" COUNT
  (1)  430B  00  00                                    .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (1)  430D  00                                        .BYTE   0               ;DATA COMPARE FLAG IF =1
  (1)  430E  07                                        .BYTE   @7              ;REQUEST CODE
1586   430F                              RIN     R12L
  (1)  430F  DB  94          10.0                      IN      R12L            ;READ R12L INTO AC
  (1)  4311  7F               4.0                      MOV     A,A             ;RETRY LINK
1587   4312  32  56  49       13.0              STA     UNITMP
1588   4315  A7               4.0              ANA     A
1589   4316  C2  27  43       10.0              JNZ     UNITSL
1590   4319                                    ERR     EXIT,DUMMY
  (1)                                           ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  4319  CD  09  28       18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
  (1)       0001                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  431C  01                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  431D  00                                        .BYTE
  (1)  431E  CD  15  28       18.0              DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
  (1)  4321  DA  95  47       10.0                      JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1591                                    ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1592                                    ;>FATAL ERROR - TEST ABORTED
1593   4324  C3  95  47       10.0              JMP     EXIT
1594
1595   4327  06  00            7.0      UNITSL: MVI     B,0
1596   4329  3A  56  49       13.0              LDA     UNITMP          ;DID THE USER SPECIFY TU
1597   432C  E6  01            7.0              ANI     @01             ;PORT 0?
1598   432E  C2  44  43       10.0              JNZ     FOUND           ;YES-GO USE PORT #0
1599   4331  04               4.0              INR     B               ;NO-UPDATE POINTER TO PORT #1
1600   4332  3A  56  49       13.0              LDA     UNITMP          ;DID THE USER SPECIFY TU
1601   4335  E6  02            7.0              ANI     @02             ;PORT 1?
1602   4337  C2  44  43       10.0              JNZ     FOUND           ;YES-GO USE PORT #1
1603   433A  04               4.0              INR     B               ;NO-UPDATE POINTER TO PORT #2
1604   433B  3A  56  49       13.0              LDA     UNITMP          ;DID THE USER SPECIFY TU
1605   433E  E6  04            7.0              ANI     @04             ;PORT 2?
1606   4340  C2  44  43       10.0              JNZ     FOUND           ;YES-GO USE PORT #2
1607   4343  04               4.0              INR     B               ;NO-ASSUME PORT #3
1608   4344  CD  A3  47       18.0      FOUND:  CALL    CLEAR           ;CLEAR ALL TU PORTS
1609   4347  DB  E0           10.0              IN      INTSTA
1610   4349  E6  80            7.0              ANI     BIT7
1611   434B  B0               4.0              ORA     B
1612   434C  D3  E0           10.0              OUT     MBSEL
1613   434E  32  53  49       13.0              STA     SUNIT           ;SAVE FOR NEXT TEST
1614
1615   4351  3E  10            7.0      TST01X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
1616   4353  D3  F0           10.0              OUT     CLKCTL
1617   4355  3E  10            7.0              MVI     A,W.GCR
1618   4357  D3  D3           10.0              OUT     WMCCTL
1619   4359  3E  04            7.0              MVI     A,P.RPEN
1620   435B  D3  4C           10.0              OUT     PENAB
1621
1622   435D  3E  60            7.0              MVI     A,P.LWR!P.LCS   ;SET THE PORT CONTROL TO
1623   435F  D3  48           10.0              OUT     PDIAG           ;LCS, LWR
1624   4361  AF               4.0              XRA     A
1625   4362  D3  44           10.0              OUT     TAMT
```

```
1626  4364  3E  0A        7.0          MVI   A,10              ;DELAY
1627  4366  3D            4.0  1$:     DCR   A                 ;50
1628  4367  C2  66  43    10.0         JNZ   1$                ;MICROSECONDS
1629
1630  436A  D3  D2        10.0         OUT   TRKENA            ;CLEAR ALL TRACKS FROM
1631  436C  D3  0A        10.0         OUT   RTIEB             ;CLEAR THE TIE BUS REGISTER
1632  436E  D3  D2        10.0         OUT   TRKENA
1633  4370  32  B1  48    13.0         STA   BAD1              ;CLEAR THE INVALID
1634  4373  32  B2  48    13.0         STA   BAD2              ;TRACK TRANSLATIONS
1635  4376  32  B3  48    13.0         STA   BAD3              ;POINTERS
1636  4379  32  54  49    13.0         STA   INDAT             ;CLEAR THE INPUT DATA BYTE
1637  437C                             ROUT  R05H
 (1)  437C  D3  8B        10.0                OUT   R05H              ;WRITE AC INTO R05H
 (1)  437E  7F            4.0                 MOV   A,A               ;RETRY LINK
1638
1639  437F  3E  A8        7.0  TST01C: MVI   A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1640                                                                  ;SET TIE BUS JAM AND PLO DISABLE
1641  4381  D3  09        10.0         OUT   RPCTL
1642  4383  D3  08        10.0         OUT   RFIFOL            ;CLOCK THE FIFO'S
1643  4385  3E  0D        7.0          MVI   A,RCLRT           ;ISSUE CLEAR ALL COMMAND
1644  4387  D3  0B        10.0         OUT   RCMD
1645  4389  00            4.0          NOP                     ;WAIT
1646  438A  00            4.0          NOP
1647  438B  00            4.0          NOP
1648  438C  00            4.0          NOP
1649  438D  00            4.0          NOP
1650  438E  3E  A9        7.0          MVI   A,R.PLO1!R.STPC.R.PLOD.R.TBJN   ;STOP THE READ PATH
1651                                                                  ;SET TIE BUS JAM AND PLO DISABLE
1652  4390  D3  09        10.0         OUT   RPCTL
1653  4392  CD  B4  48    18.0         CALL  CLECC             ;CLEAR THE ECC CHARACTER
1654  4395  21  5D  49    10.0         LXI   H,BUFFER          ;SET UP THE BUFFER POINTER
1655  4398  16  07        7.0          MVI   D,@7              ;SET UP THE LOOP COUNT
1656  439A  3A  54  49    13.0  TST01F: LDA   INDAT            ;GET THE INPUT DATA
1657  439D  77            7.0          MOV   M,A
1658  439E  3C            4.0          INR   A
1659  439F  32  54  49    13.0         STA   INDAT
1660  43A2  23            6.0          INX   H
1661  43A3  3D            4.0          DCR   A                 ;RECREATE THE ORIGINAL DATA FOR THE ECC SUB
1662  43A4  CD  B9  48    18.0         CALL  ECC               ;CALCULATE ECC FOR
1663  43A7  15            4.0          DCR   D                 ;THE 7 INPUT BYTES
1664  43A8  C2  9A  43    10.0         JNZ   TST01F
1665  43AB  0E  03        7.0          MVI   C,@3              ;INIT THE LOOP COUNT
1666  43AD  3A  54  49    13.0  TST01I: LDA   INDAT            ;GET THE INPUT DATA
1667  43B0  06  25        7.0          MVI   B,@45
1668  43B2  FE  03        7.0          CPI   @3
1669  43B4  CA  C0  43    10.0         JZ    TST01Z
1670  43B7  06  00        7.0          MVI   B,0               ;CLEAR THE BYTE COUNT
1671  43B9  04            4.0  TST01W: INR   B
1672  43BA  D6  07        7.0          SUI   @7                ;DIVIDE BY 7
1673  43BC  A7            4.0          ANA   A
1674  43BD  C2  B9  43    10.0         JNZ   TST01W
1675  43C0  78            4.0  TST01Z: MOV   A,B
1676  43C1                             ROUT  R05L
 (1)  43C1  D3  8A        10.0                OUT   R05L              ;WRITE AC INTO R05L
```

```
  (1)  43C3  7F                  4.0                  MOV     A,A             ;RETRY LINK
1677   43C4  3A  5D  49          13.0          LDA    BUFFER
1678   43C7  32  A1  48          13.0          STA    TRNIN            ;FILL THE TRANSLATOR
1679   43CA  3A  5E  49          13.0          LDA    BUFFER+1
1680   43CD  32  A2  48          13.0          STA    TRNIN+1          ;SUBROUTINE INPUT
1681   43D0  3A  5F  49          13.0          LDA    BUFFER+2
1682   43D3  32  A3  48          13.0          STA    TRNIN+2          ;BUFFER
1683   43D6  3A  60  49          13.0          LDA    BUFFER+3
1684   43D9  32  A4  48          13.0          STA    TRNIN+3          ;
1685   43DC  CD  F5  48          18.0          CALL   TRK
1686   43DF  AF                   4.0          XRA    A                ;CLEAR THE ACCUMULATOR
1687   43E0  32  B3  48          13.0          STA    BAD3             ;CLEAR THE THIRD POINTER
1688   43E3  CD  CE  47          18.0          CALL   BADT4X5          ;TRANSLATE THE SUBGROUP
1689   43E6  21  A5  48          10.0          LXI    H,TRNOUT         ;GET POINTER TO TRANSLATE DATA TABLE
1690   43E9  06  05               7.0          MVI    B,@5             ;SET UP LOOP COUNT
1691   43EB  7E                   7.0  TST01G: MOV    A,M              ;GET A DATA BYTE
1692   43EC  D3  40              10.0          OUT    TCMD             ;STORE DATA IN COMMAND ADDRESS
1693   43EE  23                   6.0          INX    H                ;POINT TO DATA PARITY
1694   43EF  7E                   7.0          MOV    A,M              ;GET THE DATA PARITY
1695   43F0  07                   4.0          RLC                     ;POSITION FOR OUTPUT
1696   43F1  F6  60               7.0          ORI    P.LWR!P.LCS      ;OR IN CONTROL BITS
1697   43F3  D3  48              10.0          OUT    PDIAG            ;OUTPUT THE DATA PARITY
1698   43F5  D3  08              10.0          OUT    RFIFOL           ;CLOCK DATA INTO THE FIFO'S
1699   43F7  23                   6.0          INX    H                ;UPDATE THE TABLE POINTER
1700   43F8  05                   4.0          DCR    B                ;DECREMENT LOOP COUNT
1701   43F9  C2  EB  43          10.0          JNZ    TST01G           ;DO UNTIL LOOP COUNT = 0
1702   43FC  3A  61  49          13.0          LDA    BUFFER+4         ;GET THE INPUT DATA
1703   43FF  32  A1  48          13.0          STA    TRNIN            ;FILL THE FIRST THREE
1704   4402  3A  62  49          13.0          LDA    BUFFER+5
1705   4405  32  A2  48          13.0          STA    TRNIN+1          ;BYTES OF THE TRANSLATOR
1706   4408  3A  63  49          13.0          LDA    BUFFER+6
1707   440B  32  A3  48          13.0          STA    TRNIN+2          ;SUBROUTINE WITH THE INPUT DATA
1708   440E  3A  E2  48          13.0          LDA    ECCCHR           ;STORE THE CALCULATED
1709   4411  32  A4  48          13.0          STA    TRNIN+3          ;ECC CHARACTER AS THE LAST CHARACTER
1710   4414  CD  F5  48          18.0          CALL   TRK
1711   4417  AF                   4.0          XRA    A                ;CLEAR THE ACCUMULATOR
1712   4418  32  B3  48          13.0          STA    BAD3             ;CLEAR THE THIRD POINTER
1713   441B  CD  CE  47          18.0          CALL   BADT4X5          ;TRANSLATE THE SECOND SUBGROUP
1714   441E  21  A5  48          10.0          LXI    H,TRNOUT         ;GET THE POINTER TO INPUT DATA TABLE
1715   4421  06  05               7.0          MVI    B,@5             ;SET UP THE LOOP COUNT
1716   4423  7E                   7.0  TST01H: MOV    A,M              ;GET A DATA BYTE
1717   4424  D3  40              10.0          OUT    TCMD             ;STORE DATA IN CMD ADDRESS
1718   4426  23                   6.0          INX    H                ;POINT TO DATA PARITY BIT
1719   4427  7E                   7.0          MOV    A,M              ;GET THE DATA PARITY
1720   4428  07                   4.0          RLC                     ;POSITION FOR OUTPUT
1721   4429  F6  60               7.0          ORI    P.LWR!P.LCS      ;OR IN CONTROL BITS
1722   442B  D3  48              10.0          OUT    PDIAG            ;OUTPUT THE DATA PARITY
1723   442D  D3  08              10.0          OUT    RFIFOL           ;CLOCK DATA INTO THE FIFOS
1724   442F  23                   6.0          INX    H                ;UPDATE THE TABLE POINTER
1725   4430  05                   4.0          DCR    B                ;DECREMENT THE LOOP COUNT
1726   4431  C2  23  44          10.0          JNZ    TST01H           ;DO UNTIL LOOP COUNT=0
1727   4434  0D                   4.0          DCR    C                ;PUT THE GROUP IN
1728   4435  C2  AD  43          10.0          JNZ    TST011           ;THE FIFO TWICE
1729   4438  3E  0B               7.0          MVI    A,DIARD          ;LOAD THE DIAGNOSTIC READ
```

```
1730  443A  D3  0B        10.0          OUT   RCMD            ;COMMAND
1731  443C  11  01  00     10.0          LXI   D,1             ;SET WATCH DOG INCREMENT
1732  443F  21  A8  FD     10.0          LXI   H,-600          ;SET WATCH DOG COUNT TO 600
1733  4442  D3  JC        10.0  2$:     OUT   RINST           ;STEP THE READ PATH
1734  4444  DB  01        10.0          IN    RPCHI           ;DATA READY SET?
1735  4446  E6  10         7.0          ANI   R.DRDY
1736  4448  C2  5D  44     10.0          JNZ   TST01A          ;YES-GO PROCESS
1737  444B  19              10.0          DAD   D               ;WATCH DOG TIMEOUT?
1738  444C  D2  42  44      10.0          JNC   2$              ;NO-CONTINUE
1739  444F                                ERR   TST01C,DUMMW,1
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  444F  CD  09  28      18.0                CALL  ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0002                      MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4452  02                        .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  4453  01                        .BYTE 1
 (1)  4454  CD  15  28      18.0  DUMMW:: CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4457  DA  7F  43      10.0          JC    TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1740                        ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
1741                        ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
1742                        ;<FATAL ERROR - MICRO TEST ABORTED
1743  445A  C3  95  47      10.0          JMP   EXIT
1744
1745  445D  D3  0C        10.0  TST01A: OUT   RINST
1746  445F  06  00         7.0          MVI   B,$0            ;INIT THE BIT TIME COUNT (LOOP COUNT)
1747  4461  00             4.0  TST01B: NOP
1748  4462  3A  50  49      13.0          LDA   TIET1           ;GET FIRST TIE BUS JAM TIME
1749  4465  B8              4.0          CMP   B               ;IS IT TIME TO JAM THE TIE BUS?
1750  4466  C2  70  44      10.0          JNZ   4$              ;NO - CONTINUE
1751  4469  3E  08         7.0          MVI   A,$8            ;LOAD A BAD
1752  446B  D3  0A        10.0          OUT   RTIEB           ;4X5 POINTER
1753  446D  C3  82  44      10.0          JMP   5$              ;
1754  4470  3A  51  49      13.0  4$:     LDA   TIET2           ;GET SECOND TIE BUS JAM TIME
1755  4473  B8              4.0          CMP   B               ;IS IT TIME TO JAM THE TIE BUS?
1756  4474  C2  7E  44      10.0          JNZ   6$              ;NO-CONTINUE
1757  4477  3E  08         7.0          MVI   A,$8            ;LOAD A BAD
1758  4479  D3  0A        10.0          OUT   RTIEB           ;4X5 POINTER
1759  447B  C3  82  44      10.0          JMP   5$              ;
1760  447E  3E  02         7.0  6$:     MVI   A,@2            ;IF NOT THE BAD TRACK, PROVIDE A PHASE TIE POINTER
1761  4480  D3  0A        10.0          OUT   RTIEB           ;LOAD THE POINTER
1762  4482  D3  0C        10.0  5$:     OUT   RINST           ;STEP CLOCK
1763  4484  04             4.0          INR   B               ;DECREMENT THE LOOP COUNT
1764  4485  78             4.0          MOV   A,B
1765  4486  FE  09         7.0          CPI   $9              ;DONE
1766  4488  C2  61  44      10.0          JNZ   TST01B          ;DO UNTIL LOOP COUNT = ZERO
1767  448B  AF             4.0          XRA   A
1768  448C  D3  0A        10.0          OUT   RTIEB           ;CLEAR THE TIE BUS REGISTER
1769  448E  D3  0C        10.0          OUT   RINST           ;CLOCK THE READ PATH
1770  4490  D3  0C        10.0          OUT   RINST           ;CLOCK THE READ PATH
1771  4492  11  01  00     10.0          LXI   D,1             ;SET WATCHDOG TIMER INCREMENT
1772  4495  21  A8  FD     10.0          LXI   H,-600          ;SET WATCHDOG COUNT TO 600
1773  4498  D3  0C        10.0  3$:     OUT   RINST           ;SINGLE STEP THE READ PATH
1774  449A  DB  01        10.0          IN    RPCHI           ;DATA READY SET?
1775  449C  E6  10         7.0          ANI   R.DRDY
1776  449E  C2  B3  44      10.0          JNZ   TST01K          ;YES-GO PROCESS
```

```
1777  44A1  19                  10.0          DAD     D               ;NO-WATCHDOG TIMEOUT?
1778  44A2  D2  98  44           10.0          JNC     3$              ;NO-CONTINUE
1779  44A5                                    ERR     TST01C,DUMMX,1
 (1)                                          ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44A5  CD  09  28           18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0003                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THI
 (1)  44A8  03                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44A9  01                                .BYTE   1
 (1)  44AA  CD  15  28           18.0          DUMMX:: CALL    CK OP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44AD  DA  7F  43           10.0                  JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1780                                    ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
1781                                    ;>SENT TO THE ECC
1782                                    ;<FATAL ERROR - MICRO TEST ABORTED
1783  44B0  C3  95  47           10.0          JMP     EXIT
1784  44B3  16  01                 7.0    TST01K: MVI     D,@1            ;INIT THE LOOP COUNT
1785  44B5  21  5D  49            10.0          LXI     H,BUFFER
1786  44B8  00                     4.0    TST01Y: NOP                     ;
1787  44B9  7E                     7.0          MOV     A,M             ;GET THE INPUT DATA
1788  44BA                                      ROUT    EDATA
 (1)  44BA  D3  95                10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  44BC  7F                     4.0                  MOV     A,A             ;RETRY LINK
1789  44BD  A7                     4.0          ANA     A               ;SET CONDITION BITS
1790  44BE  3E  00                 7.0          MVI     A,0             ;CLEAR THE ACC.
1791  44C0  E2  C4  44            10.0          JPO     1$              ;ODD # OF ONES - CONTINUE
1792  44C3  3C                     4.0          INR     A               ;EVEN # OF ONES - SET PARITY
1793  44C4  32  55  49            13.0    1$:   STA     INDATP          ;STORE THE PARITY BIT
1794  44C7  D3  CC                10.0    TST01L: OUT     RINST           ;SINGLE STEP THE READ LOGIC
1795  44C9  7E                     7.0          MOV     A,M
1796  44CA  23                     6.0          INX     H
1797  44CB  47                     4.0          MOV     B,A
1798  44CC  DB  19                10.0          IN      ECCCOR
1799  44CE  B8                     4.0          CMP     B
1800  44CF  CA  DA  44            10.0          JZ      TST01M
1801  44D2                                      ROUT    ADATA
 (1)  44D2  D3  94                10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44D4  7F                     4.0                  MOV     A,A             ;RETRY LINK
1802  44D5                                    ERRB    TST01C,TST01M,@1
 (1)                                          ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44D5  CD  12  28           18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0004                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44D8  04                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44D9  01                                .BYTE   @1              ;PRINT ROUTINE NUMBER
 (1)  44DA  CD  15  28           18.0          TST01M::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44DD  DA  7F  43           10.0                  JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1803                                    ;>CORRECTED DATA FROM THE ECC-INCORRECT
1804  44E0  DB  15                10.0          IN      RPSTA           ;GET THE ECC PARITY BIT
1805  44E2  E6  80                 7.0          ANI     E.CDP           ;GET THE CORRECTED PARITY BIT
1806  44E4  07                     4.0          RLC                     ;POSITION BIT FOR COMPARE
1807  44E5  4F                     4.0          MOV     C,A             ;SAVE IN REG. C
1808  44E6  3A  55  49            13.0          LDA     INDATP          ;GET THE EXPECTED PARITY BIT
1809  44E9  A7                     4.0          ANA     A               ;SET THE CONDITION BITS
1810  44EA  CA  01  45            10.0          JZ      TST01QO         ;GO CHECK FOR PARITY=0
1811  44ED  79                     4.0          MOV     A,C             ;ELSE CHECK FOR PARITY=1
1812  44EE  FE  01                 7.0          CPI     @1
```

```
1813  44F0  CA  F8  44      10.0          JZ      TST01RG
1814  44F3                                ERR     TST01C,TST01RG,@1
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44F3  CD  09  28      18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0005                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44F6  05                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44F7  01                                      .BYTE   @1
 (1)  44F8  CD  15  28      18.0          TST01RG::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44FB  DA  7F  43      10.0                    JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1815                            ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
1816  44FE  C3  11  45      10.0          JMP     TST01SZ         ;CONTINUE WITH TEST
1817  4501  79                4.0  TST01QO: MOV     A,C             ;CHECK FOR PARITY = ZERO
1818  4502  A7                4.0          ANA     A               ;SET CONDITION BITS
1819  4503  CA  0B  45      10.0          JZ      TST01TP         ;CONTINUE IF ZERO
1820  4506                                ERR     TST01C,TST01TP,@1
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4506  CD  09  28      18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0006                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4509  06                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  450A  01                                      .BYTE   @1
 (1)  450B  CD  15  28      18.0          TST01TP::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  450E  DA  7F  43      10.0                    JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1821                            ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
1822  4511  7A                4.0  TST01SZ: MOV     A,D             ;GET THE CHARACTER COUNT
1823  4512  3C                4.0          INR     A               ;INCREMENT
1824  4513  FE  08            7.0          CPI     $8              ;DONE?
1825  4515  CA  1C  45      10.0          JZ      TST01N          ;YES-GO CHECK ECC
1826  4518  57                4.0          MOV     D,A             ;NO-CONTINUE
1827  4519  C3  B8  44      10.0          JMP     TST01Y
1828
1829  451C  D3  0C            10.0  TST01N: OUT     RINST           ;SINGLE STEP THE READ PATH
1830  451E  3A  E2  48        13.0          LDA     ECCCHR          ;GET THE EXPECTED ECC CHARACTER
1831  4521                                  ROUT    EDATA           ;SAVE IN CAS
 (1)  4521  D3  95            10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4523  7F                4.0                   MOV     A,A             ;RETRY LINK
1832  4524  47                4.0          MOV     B,A             ;SAVE IN REGISTER B
1833  4525  DB  19            10.0          IN      ECCCOR
1834  4527                                  ROUT    ADATA
 (1)  4527  D3  94            10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4529  7F                4.0                   MOV     A,A             ;RETRY LINK
1835  452A  B8                4.0          CMP     B
1836  452B  CA  33  45        10.0          JZ      TST01U
1837  452E                                  ERR     TST01C,TST01U,1
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  452E  CD  09  28      18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0007                                    MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4531  07                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4532  01                                      .BYTE   1
 (1)  4533  CD  15  28      18.0          TST01U::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4536  DA  7F  43      10.0                    JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1838                            ;>CORRECTED ECC CHARACTER INCORRECT
1839  4539  DB  1A            10.0          IN      ECCSTA          ;GET THE ECC STATUS
1840  453B  E6  0F             7.0          ANI     @17             ;REMOVE UNWANTED BITS
1841  453D  FE  02             7.0          CPI     E.TTEC          ;IS TWO TRACK CORRECT BIT SET?
```

```
1842  453F  CA  4F  45   10.0          JZ      TST01V
1843  4542                            ROUT    ADATA
 (1)  4542  D3  94        10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4544  7F             4.0                 MOV     A,A             ;RETRY LINK
1844  4545  3E  02         7.0          MVI     A,E.TTEC
1845  4547                            ROUT    EDATA
 (1)  4547  D3  95        10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4549  7F             4.0                 MOV     A,A             ;RETRY LINK
1846  454A                            ERRB    TST01C,TST01V,1
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  454A  CD  12  28    18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)      0008                             MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  454D  08                             .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  454E  01                             .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  454F  CD  15  28    18.0         TST01V::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4552  DA  7F  43    10.0                 JC      TST01C          ;LOOP ADDRESS IF LOOP SPECIFIED
1847                            ;>ECC STATUS INCORRECT AFTER A DATA GROUP
1848  4555  3A  54  49    13.0          LDA     INDAT
1849  4558  3C             4.0          INR     A
1850  4559  32  54  49    13.0          STA     INDAT
1851  455C  C2  7F  43    10.0          JNZ     TST01C
1852  455F                            ENDTST  TST01X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  455F                            REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  455F  CD  06  28    18.0                 CALL    REQST
 (2)  4562  00                             .BYTE                   ;DATA PATTERN NUMBER
 (2)  4563  00  00                         .WORD                   ;SYSTEM "" COUNT
 (2)  4565  00  00                         .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4567  00                             .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4568  07                             .BYTE   7               ;REQUEST CODE
 (1)  4569  3A  9A  4F    13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  456C  3D             4.0          DCR     A               ;DOWNCOUNT
 (1)  456D  32  9A  4F    13.0          STA     ITERA           ;SAVE COUNT
 (1)  4570  F2  51  43    10.0          JP      TST01X          ;DO TEST UNTIL TILL = 0
```

```
                        .SBTTL  TEST 2 - ECC BAD DATA TRACKS 1 AND 9 NINE SOFT POINTERS
4573                    ST
                        ; ***************************************************************************
                        ;*TEST TITLE
                        ;*---- -----
                        ;*ECC BAD DATA TRACKS 1 AND 9 NINE SOFT POINTERS
4573                    SD
                        ; ***************************************************************************
                        ;*DESCRIPTION
                        ;*-----------
                        ;*THIS TEST CHECKS THE ABILITY OF THE ECC CONTROLLER TO GIVE PRIORITY
                        ;*FOR DATA CORRECTION TO THE EDGE TRACKS (1 AND 9) WHEN GIVEN A MULTITUDE
                        ;*OF POINTERS WITH EQUAL VALUE.  THE PROGRAM FORCES BAD DATA ON TRACKS
                        ;*1 AND 9 (DATA BIT POSITIONS 2 AND 3) AND PRESENTS THE ECC WITH 9
                        ;*POINTERS OF EQUAL VALUE (SEE TABLE).  THE PROGRAM CHECKS FOR CORRECTED
                        ;*DATA AND THE TWO TRACK ERROR CORRECT STATUS BIT FROM THE ECC.
                        ;*
                        ;*      VALUE OF THE
                        ;*       BYTE COUNT                    POINTER GENERATED
                        ;*      ------------                   -----------------
                        ;*
                        ;*        XXXXXX00                        HISTORY
                        ;*        XXXXXX01                        PHASE IE
                        ;*        XXXXXX10                        AMTIE
                        ;*        XXXXXX11                        HISTORY
                        ;*
                        ;*      X = DON'T CARE
                        ;*
                        ;*      BYTE COUNT                          DATA
                        ;*      ----------                          ----
                        ;*
                        ;*         001          000,001,002,003,004,005,006
                        ;*         002          007,010,011,012,013,014,015
                        ;*         003          016,017,020,021,022,023,024
                        ;*         004          025,026,027,030,031,032,033
                        ;*         005          034,035,036,037,040,041 042
                        ;*         006          043,044,045,046,047,050,051
                        ;*         007          052,053,054,055,056,057,060
                        ;*         010          061,062,063,064,065,066,067
                        ;*         011          070,071,072,073,074,075,076
                        ;*         012          077,100,101,102,103,104,105
                        ;*         013          106,107,110,111,112,113,114
                        ;*         014          115,116,117,120,121,122,123
                        ;*         015          124,125,126,127,130,131,132
                        ;*         016          133,134,135,136,137,140,141
                        ;*         017          142,143,144,145,146,147,150
                        ;*         020          151,152,153,154,155,156,157
                        ;*         021          160,161,162,163,164,165,166
                        ;*         022          167,170,171,172,173,174,175
                        ;*         023          176,177,200,201,202,203,204
                        ;*         024          205,206,207,210,211,212,213
                        ;*         025          214,215,216,217,220,221,222
                        ;*         026          223,224,225,226,227,230,231
                        ;*         027          232,233,234,235,236,237,240
```

```
                        ;*          030                         241,242,243,244,245,246,247
                        ;*          031                         250,251,252,253,254,255,256
                        ;*          032                         257,260,261,262,263,264,265
                        ;*          033                         266,267,270,271,272,273,274
                        ;*          034                         275,276,277,300,301,302,303
                        ;*          035                         304,305,306,307,310,311,312
                        ;*          036                         313,314,315,316,317,320,321
                        ;*          037                         322,323,324,325,326,327,330
                        ;*          040                         331,332,333,334,335,336,337
                        ;*          041                         340,341,342,343,344,345,346
                        ;*          042                         347,350,351,352,353,354,355
                        ;*          043                         356,357,360,361,362,363,364
                        ;*          044                         365,366,367,370,371,372,373
                        ;*          045                         374,375,376,377,000,001,002
4573                    $P
 (1)                    ; ********************************************************************************
 (1)                    ;*PROCEDURE
 (1)                    ;*---------
1918                    ;*BGNTST
1919                    ;*    SET NORMAL READ PATH CLOCK
1920                    ;*    CALL SUBROUTINE CLEAR
1921                    ;*    SET USER SELECTED PORT TO LOOP COMMAND STATUS, WRITE/READ, AND DP ENABLE
1922                    ;*    CLEAR ALL AMTIE LINES FROM THE TU PORT SELECTED
1923                    ;*    DELAY 50 MICROSECONDS
1924                    ;*    CLEAR ALL TRACK ENABLE LINES FROM THE TRANSLATOR
1925                    ;*    CLEAR THE INPUT DATA
1926                    ;*    CLEAR BAD TRANSLATION POINTERS
1927                    ;*  BGNDO
1928                    ;*  : ENABLE THE READ PATH CLOCK
1929                    ;*  : SET PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
1930                    ;*  : CLOCK THE FIFO'S
1931                    ;*  : ISSUE CLEAR ALL COMMAND
1932                    ;*  : WAIT
1933                    ;*  : STOP THE READ PATH, SET TIE BUS JAM AND SET PLO DISABLE
1934                    ;*  : CALL SUBROUTINE CLECC
1935                    ;*  : SET THE LOOP COUNT TO 7
1936                    ;*  : BGNDO
1937                    ;*  : :  GET THE INPUT DATA BYTE
1938                    ;*  : :  STORE IN BUFFER
1939                    ;*  : :  CALL SUBROUTINE ECC
1940                    ;*  : :  DECREMENT THE LOOP COUNT
1941                    ;*  : :  INCREMENT THE INPUT DATA
1942                    ;*  : :  DO UNTIL THE LOOP COUNT=0
1943                    ;*  : ENDDO
1944                    ;*  : SET THE LOOP COUNT TO 3
1945                    ;*  : BGNDO
1946                    ;*  : :  FILL THE TRANSLATOR SUBROUTINE BUFFER FROM THE DATA BUFFER
1947                    ;*  : :  CALL SUBROUTINE T4X5
1948                    ;*  : :  INIT THE LOOP COUNT TO 5
1949                    ;*  : :  BGNDO
1950                    ;*  : :  :  MOVE 8 BITS OF TRANSLATED DATA TO THE TU PORT CMD REGISTER
1951                    ;*  : :  :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1952                    ;*  : :  :  MOVE THE TRANSLATED PARITY BIT TO THE TU PORT CONTROL REGISTER
1953                    ;*  : :  :  CLOCK THE DATA INTO THE FIFO
```

```
1954                    ;*  : : :  DECREMENT THE LOOP COUNT
1955                    ;*  : : :  INCREMENT THE TRANSLATED DATA BUFFER ADDRESS
1956                    ;*  : : :  DO UNTIL LOOP COUNT = 0
1957                    ;*  : : ENDDO
1958                    ;*  : : FILL THREE BYTES OF TRANSLATOR SUBROUTINE INPUT BUFFER FROM THE DATA BUFFER
1959                    ;*  : : FILL THE FOURTH BYTE OF TRANSLATOR SUBROUTINE INPUT BUFFER WITH THE ECC CHARACT
1960                    ;*  : : CALL SUBROUTINE T4X5
1961                    ;*  : : INIT THE LOOP COUNT TO 5
1962                    ;*  : : BGNDO
1963                    ;*  : : :  MOVE 8 BITS OF DATA FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CMD REG.
1964                    ;*  : : :  MOVE THE PARITY BIT FROM THE TRANSLATED DATA BUFFER TO THE TU PORT CONTROL R
1965                    ;*  : : :  CLOCK THE DATA INTO THE FIFO
1966                    ;*  : : :  DECREMENT THE LOOP COUNT
1967                    ;*  : : :  DO UNTIL THE LOOP COUNT = 0
1968                    ;*  : : ENDDO
1969                    ;*  : : DECREMENT THE LOOP COUNT
1970                    ;*  : : DO UNTIL THE LOOP COUNT = 0
1971                    ;*  : ENDDO
1972                    ;*  : ISSUE DIAGNOSTIC READ COMMAND
1973                    ;*  : SET UP WATCHDOG TIMER COUNT
1974                    ;*  : BGNDO
1975                    ;*  : :    SINGLE STEP THE READ PATH
1976                    ;*  : :    DECREMENT THE WATCHDOG TIMER
1977                    ;*  : :    DO UNTIL WATCHDOG TIMER = 0 OR DATA READY SETS
1978                    ;*  : ENDDO
1979                    ;*  : IF WATCHDOG TIMER=0
1980                    ;*  : :    THEN-ERROR
1981                    ;*  : :    ELSE-CONTINUE
1982                    ;*  : ENDIF
1983                    ;*  : GET THE TIE BUS TABLE ADDRESS
1984                    ;*  : SINGLE STEP THE READ PATH
1985                    ;*  : INIT THE LOOP COUNT TO 0
1986                    ;*  : BGNDO
1987                    ;*  : :    IF LOOP COUNT=TIE BUS JAM COUNT 1, 2 OR 3
1988                    ;*  : :    :  THEN-LOAD TIE BUS WITH 10(8)
1989                    ;*  : :    :  ELSE-CLEAR TIE BUS POINTER
1990                    ;*  : :    ENDIF
1991                    ;*  : :    SINGLE STEP THE READ PATH
1992                    ;*  : :    DECREMENT THE LOOP COUNT
1993                    ;*  : :    DO UNTIL LOOP COUNT = 9
1994                    ;*  : ENDDO
1995                    ;*  : SET UP WATCHDOG TIMER COUNT
1996                    ;*  : BGNDO
1997                    ;*  : :  SINGLE STEP THE READ PATH
1998                    ;*  : :  DECREMENT THE WATCHDOG TIMER COUNT
1999                    ;*  : :  DO UNTIL WATCHDOG TIMER=0 OR DATA READY SETS
2000                    ;*  : ENDDO
2001                    ;*  : IF WATCHDOG TIMER=0
2002                    ;*  : :  THEN-ERROR-EXIT TEST
2003                    ;*  : :  ELSE-CONTINUE
2004                    ;*  : ENDIF
2005                    ;*  : SINGLE STEP THE READ PATH
2006                    ;*  : INIT THE  LOOP COUNT TO 7
2007                    ;*  : BGNDO
```

```
                                ;*   : :  SINGLE STEP THE READ PATH
                                ;*   : :  COMPARE CORRECTED DATA WITH INPUT DATA
                                ;*   : :  IF NOT EQUAL
                                ;*   : :  :  THEN-ECC ERROR
                                ;*   : :  :  ELSE-CONTINUE
                                ;*   : :  ENDIF
                                ;*   : :  COMPARE CORRECTED PARITY WITH INPUT PARITY
                                ;*   : :  IF NOT EQUAL
                                ;*   : :  :  THEN-ECC ERROR
                                ;*   : :  :  ELSE-CONTINUE
                                ;*   : :  ENDIF
                                ;*   : :  DECREMENT THE LOOP COUNT
                                ;*   : :  DO UNTIL THE LOOP COUNT = 0
                                ;*   :.ENDDO
                                ;*   : SINGLE STEP THE READ PATH
                                ;*   : IF EXPECTED NOT EQUAL
                                ;*   : :  THEN-ECC ERROR
                                ;*   : :  ELSE-CONTINUE
                                ;*   : ENDIF
                                ;*   : INPUT THE ECC STATUS
                                ;*   : CLEAR THE CORRECTED DATA PARITY BIT IN THE STATUS BYTE
                                ;*   : IF THE RESULTING STATUS=SINGLE TRACK ERROR CORRECT + POINTER ERROR
                                ;*   : :  THEN-CONTINUE
                                ;*   : :  ELSE-ERROR
                                ;*   : ENDIF
                                ;*   : INCREMENT THE INPUT DATA
                                ;*   : DO UNTIL INPUT DATA = 0
                                ;*  ENDDO
                                ;*ENDTST
2037  4573                      SE
 (1)                            ; *****************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*ECC5 MICRO TEST 02
                                ;*ECC5 MICRO ERROR 11
                                ;*ECC5-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
                                ;*M8951, M8950'S
                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET
                                ;*AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
                                ;*BYTE/SCLK COUNT NUMBER = LLL
                                ;*FATAL ERROR - MICRO TEST ABORTED
                                ;*
                                ;*ECC5 MICRO TEST 02
                                ;*ECC5 MICRO ERROR 12
                                ;*ECC5-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
                                ;*M8951, M8950'S
                                ;*TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
                                ;*SENT TO THE ECC
                                ;*BYTE/SCLK COUNT NUMBER = LLL
                                ;*FATAL ERROR - MICRO TEST ABORTED
                                ;*
                                ;*ECC5 MICRO TEST 02
                                ;*ECC5 MICRO ERROR 13
                                ;*ECC5-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
```

```
2059                                    ;*M8951, M8950'S
2060                                    ;*DATA FROM THE ECC-INCORRECT
2061                                    ;*DATA SHOULD NOT BE CHANGED
2062                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2063                                    ;*ACTUAL = NNNN
2064                                    ;*EXPECTED = NNNN
2065                                    ;*
2066                                    ;*ECC5 MICRO TEST 02
2067                                    ;*ECC5 MICRO ERROR 14
2068                                    ;*ECC5-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2069                                    ;*M8951, M8950'S
2070                                    ;*CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
2071                                    ;*PARITY SHOULD NOT BE CHANGED
2072                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2073                                    ;*
2074                                    ;*ECC5 MICRO TEST 02
2075                                    ;*ECC5 MICRO ERROR 15
2076                                    ;*ECC5-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2077                                    ;*M8951, M8950'S
2078                                    ;*CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
2079                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2080                                    ;*
2081                                    ;*ECC5 MICRO TEST 02
2082                                    ;*ECC5 MICRO ERROR 16
2083                                    ;*ECC5-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2084                                    ;*M8951, M8950'S
2085                                    ;*CORRECTED ECC CHARACTER INCORRECT
2086                                    ;*ECC CHARACTER SHOULD NOT BE CHANGED
2087                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2088                                    ;*
2089                                    ;*ECC5 MICRO TEST 02
2090                                    ;*ECC5 MICRO ERROR 17
2091                                    ;*ECC5-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2092                                    ;*M8951, M8950'S
2093                                    ;*ECC STATUS INCORRECT AFTER A DATA GROUP
2094                                    ;*BYTE/SCLK COUNT NUMBER = LLL
2095                                    ;*ACTUAL = NNNN
2096                                    ;*EXPECTED = NNNN
2097  4573                              $
 (1)                                    ; ***************************************************************************
2098
2099  4573                              TEST2:  TESTX   @2              ;INITIALIZE THE TEST
 (1)  4573  3E  02          7.0                  MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4575  CD  03  28     18.0                  CALL    TSET            ;SETUP THE TEST
2100                                    ;%ECC5-ECC CONTROLLER-GOOD DATA-THREE HARD POINTERS
2101                                    ;&M8951, M8950'S
2102
2103  4578  3A  53  49     13.0          LDA     SUNIT
2104  457B  D3  E0         10.0          OUT     MBSEL
2105  457D  3E  10          7.0  TST02X: MVI     A,RDCLK         ;SET NORMAL READ PATH CLOCKS
2106  457F  D3  F0         10.0          OUT     CLKCTL
2107  4581  3E  10          7.0          MVI     A,W.GCR
2108  4583  D3  D3         10.0          OUT     WMCCTL
2109  4585  3E  04          7.0          MVI     A,P.RPEN
```

```
2110  4587  D3  4C          10.0          OUT   PENAB
2111
2112  4589  3E  60           7.0          MVI   A,P.LWR.P.LCS   ;SET THE PORT CONTROL TO
2113  458B  D3  48          10.0          OUT   PDIAG           ;LCS, LWR
2114  458D  AF               4.0          XRA   A
2115  458E  D3  44          10.0          OUT   TAMT
2116  4590  3E  0A           7.0          MVI   A,10            ;DELAY
2117  4592  3D               4.0    1$:   DCR   A               ;50
2118  4593  C2  92  45      10.0          JNZ   1$              ;MICROSECONDS
2119  4596  AF               4.0          XRA   A               ;CLEAR THE ACCUMULATOR
2120  4597  D3  D2          10.0          OUT   TRKENA          ;CLEAR ALL TRACKS FROM
2121  4599  D3  0A          10.0          OUT   RTIEB           ;CLEAR THE TIE BUS REGISTER
2122  459B  D3  D2          10.0          OUT   TRKENA
2123  459D  32  54  49      13.0          STA   INDAT           ;CLEAR THE INPUT DATA BYTE
2124  45A0                                ROUT  R05H
  (1) 45A0  D3  8B          10.0                OUT   R05H            ;WRITE AC INTO R05H
  (1) 45A2  7F               4.0                MOV   A,A             ;RETRY LINK
2125  45A3  32  B3  48      13.0          STA   BAD3            ;NO THIRD FAILURE
2126  45A6  3E  03           7.0          MVI   A,3
2127  45A8  32  B1  48      13.0          STA   BAD1            ;DATA BIT 2 BAD
2128  45AB  3E  04           7.0          MVI   A,4
2129  45AD  32  B2  48      13.0          STA   BAD2            ;DATA BIT 3 BAD
2130  45B0  3E  A8           7.0    TST02C: MVI  A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
2131                                                            ;SET TIE BUS JAM AND PLO DISABLE
2132  45B2  D3  09          10.0          OUT   RPCTL
2133  45B4  D3  08          10.0          OUT   RFIFOL          ;CLOCK THE FIFO'S
2134  45B6  3E  0D           7.0          MVI   A,RCLRT         ;ISSUE CLEAR ALL COMMAND
2135  45B8  D3  0B          10.0          OUT   RCMD
2136
2137  45BA  00               4.0          NOP                   ;WAIT
2138  45BB  00               4.0          NOP
2139  45BC  00               4.0          NOP
2140  45BD  00               4.0          NOP
2141  45BE  00               4.0          NOP
2142
2143  45BF  3E  A9           7.0          MVI   A,R.PLO1!R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
2144                                                            ;SET TIE BUS JAM AND PLO DISABLE
2145  45C1  D3  09          10.0          OUT   RPCTL
2146  45C3  CD  B4  48      18.0          CALL  CLECC           ;CLEAR THE ECC CHARACTER
2147  45C6  21  5D  49      10.0          LXI   H,BUFFER         ;SET UP THE BUFFER POINTER
2148  45C9  16  07           7.0          MVI   D,@7            ;SET UP THE LOOP COUNT
2149  45CB  3A  54  49      13.0    TST02F: LDA  INDAT          ;GET THE INPUT DATA
2150  45CE  77               7.0          MOV   M,A             ;SAVE CHARACTER IN THE BUFFER
2151  45CF  3C               4.0          INR   A               ;UPDATE THE DATA
2152  45D0  32  54  49      13.0          STA   INDAT           ;SAVE IT
2153  45D3  23               6.0          INX   H               ;UPDATE THE BUFFER POINTER
2154  45D4  3D               4.0          DCR   A               ;RECREATED THE DATA FOR THE ECC SUBROUTINE
2155  45D5  CD  B9  48      18.0          CALL  ECC             ;CALCULATE ECC FOR
2156  45D8  15               4.0          DCR   D               ;THE 7 INPUT BYTES
2157  45D9  C2  CB  45      10.0          JNZ   TST02F
2158
2159  45DC  0E  03           7.0          MVI   C,@3            ;INIT THE LOOP COUNT
2160  45DE  3A  54  49      13.0    TST02I: LDA  INDAT          ;GET THE INPUT DATA
2161  45E1  06  25           7.0          MVI   B,@45
```

```
2162  45E3  FE  03        7.0          CPI   @3
2163  45E5  CA  F1  45   10.0          JZ    TST02Z
2164  45E8  06  00        7.0          MVI   B,0
2165  45EA  04            4.0  TST02W: INR   B
2166  45EB  D6  07        7.0          SUI   @7
2167  45ED  A7            4.0          ANA   A
2168  45EE  C2  EA  45   10.0          JNZ   TST02W
2169  45F1  78            4.0  TST02Z: MOV   A,B
2170  45F2                             ROUT  R05L
 (1)  45F2  D3  8A        10.0                OUT   R05L          ;WRITE AC INTO R05L
 (1)  45F4  7F            4.0                 MOV   A,A           ;RETRY LINK
2171  45F5  32  4F  49   13.0          STA   TEMPX
2172  45F8  3A  5D  49   13.0          LDA   BUFFER
2173  45FB  32  A1  48   13.0          STA   TRNIN               ;FILL THE TRANSLATOR
2174  45FE  3A  5E  49   13.0          LDA   BUFFER+1
2175  4601  32  A2  48   13.0          STA   TRNIN+1             ;SUBROUTINE INPUT
2176  4604  3A  5F  49   13.0          LDA   BUFFER+2
2177  4607  32  A3  48   13.0          STA   TRNIN+2             ;BUFFER
2178  460A  3A  60  49   13.0          LDA   BUFFER+3
2179  460D  32  A4  48   13.0          STA   TRNIN+3             ;
2180  4610  CD  CE  47   18.0          CALL  BADT4X5             ;TRANSLATE THE SUBGROUP
2181
2182  4613  21  A5  48   10.0          LXI   H,TRNOUT            ;GET POINTER TO TRANSLATED DATA TABLE
2183  4616  06  05        7.0          MVI   B,@5                ;SET UP LOOP COUNT
2184  4618  7E            7.0  TST02G: MOV   A,M                 ;GET A DATA BYTE
2185  4619  D3  40       10.0          OUT   TCMD                ;STORE DATA IN COMMAND ADDRESS
2186  461B  23            6.0          INX   H                   ;POINT TO DATA PARITY
2187  461C  7E            7.0          MOV   A,M                 ;GET THE DATA PARITY
2188
2189  461D  07            4.0          RLC                       ;POSITION FOR OUTPUT
2190  461E  F6  60        7.0          ORI   P.LWR!P.LCS         ;OR IN CONTROL BITS
2191  4620  D3  48       10.0          OUT   PDIAG               ;OUTPUT THE DATA PARITY
2192
2193  4622  D3  08       10.0          OUT   RFIFOL              ;CLOCK DATA INTO THE FIFO'S
2194
2195  4624  23            6.0          INX   H                   ;UPDATE THE TABLE POINTER
2196  4625  05            4.0          DCR   B                   ;DECREMENT LOOP COUNT
2197  4626  C2  18  46   10.0          JNZ   TST02G              ;DO UNTIL LOOP COUNT = 0
2198
2199  4629  3A  61  49   13.0          LDA   BUFFER+4
2200  462C  32  A1  48   13.0          STA   TRNIN               ;FILL THE FIRST THREE
2201  462F  3A  62  49   13.0          LDA   BUFFER+5
2202  4632  32  A2  48   13.0          STA   TRNIN+1             ;BYTES OF THE TRANSLATOR
2203  4635  3A  63  49   13.0          LDA   BUFFER+6
2204  4638  32  A3  48   13.0          STA   TRNIN+2             ;SUBROUTINE WITH THE INPUT DATA
2205  463B  3A  E2  48   13.0          LDA   ECCCHR              ;STORE THE CALCULATED
2206  463E  32  A4  48   13.0          STA   TRNIN+3             ;ECC CHARACTER AS THE LAST CHARACTER
2207
2208  4641  CD  CE  47   18.0          CALL  BADT4X5             ;TRANSLATE THE SECOND SUBGROUP
2209
2210  4644  21  A5  48   10.0          LXI   H,TRNOUT            ;GET THE POINTER TO INPUT DATA TABLE
2211  4647  06  05        7.0          MVI   B,@5                ;SET UP THE LOOP COUNT
2212  4649  7E            7.0  TST02H: MOV   A,M                 ;GET A DATA BYTE
2213  464A  D3  40       10.0          OUT   TCMD                ;STORE DATA IN CMD ADDRESS
```

```
2214  464C  23              6.0          INX   H                ;POINT TO DATA PARITY BIT
2215  464D  7E              7.0          MOV   A,M              ;GET THE DATA PARITY
2216  464E  07              4.0          RLC                    ;POSITION FOR OUTPUT
2217  464F  F6  60          7.0          ORI   P.LWR!P.LCS      ;OR IN CONTROL BITS
2218  4651  D3  48         10.0          OUT   PDIAG            ;OUTPUT THE DATA PARITY
2219  4653  D3  08         10.0          OUT   RFIFOL           ;CLOCK DATA INTO THE FIFOS
2220  4655  23              6.0          INX   H                ;UPDATE THE TABLE POINTER
2221  4656  05              4.0          DCR   B                ;DECREMENT THE LOOP COUNT
2222  4657  C2  49  46      10.0          JNZ   TST02H           ;DO UNTIL LOOP COUNT=0
2223  465A  0D              4.0          DCR   C                ;PUT THE GROUP IN
2224  465B  C2  DE  45      10.0          JNZ   TST02I           ;THE FIFO TWICE
2225
2226  465E  3E  0B          7.0          MVI   A,DIARD          ;LOAD THE DIAGNOSTIC READ
2227  4660  D3  0B         10.0          OUT   RCMD             ;COMMAND
2228  4662  11  01  00     10.0          LXI   D,1              ;SET WATCH DOG INCREMENT
2229  4665  21  A8  FD     10.0          LXI   H,-600           ;SET WATCH DOG COUNT TO 600
2230  4668  D3  0C         10.0  2$:     OUT   RINST            ;STEP THE READ PATH
2231  466A  DB  01         10.0          IN    RPCHI            ;DATA READY SET?
2232  466C  E6  10          7.0          ANI   R.DRDY
2233  466E  C2  83  46      10.0          JNZ   TST02A           ;YES-GO PROCESS
2234  4671  19              10.0          DAD   D                ;WATCH DOG TIMEOUT?
2235  4672  D2  68  46      10.0          JNC   2$               ;NO-CONTINUE
2236  4675                               ERR   RETRY,DUMMU,1
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4675  CD  09  28      18.0                CALL  ERLP             ;PROCESS ERROR - DO 2.3
 (1)       0009                         MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4678  09                          .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  4679  01                          .BYTE 1
 (1)  467A  CD  15  28      18.0  DUMMU:: CALL  CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  467D  DA  98  47      10.0          JC    RETRY            ;LOOP ADDRESS IF LOOP SPECIFIED
2237                                     ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET
2238                                     ;>AFTER ISSUEING A READ COMMAND TO THE READ CHANNEL
2239                                     ;<FATAL ERROR - MICRO TEST ABORTED
2240  4680  C3  95  47      10.0          JMP   EXIT
2241
2242  4683  D3  0C         10.0  TST02A: OUT   RINST
2243  4685  06  00          7.0          MVI   B,$0             ;INIT THE BIT TIME COUNT (LOOP COUNT)
2244  4687  00              4.0  TST02B: NOP
2245  4688  CD  79  48      18.0  6$:     CALL  POINTER
2246  468B  D3  0C         10.0          OUT   RINST            ;STEP CLOCK
2247  468D  04              4.0          INR   B                ;DECREMENT THE LOOP COUNT
2248  468E  78              4.0          MOV   A,B
2249  468F  FE  09          7.0          CPI   $9               ;DONE
2250  4691  C2  87  46      10.0          JNZ   TST02B           ;DO UNTIL LOOP COUNT = ZERO
2251  4694  AF              4.0          XRA   A
2252  4695  D3  0A         10.0          OUT   RTIEB            ;CLEAR THE TIE BUS REGISTER
2253  4697  D3  0C         10.0          OUT   RINST            ;CLOCK THE READ PATH
2254  4699  D3  0C         10.0          OUT   RINST            ;CLOCK THE READ PATH
2255  469B  11  01  00     10.0          LXI   D,1              ;SET WATCHDOG TIMER INCREMENT
2256  469E  21  A8  FD     10.0          LXI   H,-600           ;SET WATCHDOG COUNT TO 600
2257  46A1  D3  0C         10.0  3$:     OUT   RINST            ;SINGLE STEP THE READ PATH
2258  46A3  DB  01         10.0          IN    RPCHI            ;DATA READY SET?
2259  46A5  E6  10          7.0          ANI   R.DRDY
2260  46A7  C2  BC  46      10.0          JNZ   TST02K           ;YES-GO PROCESS
```

```
2261  46AA  19                    10.0          DAD     D                ;NO-WATCHDOG TIMEOUT?
2262  46AB  D2  A1  46            10.0          JNC     3$               ;NO-CONTINUE
2263  46AE                                      ERR     RETRY,DUMMV,1
 (1)                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  46AE  CD  09  28            18.0                  CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)       000A                                         MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46B1  0A                                          .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  46B2  01                                          .BYTE   1
 (1)  46B3  CD  15  28            18.0          DUMMV:: CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  46B6  DA  98  47            10.0                  JC      RETRY            ;LOOP ADDRESS IF LOOP SPECIFIED
2264                          ;>TIMEOUT WHILE WAITING FOR DATA READY TO SET AFTER ONE GROUP HAS BEEN
2265                          ;>SENT TO THE ECC
2266                          ;<FATAL ERROR - MICRO TEST ABORTED
2267  46B9  C3  95  47            10.0          JMP     EXIT
2268  46BC  16  01                 7.0   TST02K: MVI     D,@1             ;INIT THE LOOP COUNT
2269  46BE  21  5D  49             10.0          LXI     H,BUFFER
2270  46C1  AF                     4.0   TST02Y: XRA     A
2271  46C2                                      ROU     R05H
 (1)  46C2  D3  8B                 10.0                  OUT     R05H             ;WRITE AC INTO R05H
 (1)  46C4  7F                     4.0                   MOV     A,A              ;RETRY LINK
2272  46C5  7E                     7.0           MOV     A,M              ;GET THE INPUT DATA
2273  46C6                                      ROUT    EDATA
 (1)  46C6  D3  95                 10.0                  OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  46C8  7F                     4.0                   MOV     A,A              ;RETRY LINK
2274  46C9  A7                     4.0           ANA     A                ;SET CONDITION BITS
2275  46CA  3E  00                 7.0           MVI     A,0              ;CLEAR THE ACC.
2276  46CC  E2  D0  46             10.0          JPO     1$               ;ODD # OF ONES - CONTINUE
2277  46CF  3C                     4.0           INR     A                ;EVEN # OF ONES - SET PARITY
2278  46D0  32  55  49             13.0  1$:     STA     INDATP           ;STORE THE PARITY BIT
2279  46D3  D3  0C                 10.0  TST02L: OUT     RINST            ;SINGLE STEP THE READ LOGIC
2280  46D5  7E                     7.0           MOV     A,M
2281  46D6  23                     6.0           INX     H
2282  46D7  47                     4.0           MOV     B,A
2283  46D8  DB  19                 10.0          IN      ECCCOR
2284  46DA  B8                     4.0           CMP     B
2285  46DB  CA  EA  46             10.0          JZ      TST02M
2286  46DE                                      ROUT    ADATA
 (1)  46DE  D3  94                 10.0                  OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  46E0  7F                     4.0                   MOV     A,A              ;RETRY LINK
2287  46E1  78                     4.0           MOV     A,B              ;GET THE BYTE COUNT
2288  46E2                                      ROUT    R05L             ;STORE IN CAS
 (1)  46E2  D3  8A                 10.0                  OUT     R05L             ;WRITE AC INTO R05L
 (1)  46E4  7F                     4.0                   MOV     A,A              ;RETRY LINK
2289  46E5                                      ERRB    RETRY,TST02M,@1
 (1)                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46E5  CD  12  28            18.0                  CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)       000B                                         MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46E8  0B                                          .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  46E9  01                                          .BYTE   @1               ;PRINT ROUTINE NUMBER
 (1)  46EA  CD  15  28            18.0          TST02M::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46ED  DA  98  47            10.0                  JC      RETRY            ;LOOP ADDRESS IF LOOP SPECIFIED
2290                          ;>DATA FROM THE ECC-INCORRECT
2291                          ;>DATA SHOULD NOT BE CHANGED
2292  46F0  DB  15                 10.0           IN      RPSTA            ;GET THE ECC PARITY BIT
```

```
2293  46F2  E6  80        7.0          ANI     E.CDP           ;GET THE CORRECTED PARITY BIT
2294  46F4  07            4.0          RLC                     ;POSITION BIT FOR COMPARE
2295  46F5  4F            4.0          MOV     C,A             ;SAVE IN REG. C
2296  46F6  3A  55  49     13.0         LDA     INDATP          ;GET THE EXPECTED PARITY BIT
2297  46F9  A7            4.0          ANA     A               ;SET THE CONDITION BITS
2298  46FA  CA  15  47     10.0         JZ      TST02Q0         ;GO CHECK FOR PARITY=0
2299  46FD  79            4.0          MOV     A,C             ;ELSE CHECK FOR PARITY=1
2300  46FE  FE  01        7.0          CPI     @1
2301  4700  CA  0C  47     10.0         JZ      TST02RG
2302  4703  78            4.0          MOV     A,B
2303  4704                             ROUT    RO5L
 (1)  4704  D3  8A        10.0                 OUT     RO5L            ;WRITE AC INTO RO5L
 (1)  4706  7F            4.0                  MOV     A,A             ;RETRY LINK
2304  4707                             ERR     RETRY,TST02RG,@1
 (1)                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4707  CD  09  28     18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)      000C                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  470A  0C                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  470B  01                                .BYTE   @1
 (1)  470C  CD  15  28     18.0         TST02RG::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  470F  DA  98  47     10.0                 JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2305                             ;>CORRECTED DATA PARITY BIT = ZERO SHOULD = ONE
2306                             ;>PARITY SHOULD NOT BE CHANGED
2307  4712  C3  29  47     10.0         JMP     TST02SZ         ;CONTINUE WITH TEST
2308  4715  79            4.0  TST02Q0: MOV     A,C             ;CHECK FOR PARITY = ZERO
2309  4716  A7            4.0          ANA     A               ;SET CONDITION BITS
2310  4717  CA  23  47     10.0         JZ      TST02TP         ;CONTINUE IF ZERO
2311  471A  78            4.0          MOV     A,B
2312  471B                             ROUT    RO5L
 (1)  471B  D3  8A        10.0                 OUT     RO5L            ;WRITE AC INTO RO5L
 (1)  471D  7F            4.0                  MOV     A,A             ;RETRY LINK
2313  471E                             ERR     RETRY,TST02TP,@1
 (1)                                 ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  471E  CD  09  28     18.0                 CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)      000D                                MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4721  0D                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4722  01                                .BYTE   @1
 (1)  4723  CD  15  28     18.0         TST02TP::       CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4726  DA  98  47     10.0                 JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2314                             ;>CORRECTED DATA PARITY BIT = ONE SHOULD = ZERO
2315  4729  7A            4.0  TST02SZ: MOV     A,D             ;GET THE CHARACTER COUNT
2316  472A  3C            4.0          INR     A               ;INCREMENT
2317  472B  FE  08        7.0          CPI     $8              ;DONE?
2318  472D  CA  34  47     10.0         JZ      TST02N          ;YES-GO CHECK ECC
2319  4730  57            4.0          MOV     D,A             ;NO-CONTINUE
2320  4731  C3  C1  46     10.0         JMP     TST02Y
2321
2322  4734  D3  0C        10.0 TST02N: OUT     RINST           ;SINGLE STEP THE READ PATH
2323  4736  3A  E2  48     13.0         LDA     ECCCHR          ;GET THE EXPECTED ECC CHARACTER
2324  4739                             ROUT    EDATA           ;SAVE IN CAS
 (1)  4739  D3  95        10.0                 OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  473B  7F            4.0                  MOV     A,A             ;RETRY LINK
2325  473C  47            4.0          MOV     B,A             ;SAVE IN REGISTER B
2326  473D  DB  19        10.0         IN      ECCCOR
```

```
2327  473F                                    ROUT    ADATA
 (1)  473F  D3  94          10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4741  7F               4.0                     MOV     A,A             ;RETRY LINK
2328  4742  B8               4.0             CMP     B
2329  4743  CA  51  47       10.0             JZ      TST02U
2330  4746  3A  54  49       13.0             LDA     INDAT           ;GET THE DATA/COUNT
2331  4749                                    ROUT    RO5L            ;SAVE IN THE CAS
 (1)  4749  D3  8A          10.0                     OUT     RO5L            ;WRITE AC INTO RO5L
 (1)  474B  7F               4.0                     MOV     A,A             ;RETRY LINK
2332  474C                                    ERR     RETRY,TST02U,1
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  474C  CD  09  28       18.0                     CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        000E                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  474F  0E                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4750  01                                        .BYTE   1
 (1)  4751  CD  15  28       18.0             TST02U::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4754  DA  98  47       10.0                     JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2333                                    ;>CORRECTED ECC CHARACTER INCORRECT
2334                                    ;>ECC CHARACTER SHOULD NOT BE CHANGED
2335  4757  DB  1A             10.0             IN      ECCSTA          ;GET THE ECC STATUS
2336  4759  E6  0F              7.0             ANI     @17             ;REMOVE UNWANTED BITS
2337  475B  FE  02              7.0             CPI     E.TTEC          ;IS UNCORRECTABLE BIT SET?
2338  475D  CA  73  47          10.0             JZ      TST02V
2339  4760                                    ROUT    ADATA
 (1)  4760  D3  94             10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4762  7F                  4.0                     MOV     A,A             ;RETRY LINK
2340  4763  3E  02              7.0             MVI     A,E.TTEC
2341  4765                                    ROUT    EDATA
 (1)  4765  D3  95             10.0                     OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4767  7F                  4.0                     MOV     A,A             ;RETRY LINK
2342  4768  3A  54  49          13.0             LDA     INDAT           ;GET THE DATA/COUNT
2343  476B                                    ROUT    RO5L            ;SAVE IN THE CAS
 (1)  476B  D3  8A             10.0                     OUT     RO5L            ;WRITE AC INTO RO5L
 (1)  476D  7F                  4.0                     MOV     A,A             ;RETRY LINK
2344  476E                                    ERRB    RETRY,TST02V,1
 (1)                                        ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  476E  CD  12  28       18.0                     CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000F                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4771  0F                                        .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4772  01                                        .BYTE   1               ;PRINT ROUTINE NUMBER
 (1)  4773  CD  15  28       18.0             TST02V::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4776  DA  98  47       10.0                     JC      RETRY           ;LOOP ADDRESS IF LOOP SPECIFIED
2345                                    ;>ECC STATUS INCORRECT AFTER A DATA GROUP
2346  4779  3A  54  49          13.0             LDA     INDAT
2347  477C  FE  03              7.0             CPI     @3              ;DONE
2348  477E  C2  B0  45          10.0             JNZ     TST02C
```

```
2350  4781                                    ENDTST  TST02X
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4781                                   REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4781  CD  06  28       18.0                            CALL    REQST
 (2)  4784  00                               .BYTE                   ;DATA PATTERN NUMBER
 (2)  4785  00  00                           .WORD                   ;SYSTEM '*' COUNT
 (2)  4787  00  00                           .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4789  00                               .BYTE                   ;DATA COMPARE FLAG ,F =1
 (2)  478A  07                               .BYTE   7               ;REQUEST CODE
 (1)  478B  3A  9A  4F       13.0            LDA     ITERA           ;GET ITERATION COUNT
 (1)  478E  3D                4.0            DCR     A               ;DOWNCOUNT
 (1)  478F  32  9A  4F       13.0            STA     ITERA           ;SAVE COUNT
 (1)  4792  F2  7D  45       10.0            JP      TST02X          ;DO TEST UNTIL TILL = 0
2351
2352  4795  C3  18  28       10.0  EXIT:     JMP     TSTEND
2353  4798  3A  54  49       13.0  RETRY:    LDA     INDAT
2354  479B  D6  07             7.0           SUI     @7
2355  479D  32  54  49       13.0            STA     INDAT
2356  47A0  C3  B0  45       10.0            JMP     TST02C
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                        SSUB
                                        ; ************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ; ************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ; ************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*    CLEAR PORT SELECT FOR TRANSPORT
                                        ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*    CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ; ************************************************************************
47A3  F5              12.0  CLEAR:  PUSH   PSW          ;SAVE THE SELECTED PORT #
47A4  C5              12.0          PUSH   B
47A5  06  00           7.0          MVI    B,0         ;START TO CLEAR AT PORT #0
47A7  DB  E0          10.0  CLRLP:  IN     INTSTA      ;GET MB SELECT INFO
47A9  E6  80           7.0          ANI    BIT7        ;SAVE ONLY THE MASSBUS SELECT BIT
47AB  B0               4.0          ORA    B           ;ADD IN THE SELECTED PORT #
47AC  D3  E0          10.0          OUT    MBSEL       ;RESET TO THIS PORT
47AE  3E  80           7.0          MVI    A,@200      ;LOAD MTA REGISTER #0 SELECT CODE
47B0  D3  40          10.0          OUT    TCMD        ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
47B2  AF               4.0          XRA    A           ;CLEAR TU COMMAND A
47B3  D3  40          10.0          OUT    TCMD        ;
47B5  3E  81           7.0          MVI    A,@201      ;LOAD MTA REGISTER #1 SELECT CODE
47B7  D3  40          10.0          OUT    TCMD        ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
47B9  3E  00           7.0          MVI    A,SELCLR    ;LOAD TU "CLEAR SELECT" COMMAND
47BB  D3  40          10.0          OUT    TCMD        ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
47BD  AF               4.0          XRA    A
47BE  D3  44          10.0          OUT    TAMT        ;CLEAR AMTIE WORD
47C0  D3  48          10.0          OUT    PDIAG       ;CLEAR DIAG CONTROL WORD
47C2  D3  4C          10.0          OUT    PENAB       ;CLEAR PORT ENABLE WORD
47C4  04               4.0          INR    B           ;POINT TO THE NEXT PORT TO CLEAR
47C5  78               4.0          MOV    A,B
47C6  FE  04           7.0          CPI    4           ;DONE?
47C8  C2  A7  47      10.0          JNZ    CLRLP       ;NO - CLEAR THIS PORT ALSO
47CB  C1              10.0          POP    B           ;RESET B & C
47CC  F1              10.0          POP    PSW         ;ALL DONE
47CD  C9              10.0          RET                ;EXIT
```

```
                                .SBTTL  SUBROUTINE 4X5 TRANSLATE A SUBGROUP
47CE                            SSUB
 (1)                            ; ***************************************************************************
 (1)                            ;*SUBROUTINE TITLE
 (1)                            ;*---------- -----
                                ;*4X5 TRANSLATE A SUBGROUP
47CE                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS SUBROUTINE IS USED TO 4X5 TRANSLATE A 4 BYTE DATA SUBGROUP INTO
                                ;*A 5 BYTE SUBGROUP AS RECORDED ON TAPE OR READ FROM TAPE, AT THE ENTRY
                                ;*T4X5.  AT ENTRY BADT4X5 THE VARIABLES BAD 1, 2, 3 CAN CONTAIN NUMBERS
                                ;*FROM 1-9 TO INDICATE TRACKS THAT WILL BE TRANSLATED TO A VALID VALUE,
                                ;*BUT NOT CORRECT FOR THE INPUT DATA SIMULATING BAD DATA.
47CE                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNSUB
                                ;*    CLEAR THE BAD TRANSLATION POINTERS
                                ;*    GET THE 4 (INPUT) DATA BYTES TO BE TRANSLATED
                                ;*    CLEAR THE TRANSLATED (OUTPUT) BUFFER (OUTPUT BUFFER IS 9X5)
                                ;*    SET LOOP COUNT TO 1
                                ;*    BGNDO
                                ;*    :    CALCULATE ODD PARITY FOR THE FOUR (INPUT) DATA BYTES (INPUT BUFFER IS NOW 9X4
                                ;*    :    ROTATE EACH (5) ENTRY OF THE TRANSLATED (OUTPUT) BUFFER RIGHT
                                ;*    :    GET THE LEAST SIGNIFICANT BIT OF EACH (INPUT) DATA BYTE AND
                                ;*    :        FORM A TABLE INDEX (MS BIT OF INDEX = LS BIT OF THE FIRST
                                ;*    :        BYTE OF THE INPUT DATA, ETC.)
                                ;*    :    REMOVE THE LS BIT OF EACH INPUT DATA BYTE (THE ONE JUST USED
                                ;*    :        TO BUILD THE INDEX)
                                ;*    :    USE THE 4 BIT INDEX TO GET THE TRANSLATION VALUE
                                ;*    :    PUT ONE BIT OF THE TRANSLATION VALUE, AS THE MOST SIGNIFICANT
                                ;*    :        BIT EACH OF THE 5 TRANSLATED DATA BYTES (LS BIT OF THE
                                ;*    :        TRANSLATION VALUE REPLACES MS BIT OF THE FIRST TRANSLATED
                                ;*    :        DATA BYTE.
                                ;*    :    INCREMENT THE LOOP COUNT
                                ;*    :    DO UNTIL THE LOOP COUNT = 10
                                ;*    ENDDO
                                ;*ENDSUB
47CE                            S
 (1)                            ; ***************************************************************************
                                
47CE F5            12.0 BADT4X5: PUSH   PSW              ;SAVE REGISTER A + PSW
47CF C3 DD 47      10.0          JMP    BAD
47D2 F5            12.0 T4X5:    PUSH   PSW              ;SAVE REGISTER A + PSW
47D3 AF             4.0          XRA    A                ;CLEAR A
47D4 32 B1 48      13.0          STA    BAD1             ;CLEAR BAD TRANSLATION 1
47D7 32 B2 48      13.0          STA    BAD2             ;CLEAR BAD TRANSLATION 2
47DA 32 B3 48      13.0          STA    BAD3             ;CLEAR BAD TRANSLATION 3
47DD C5            12.0 BAD:     PUSH   B                ;SAVE REGISTER B + C
47DE D5            12.0          PUSH   D                ;SAVE REGISTER D + E
47DF E5            12.0          PUSH   H                ;SAVE REGISTER H + L
```

```
2446                                        ;FOUR DATA BYTES ARE NOW IS THE BUFFER TABLE
2447                                        ;CLEAR THE TRANSLATED DATA TABLE.
2448  47E0  06  0A          7.0           MVI     B,10            ;SET UP LOOP COUNT
2449  47E2  11  A5  48      10.0          LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
2450  47E5  AF               4.0           XRA     A               ;CLEAR A
2451  47E6  12               7.0  D4X5:    STAX    D               ;STORE IN THE TABLE
2452  47E7  13               6.0           INX     D               ;UPDATE TABLE POINTER
2453  47E8  05               4.0           DCR     B               ;DECREMENT LOOP COUNT
2454  47E9  C2  E6  47       10.0          JNZ     D4X5            ;DO UNTIL LOOP COUNT=0
2455  47EC  0E  01           7.0           MVI     C,1             ;SET UP TRACK COUNT
2456  47EE  06  04           7.0  B4X5:    MVI     B,4             ;SET UP BIT COUNT
2457  47F0  11  A1  48       10.0          LXI     D,TRNIN
2458  47F3  AF               4.0           XRA     A               ;CLEAR THE GROUP POSITION COUNT
2459  47F4  32  AF  48       13.0          STA     GP4X5
2460  47F7  1A               7.0  C4X5:    LDAX    D               ;GET A DATA BYTE
2461  47F8  A7               4.0           ANA     A               ;SET CONDITION BITS
2462  47F9  E2  FD  47       10.0          JPO     PO4X5           ;ODD PARITY LEAVE CARRY CLEAR
2463  47FC  37               4.0           STC                     ;EVEN PARITY SET CARRY
2464  47FD  1F               4.0  PO4X5:   RAR                     ;SHIFT OUT DESIRED BIT
2465  47FE  12               7.0           STAX    D               ;STORE RESULT BACK IN TEMP TABLE
2466  47FF  3A  AF  48       13.0          LDA     GP4X5           ;GET THE GROUP POSITION BYTE
2467  4802  17               4.0           RAL                     ;PUT IN THIS DATA BIT
2468  4803  32  AF  48       13.0          STA     GP4X5           ;SAVE THE UPDATED GROUP POSITION BYTE
2469  4806  13               6.0           INX     D               ;UPDATE THE TABLE POINTER
2470  4807  05               4.0           DCR     B               ;DECREMENT THE BIT COUNT
2471  4808  C2  F7  47       10.0          JNZ     C4X5            ;DO UNTIL ALL 4 BITS ARE RECEIVED
2472
2473                                        ;AT THIS POINT GP4X5 CONTAINS 4 BITS - ONE FROM EACH OF THE FOUR BYTES
2474
2475                                        ;PERFORM A DOUBLE PRECISION SHIFT RIGHT ON THE TRANSLATED DATA
2476  480B  06  05           7.0           MVI     B,5             ;SET UP LOOP COUNT
2477  480D  11  A5  48       10.0          LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
2478  4810  13               6.0  E4X5:    INX     D               ;POINT TO PARITY BIT
2479  4811  1A               7.0           LDAX    D               ;GET PARITY BIT
2480  4812  A7               4.0           ANA     A               ;CLEAR THE CARRY BIT
2481  4813  1F               4.0           RAR                     ;SHIFT TO THE CARRY BIT
2482  4814  12               7.0           STAX    D               ;STORE IT BACK
2483  4815  1B               6.0           DCX     D               ;DECREMENT TO DATA BITS
2484  4816  1A               7.0           LDAX    D               ;GET THE DATA BITS
2485  4817  1F               4.0           RAR                     ;SHIFT IN CARRY BITS
2486  4818  12               7.0           STAX    D               ;STORE BACK IN TABLE
2487  4819  13               6.0           INX     D               ;POINT TO NEXT TABLE ENTRY
2488  481A  13               6.0           INX     D               ;
2489  481B  05               4.0           DCR     B               ;DECREMENT LOOP COUNT
2490  481C  C2  10  48       10.0          JNZ     E4X5            ;DO UNTIL LOOP COUNT=0
2491  481F  3A  B1  48       13.0          LDA     BAD1            ;IS A BAD POINTER SPECIFIED?
2492  4822  A7               4.0           ANA     A               ;SET CONDITION BITS
2493  4823  CA  2A  48       10.0          JZ      BAD1C           ;NO-CONTINUE
2494  4826  B9               4.0           CMP     C               ;YES-IS IT THIS DATA POSITION?
2495  4827  CA  40  48       10.0          JZ      INV4X5          ;YES-GO PERFORM INCORRECT TRANSLATION
2496  482A  3A  B2  48       13.0  BAD1C:   LDA     BAD2            ;IS A BAD POINTER SPECIFIED
2497  482D  A7               4.0           ANA     A               ;SET CONDITION BITS
2498  482E  CA  35  48       10.0          JZ      BAD2C           ;NO-CONTINUE
2499  4831  B9               4.0           CMP     C               ;YES-IS IT THIS DATA POSITION
```

```
2500  4832  CA 40 48   10.0          JZ      INV4X5          ;YES-GO PERFORM INCORRECT TRANSLATION
2501  4835  3A B3 48   13.0  BAD2C:  LDA     BAD3            ;IS A BAD POINTER SPECIFIED
2502  4838  A7          4.0          ANA     A               ;SET CONDITION BITS
2503  4839  CA 47 48   10.0          JZ      BAD3C           ;NO-PERFORM PROPER TRANSLATION
2504  483C  B9          4.0          CMP     C               ;YES-IS IT THIS DATA POSITION
2505  483D  C2 47 48   10.0          JNZ     BAD3C           ;NO-GO PERFORM PROPER TRANSLATION
2506  4840  3A AF 48   13.0  INV4X5: LDA     GP4X5           ;GET THE GROUP POSITION COUNT
2507  4843  3C          4.0          INR     A               ;INCREMENT-TO INCORRECT TRANSLATION
2508  4844  32 AF 48   13.0          STA     GP4X5           ;STORE IT
2509  4847  11 91 48   10.0  BAD3C:  LXI     D,TAB4X5        ;LOAD ADDRESS OF TRANSLATION TABLE
2510  484A  26 00        7.0          MVI     H,0             ;CLEAR REGISTER H
2511  484C  3A AF 48   13.0          LDA     GP4X5           ;GET GROUP POSITION COUNT
2512  484F  E6 0F        7.0          ANI     $0F
2513  4851  6F          4.0          MOV     L,A
2514  4852  19         10.0          DAD     D               ;ADD GROUP POSITION COUNT TO TABLE
2515  4853  7E          7.0          MOV     A,M             ;GET TRANSLATED DATA
2516  4854  32 B0 48   13.0          STA     TRNTMP
2517                              ;''TRNTMP'' = THE TRANSLATED DATA
2518
2519  4857  11 A5 48   10.0          LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
2520  485A  06 05        7.0          MVI     B,5             ;SET UP LOOP COUNT
2521  485C  3A B0 48   13.0  F4X5:   LDA     TRNTMP          ;GET TRANSLATED DATA
2522  485F  13           6.0          INX     D               ;POINT TO PARITY POSITION
2523  4860  A7          4.0          ANA     A               ;CLEAR CARRY BIT
2524  4861  1F          4.0          RAR                     ;SHIFT OUT A BIT
2525  4862  32 BC 48   13.0          STA     TRNTMP          ;STORE TRANSLATED DATA
2526  4865  1A          7.0          LDAX    D               ;GET THE PARITY BIT (ALWAYS ZERO)
2527  4866  17          4.0          RAL                     ;ROLL IN THE CARRY BIT
2528  4867  12          7.0          STAX    D               ;STORE AS THE PARITY BIT
2529  4868  13          6.0          INX     D               ;UPDATE TABLE POINTER
2530  4869  05          4.0          DCR     B               ;DECREMENT THE LOOP COUNT
2531  486A  C2 5C 48   10.0          JNZ     F4X5            ;DO UNTIL LOOP COUNT=0
2532
2533                          ;NOW THE TRANSLATED DATA TABLE HAS BEEN UPDATED
2534
2535  486D  0C          4.0          INR     C               ;DECREMENT THE TRACK COUNT
2536  486E  79          4.0          MOV     A,C
2537  486F  FE 0A        7.0          CPI     10
2538  4871  C2 EE 47   10.0          JNZ     B4X5            ;DO UNTIL ALL TRACKS TRANSLATED
2539  4874  E1         10.0          POP     H               ;RESTORE REGISTER H + L
2540  4875  D1         10.0          POP     D               ;RESTORE REGISTER D + E
2541  4876  C1         10.0          POP     B               ;RESTORE REGISTER B + C
2542  4877  F1         10.0          POP     PSW             ;RESTORE REGISTER A + PSW
2543  4878  C9         10.0          RET                     ;RETURN TO USER
2544
```

```
                                        .SBTTL  SUBROUTINE CALCULATE POINTER
                                        ;*THE POINTERS PROVIDED FOR THE TRACKS ARE ALSO DERIVED FROM THE INPUT/
                                        ;*EXPECTED DATA AS PER THE FOLLOWING TABLE.
                                        ;*
                                        ;*        BYTE COUNT                    POINTER GENERATED
                                        ;*      --------------                  -----------------
                                        ;*
                                        ;*        XXXXXX00                        HISTORY
                                        ;*        XXXXXX01                        PHASE TIE
                                        ;*        XXXXXX10                        AMTIE
                                        ;*        XXXXXX11                        ILLEGAL 5X4
                                        ;*
                                        ;*      X = DON'T CARE BITS
4879                                    SP
 (1)                                    ; ***************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;
4879  F5              12.0  POINTER: PUSH   PSW             ;SAVE THE ACCUMULATOR
487A  C5              12.0           PUSH   B               ;SAVE B + C
487B  3A  4F  49      13.0           LDA    TEMPX           ;GET BYTE COUNT
487E  E6  03           7.0           ANI    $3              ;REMOVE THE WORKING BITS
4880  21  8D  48      10.0           LXI    H,POINTBL       ;GET BASE ADDRESS OF TABLE
4883  4F               4.0           MOV    C,A
4884  06  00           7.0           MVI    B,0
4886  09              10.0           DAD    B
4887  7E               7.0           MOV    A,M             ;GET THE ECC POINTER
4888  D3  0A          10.0           OUT    RTIEB           ;SET THE TIE BUS VALUE
488A  C1              10.0           POP    B               ;RESTORE B + C
488B  F1              10.0           POP    PSW             ;RESTORE ACCUMULATOR
488C  C9              10.0           RET

488D  01                     POINTBL: .BYTE  1
488E  02                              .BYTE  2
488F  04                              .BYTE  4
4890  01                              .BYTE  1
```

```
2581                                    .SBTTL  TABLE 4 X 5 TRANSLATION
2582
2583                                    ;THIS TABLE TRANSLATED DATA FOR AS FOLLOWS
2584                                    ;   INPUT GROUP POSITIONS              OUTPUT GROUP POSITIONS
2585                                    ;    1 2 3 4 / 5 6 7 8                 10 9 8 7 6 / 5 4 3 2 1
2586                                    ;
2587                                    ;        0000                                  10011
2588                                    ;        0001                                  11011
2589                                    ;        0010                                  01001
2590                                    ;        0011                                  11001
2591                                    ;        0100                                  10111
2592                                    ;        0101                                  10101
2593                                    ;        0110                                  01101
2594                                    ;        0111                                  11101
2595                                    ;        1000                                  01011
2596                                    ;        1001                                  10010
2597                                    ;        1010                                  01010
2598                                    ;        1011                                  11010
2599                                    ;        1100                                  01111
2600                                    ;        1101                                  10110
2601                                    ;        1110                                  01110
2602                                    ;        1111                                  11110
2603
2604  4891  13                          TAB4X5: .BYTE   @23
2605  4892  1B                                  .BYTE   @33
2606  4893  09                                  .BYTE   @11
2607  4894  19                                  .BYTE   @31
2608  4895  17                                  .BYTE   @27
2609  4896  15                                  .BYTE   @25
2610  4897  0D                                  .BYTE   @15
2611  4898  1D                                  .BYTE   @35
2612  4899  0B                                  .BYTE   @13
2613  489A  12                                  .BYTE   @22
2614  489B  0A                                  .BYTE   @12
2615  489C  1A                                  .BYTE   @32
2616  489D  0F                                  .BYTE   @17
2617  489E  16                                  .BYTE   @26
2618  489F  0E                                  .BYTE   @16
2619  48A0  1E                                  .BYTE   @36
2620
```

```
2622                                    .SBTTL  SUBROUTINE VARIABLES
2623
2624  48A1 0004                         TRNIN:  .BLKB   4               ;TABLE CONTAINING THE 4 DATA BYTES TO
2625                                                                    ;BE TRANSLATED
2626  48A5 000A                         TRNOUT: .BLKB   10              ;TRANSLATED DATA TABLE DATA AFTER 4X5
2627                                                                    ;TRANSLATION
2628  48AF   00                         GP4X5:  .BYTE   0               ;ACCUMULATOR FOR THE 4 BIT SUBGROUP TO
2629                                                                    ;BE TRANSLATED
2630  48B0   00                         TRNTMP: .BYTE   0               ;TEMP. STORAGE FOR THE TRANSLATED SUBGROUP.
2631  48B1   00                         BAD1:   .BYTE   0               ;BAD TRANSLATION POINTER 1
2632  48B2   00                         BAD2:   .BYTE   0               ;BAD TRANSLATION POINTER 2
2633  48B3   00                         BAD3:   .BYTE   0               ;BAD TRANSLATION POINTER 3
2634
```

```
2636                                    .SBTTL  SUBROUTINE CLEAR ECC
2637  48B4                              S
 (1)                                    ; ************************************************************************
2638                                    ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION "ECCCHR" -
2639                                    ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
2640                                    ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
2641  48B4                              S
 (1)                                    ; ************************************************************************
2642
2643  48B4  AF                    4.0   CLECC:  XRA     A               ;CLEAR THE ACCUMULATOR
2644  48B5  32  E2  48           13.0          STA     ECCCHR          ;CLEAR THE ECC CHARACTER
2645  48B8  C9                    10.0          RET                     ;RETURN TO USER
2646
```

```
                                        .SBTTL  SUBROUTINE CALCUALTE ECC CHARACTER
48B9                                    S
 (1)                                    ; ************************************************************************
                                        ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
                                        ;THE CONTENTS OF "ECCCHR" TO UPDATE THE CONTENTS OF "ECCCHR" ACCORDING
                                        ;TO THE ANSI STANDARD ECC POLYNOMIAL.
48B9                                    S
 (1)                                    ; ************************************************************************

48B9  E5              12.0  ECC:    PUSH    H               ;SAVE H&L
48BA  D5              12.0          PUSH    D               ;SAVE D&E
48BB  F5              12.0          PUSH    PSW             ;SAVE THE ACCUMULATOR
48BC  21  E2  48      10.0          LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
48BF  AE               7.0          XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
48C0  5F               4.0          MOV     E,A             ;SAVE XOR RESULT IN E
48C1  E6  10           7.0          ANI     $10             ;IS BIT #4 OF RESULT SET
48C3  7B               4.0          MOV     A,E             ;RESTORE XOR RESULT FROM B
48C4  CA  C9  48      10.0          JZ      ECC1            ;CONTINUE IF BIT #4 RESET
48C7  EE  23           7.0          XRI     $23             ;ELSE-XOR WITH 23
48C9  5F               4.0  ECC1:   MOV     E,A             ;STORE THE ECC RESULT IN E
                                                            ;
48CA  AF               4.0          XRA     A               ;CLEAR A
48CB  4F               4.0          MOV     C,A             ;CLEAR THE TRANSLATE RESULT
48CC  21  DA  48      10.0          LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
48CF  CD  E3  48      18.0          CALL    TRANS           ;TRANSLATE THE BITS
48D2  79               4.0          MOV     A,C             ;GET THE TRANSLATED RESULT
48D3  32  E2  48      13.0          STA     ECCCHR          ;STORE RESULT
48D6  F1              10.0          POP     PSW             ;RESTORE THE ACCUMULATOR
48D7  D1              10.0          POP     D               ;RESTORE D&E
48D8  E1              10.0          POP     H               ;RESTORE H&L
48D9  C9              10.0          RET

48DA  08                    ECCTBL: $08                     ;BIT 0 = POSITION 3
48DB  20                            $20                     ;BIT 1 = POSITION 5
48DC  02                            $02                     ;BIT 2 = POSITION 1
48DD  40                            $40                     ;BIT 3 = POSITION 6
48DE  80                            $80                     ;BIT 4 = POSITION 7
48DF  01                            $01                     ;BIT 5 = POSITION 0
48E0  10                            $10                     ;BIT 6 = POSITION 4
48E1  04                            $04                     ;BIT 7 = POSITION 2

48E2  00                    ECCCHR: .BYTE   0
```

```
2690                                    .SBTTL  SUBROUTINE POLYNOMIAL BIT TRANSLATION
2691
2692                                    ;
2693                                    ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
2694                                    ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
2695                                    ;
2696
2697  48E3  06  01            7.0  TRANS:  MVI   B,1          ;INIT 'B' TO BIT POSITION 0
2698  48E5  7B                 4.0  TRANS1: MOV   A,E          ;GET CHAR TO BE TRANSLATED
2699  48E6  A0                 4.0          ANA   B            ;SEE IF BIT POSITION IN 'B' IS SET
2700  48E7  CA  ED  48        10.0          JZ    TRANS2       ;DO NEXT BIT POSITION IF NOT SET
2701  48EA  79                 4.0          MOV   A,C          ;GET PREVIOUS RESULT OF 'OR'
2702  48EB  B6                 7.0          ORA   M            ;'OR' IN NEW POSITION
2703  48EC  4F                 4.0          MOV   C,A          ;SAVE RESULT
2704
2705  48ED  78                 4.0  TRANS2: MOV   A,B
2706  48EE  07                 4.0          RLC                ;POSITION MASK TO NEXT BIT
2707  48EF  47                 4.0          MOV   B,A
2708  48F0  D8                12.0          RC                 ;EXIT WHEN ALL POSITIONS DONE
2709  48F1  23                 6.0          INX   H            ;POINT TO NEXT TABLE ENTRY
2710  48F2  C3  E5  48        10.0          JMP   TRANS1       ;PROCESS NEXT BIT
2711
```

```
                                        .SBTTL  SUBROUTINE TRK
48F5                                    SSUB
 (1)                                    ; ***************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
                                        ;*GENERATE BAD TRACK TRANSLATION
48F5                                    SD
 (1)                                    ; ***************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS SUBROUTINE GENRATES THREE TRACK NUMBERS FOR THE BADT4X5 SUBROUTINE
                                        ;*BASED ON THE VALUE OF INDAT AND THREE TIE BUS TIMES
                                        ;*INCORRECT TRANSLATIONS ARE MADE ON THE BASIS OF THE INPUT DATA TO THE
                                        ;*READ PATH FIFO.  THIS DATA APPEARS AS THE EXPECTED DATA IN AN ERROR
                                        ;*MESSAGE, AND BITS SIMULATED IN ERROR CAN BE DETERMINED FROM THE
                                        ;*FOLLOWING TABLE.
                                        ;*
                                        ;*      INPUT/EXPECTED          DATA BIT POSITIONS
                                        ;*          DATA                WITH ECC CORRECTION
                                        ;*      -------------           ------------------
                                        ;*
                                        ;*         XXXX0000                    0,1,2
                                        ;*         XXXX0001                    1,2,3
                                        ;*         XXXX0010                    2,3,4
                                        ;*         XXXX0011                    3,4,5
                                        ;*         XXXX0100                    4,5,6
                                        ;*         XXXX0101                    5,6,7
                                        ;*         XXXX0110                    6,7,P
                                        ;*         XXXX0111                    7,P,0
                                        ;*         XXXX1000                    P,0,1
                                        ;*         XXXX1001                    1,2,3
                                        ;*         XXXX1010                    2,3,4
                                        ;*         XXXX1011                    3,4,5
                                        ;*         XXXX1100                    4,5,6
                                        ;*         XXXX1101                    5,6,7
                                        ;*         XXXX1110                    6,7,P
                                        ;*         XXXX1111                    7,P,0
                                        ;*
                                        ;*      X = DON'T CARE BITS
48F5                                    S
 (1)                                    ; ***************************************************************************
                                        
48F5  3A  54  49      13.0  TRK:  LDA   INDAT           ;GET INPUT DATA
48F8  E6  0F           7.0        ANI   $0F            ;REMOVE 4 BITS
48FA  FE  08           7.0        CPI   $8             ;PARITY TRACK?
48FC  CA  01  49      10.0        JZ    TRKA           ;YES
48FF  E6  07           7.0        ANI   $7             ;NO-IGNORE BIT3
4901  3C               4.0  TRKA: INR   A              ;INCREMENT FOR SUBROUTINE
4902  32  B1  48      13.0        STA   BAD1           ;SAVE IT.
4905  3D               4.0        DCR   A              ;PREPARE TO USE AS TABLE INDEX
4906  21  46  49      10.0        LXI   H,TIMTBL       ;INCORRECT TIE BUS TIME TABLE START
4909  5F               4.0        MOV   E,A            ;LOW HALF OF TABLE INDEX
490A  16  00           7.0        MVI   D,0            ;ZERO HIGH HALF
490C  19              10.0        DAD   D              ;PRODUCE TABLE ENTRY ADDRESS
```

```
2760  490D  7E          7.0          MOV   A,M          ;GET INCORRECT TIE BUS TIME FROM TABLE
2761  490E  32  50  49   13.0          STA   TIET1        ;SAVE FIRST TIE BUS JAM TIME
2762  4911  3A  B1  48   13.0          LDA   BAD1
2763  4914  3C           4.0          INR   A            ;GENERATE VALUE FOR BAD2
2764  4915  FE  0A       7.0          CPI   $A           ;TOO BIG?
2765  4917  C2  1C  49   10.0          JNZ   TRKB         ;JUMP IF OK
2766  491A  3E  01       7.0          MVI   A,@1         ;YES
2767  491C  32  B2  48   13.0  TRKB:   STA   BAD2         ;SAVE IT
2768  491F  3D           4.0          DCR   A            ;PREPARETO USE AS TABLE INDEX
2769  4920  21  46  49   10.0          LXI   H,TIMTBL     ;TIE BUS TIME TABLE START
2770  4923  5F           4.0          MOV   E,A          ;LOW HALF OF TABLE INDEX
2771  4924  16  00       7.0          MVI   D,0          ;CLEAR UPPER HALF
2772  4926  19          10.0          DAD   D            ;PRODUCE TABLE ENTRY ADDRESS
2773  4927  7E           7.0          MOV   A,M          ;GET SECOND TIE BUS JAM TIME
2774  4928  32  51  49   13.0          STA   TIET2        ;SAVE IT
2775  492B  3A  B2  48   13.0          LDA   BAD2
2776  492E  3C           4.0          INR   A
2777  492F  FE  0A       7.0          CPI   $A
2778  4931  C2  36  49   10.0          JNZ   TRKC
2779  4934  3E  01       7.0          MVI   A,@1
2780  4936  32  B3  48   13.0  TRKC:   STA   BAD3
2781  4939  3D           4.0          DCR   A
2782  493A  21  46  49   10.0          LXI   H,TIMTBL
2783  493D  5F           4.0          MOV   E,A
2784  493E  16  00       7.0          MVI   D,0
2785  4940  19          10.0          DAD   D
2786  4941  7E           7.0          MOV   A,M
2787  4942  32  52  49   13.0          STA   TIET3
2788  4945  C9          10.0          RET
2789  4946  02                TIMTBL: .BYTE 2            ; TIE BUS TIME FOR BIT 0
2790  4947  03                        .BYTE 3            ; TIE BUS TIME FOR BIT 1
2791  4948  00                        .BYTE 0            ; TIE BUS TIME FOR BIT 2
2792  4949  01                        .BYTE 1            ; TIE BUS TIME FOR BIT 3
2793  494A  04                        .BYTE 4            ; TIE BUS TIME FOR BIT 4
2794  494B  08                        .BYTE 8            ; TIE BUS TIME FOR BIT 5
2795  494C  07                        .BYTE 7            ; TIE BUS TIME FOR BIT 6
2796  494D  05                        .BYTE 5            ; TIE BUS TIME FOR BIT 7
2797  494E  06                        .BYTE 6            ; TIE BUS TIME FOR BIT P
2798
2799
```

```
2801                                    .SBTTL  PROGRAM VARIABLES
2802
2803  494F   00                         TEMPX:  .BYTE   0               ;TEMP STORAGE
2804  4950   00                         TIET1:  .BYTE   0               ;FIRST TIE JAM TIME
2805  4951   00                         TIET2:  .BYTE   0               ;SECOND TIE BUS JAM TIME
2806  4952   00                         TIET3:  .BYTE   0               ;THIRD TIE BUS JAM TIME
2807  4953   00                         SUNIT:  .BYTE   0               ;TU PORT #
2808  4954   00                         INDAT:  .BYTE   0               ;DATA BYTE TO ECC
2809  4955   00                         INDATP: .BYTE   0               ;DATA BYTE PARITY
2810  4956   00                         UNITMP: .BYTE   0               ;UNIT MAP
2811  4957   00   00                    AMTMSK: .WORD   0               ;AMTIE MASK WORD
2812  4959   00                         DATAA:  .BYTE   0               ;ACTUAL DATA AFTER TRANSLATION
2813  495A   00                         DATAAP: .BYTE   0               ;ACTUAL DATA PARITY AFTER TRANSLATION
2814  495B   00                         DATAE:  .BYTE   0               ;EXPECTED DATA AFTER TRANSLATION
2815  495C   00                         DATAEP: .BYTE   0               ;EXPECTED DATA PARITY AFTER TRANSLATION
2816  495D 0007                         BUFFER: .BLKB   7               ;DATA BYTE STORAGE
2817       0000                                 .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7- 0002
AMTMSK  4957           ARAIDF= 0098          ASAVE   4F9B          ATTCD   4F97
AXNUM   4F91           B      =%0000          BAD     47DD          BADST  = 0090
BADT4X  47CF           BAD1    48B1          BAD1C   482A          BAD2    48B2
BAD2C   4835           BAD3    48B3          BAD3C   4847          BIT0   = 0001
BIT1   = 0002          BIT15  = 8000          BIT2   = 0004         BIT3   = 0008
BIT4   = 0010          BIT5   = 0020          BIT6   = 0040         BIT7   = 0080
BIT8   = 0100          BIT9   = 0200          BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60           BRKXCT= 4F00          BSAVE   4F9C          BUFFER  495D
BYTCNT= 00D4           BYTEH   4F24          BYTEL   4F23          B4X5    47EE
C      =%0001          CASCT   4F21          CASCTL= 00A0          CASSTA= 00A0
CATTH  = 0089          CATTL  = 0088         CBUSST= 00A1          CBYTH  = 008B
CBYTL  = 008A          CDG1H  = 0087         CDG1L  = 0086         CDG2H  = 0093
CDG2L  = 0092          CDG3H  = 0095         CDG3L  = 0094         CDVTH  = 008D
CDVTL  = 008C          CHPTIE= 0028          CH0TIE= 0020          CH1TIE= 0021
CH2TIE= 0022           CH3TIE= 0023          CH4TIE= 0024          CH5TIE= 0025
CH6TIE= 0026           CH7TIE= 0027          CKLOP  = 2815         CLEAR   47A3
CLECC   48B4           CLKCTL= 00F0          CLOCK   4F26          CLRLP   47A7
CMC0H  = 0099          CMC0L  = 0098         CMC1H  = 009B         CMC1L  = 009A
CMC2H  = 009D          CMC2L  = 009C         CMC3H  = 009F         CMC3L  = 009E
CMINH  = 0097          CMINL  = 0096         CNTCTL= 00D7          CRCWRD= 0018
CSAVE   4F9D           CSRLH  = 0091         CSRLL  = 0090         CTCH   = 0085
CTCL   = 0084          CTSTH  = 008F         CTSTL  = 008E         CXCTH  = 0081
CXCTL  = 0080          CXINH  = 0083         CXINL  = 0082         C.     = 0001
C.AVAI= 0080           C.DP   = 0008         C.DSE  = 0010         C.DTU  = 0003
C.DVA  = 0008          C.FAIL= 00FC          C.FMT  = 0070         C.FNCT= 003E
C.GO   = 0001          C.INTC= 00FE          C.MAIN= 0020          C.NSA  = 0080
C.RCT  = 00FC          C.SER  = 0080         C.SHR  = 0040         C.SKPC= 000F
C.TAPE= 0040           C.WCS  = 0002         C4X5    47F7          D      =%0002
DATAA   4959           DATAAP  495A          DATACT= 00D0          DATAE   495B
DATAEP  495C           DBUS    4F28          DBUSCT= 00C0          DBUSST= 00C0
DDRA   = 00D8          DDRAIN= 0010          DDRB   = 00D9         DDRBIN= 0002
DDRC   = 00DA          DDRCIN= 0001          DDRCO  = 0088         DDRCTL= 00DB
DIAFLG  4F22           DIAGPG= 4300          DIAGRM= 4F90          DIARD  = 000B
DONE1  = 0045          DONINT= 0010          DSAVE   4F9E          DSE    = 0006
DUMMU   467A  G        DUMMV   46B3  G       DUMMW   4454  G       DUMMX   44AA  G
DUMMY   431E  G        D.ATH0= 0001          D.ATH1= 0002          D.EOTD= 0010
D.LAGC= 0020           D.NOTW= 0040          D.NTHR= 0004          D.TACH= 0008
D.WR4  = 0080          D4X5    47E6          E      =%0003         ECC     48B9
ECCBAD= 0042           ECCCHR  48E2          ECCCOR= 0019          ECCOK  = 0041
ECCSTA= 001A           ECCTBL  48DA          ECCTST= 000E          ECC1    48C9
EDATA  = 0095          EOTCLR= 0003          ERFLG   4F93          ERLP   = 2809
ERLPA  = 280F          ERLPB  = 2812         ERLPE  = 280C         ERNUM   4F90
ERRCNT= 00D6           ESAVE   4F9F          EXIT    4795          E.ACRC= 0010
E.AMT  = 0020          E.CDP  = 0080         E.CRC  = 0080         E.PNTR= 0008
E.RPE  = 0040          E.STEC= 0001          E.TTEC= 0002          E.UNC  = 0004
E4X5    4810           FIFORD= 006A          FORMAT  4F25          FOUND   4344
FWDTST= 0061           F4X5    485C          GCRID  = 0089         GCRSET= 0002
GOODTM= 0092           GP4X5   48AF          H      =%0004         HLSAVE  4FA0
IE     = 0008          INDAT   4954          INDATP  4955          INTSTA= 00E0
INV4X5  4840           ITERA   4F9A          I.PWR  = 0020         I.RMPE= 0040
I5.5   = 0010          I6.5   = 0020         I7.5   = 0040         KCALL  = 005F
KCLR   = 007B          KDEP   = 003F         KENAB  = 0078         KEXAM  = 003E
KEYBRD= 00C8           KEY1   = 0078         KEY10  = 006D         KEY11  = 006E
```

```
KEY12 = 006F    KEY13 = 005C    KEY14 = 005D    KEY15 = 005E
KEY16 = 005F    KEY17 = 003C    KEY18 = 003D    KEY19 = 003E
KEY2  = 0079    KEY20 = 003F    KEY3  = 007A    KEY4  = 007B
KEY5  = 0074    KEY6  = 0075    KEY7  = 0076    KEY8  = 0077
KEY9  = 006C    KINTA = 006F    KLDAD = 003D    KN0   = 003C
KN1   = 005C    KN2   = 005D    KN3   = 005E    KN4   = 006C
KN5   = 006D    KN6   = 006E    KN7   = 0074    KN8   = 0075
KN9   = 0076    KU2   = 0079    KU3   = 007A    KU8   = 0077
L     =%0005    LBLANK= 000F    LCE   = 000E    LCH   = 000C
LCL   = 000D    LCP   = 000E    LC0   = 0000    LC1   = 0001
LC2   = 0002    LC3   = 0003    LC4   = 0004    LC5   = 0005
LC6   = 0006    LC7   = 0007    LC8   = 0008    LC9   = 0009
LDLEDA= 00CA    LDLEDB= 00CB    LDLEDC= 00CC    LDLEDD= 00CD
LDLEDE= 00CE    LDLEDF= 00CF    LKDIAG= 2800    LKKBD = 004C
LKKEY = 0049    LKLWMG= 0058    LKLWMP= 0055    LKLWPG= 0052
LKLWPP= 004F    LKMOD7= 0046    LKOPR = 0046    LPFLG   4F94
LPNUM   4F92    M     =%0006    MBSEL = 00E0    MB.A  = 0008
MB.B  = 0004    MEMTOP= 4FFF    MINUS = 000A    MM    = 8000
MSE   = 0008    MSGN  = 000F    MTACLR= 0000    MT.ARA= 0020
MT.CPE= 0080    MT.DSE= 0001    MT.FWD= 0040    MT.INH= 0008
MT.LWR= 0004    MT.MOT= 0002    MT.NWT= 0080    MT.PEC= 0040
MT.PSB= 0004    MT.PS0= 0001    MT.PS1= 0002    MT.REV= 0020
MT.WRT= 0010    MT.Z  = 0008    M.ATA = 0080    M.CAPE= 0020
M.CONT= 0080    M.DEM = 0020    M.EBL = 0010    M.EXC = 0008
M.FAIL= 0008    M.ILR = 0010    M.INIT= 0010    M.OCC = 0020
M.ONLI= 0001    M.PE  = 0040    M.PORT= 0080    M.RDEN= 0002
M.RDPE= 0008    M.RUN = 0004    M.SCLK= 0001    M.TRA = 0040
M.UNIT= 0007    M.WCLK= 0040    M.WCLN= 0080    M.WREN= 0080
M5.5  = 0001    M6.5  = 0002    M7.5  = 0004    NOTCAP= 0088
OKAY  = 00FF    OPRRAM= 4300    OPSTRT= 0058    OPVER = 0040
PADCNT= 00D5    PADCRC= 0080    PDIAG = 0048    PEID  = 008A
PENAB = 004C    PESET = 0001    PL    = 00B1    POINTB  488D
POINIE  4879    PO4X5   47FD    PRDD  = 004C    PRENF = 009C
PS    = 00B2    PSTAT = 0048    PSW   =%0009    P.AMTP= 0001
P.BCTC= 0040    P.CMDP= 0020    P.INTE= 0080    P.LCS = 0040
P.LWR = 0020    P.RDP = 0002    P.RPEN= 0004    P.RPST= 0002
P.RPOE= 0020    P.RP1E= 0010    P.RP2E= 0020    P.RP3E= 0010
P.SING= 0080    P.STAT= 0002    P.STPE= 0080    P.TACH= 0008
P.TUPR= 0010    P.WCSP= 0004    P.WDS = 0040    P.WFLP= 0001
P.WPEN= 0010    P.WPOE= 0008    P.WP1E= 0004    P.WP2E= 0008
P.WP3E= 0004    P.5VOK= 0002    QUE   = 281B    QUEM  = 281E
RAMT  = 0010    RARA  = 0006    RARAI = 0004    RCHBD0= 0048
RCHBD1= 0047    RCHOK = 0046    RCHTST= 000C    RCLRT = 000D
RCMD  = 000B    RCMLP = 0003    RCONT = 0080    RDATA = 0017
RDCLK = 0010    RDON  = 0011    READG = 0007    REND  = 0014
REQST = 2806    RESCHR= 00D1    RETRY   4798    REVTST= 0064
REWIND= 0004    RFIFOL= 0008    RGCLK = 0002    RGCRI = 0003
RIBG  = 0001    RILL  = 0012    RINST = 00CC    RMCTST= 0008
RMK2  = 0013    RNOP  = 0000    RPATH = 0001    RPBAD = 0044
RPCHI = 0001    RPCLK = 0003    RPCTL = 0009    RPEI  = 0002
RPFAIL= 0000    RPF1  = 009D    RPF2  = 009E    RPOK  = 0043
RPOSTN= 0016    RPSTA = 0015    RRCMT = 000A    RSTAT = 0002
RTIEB = 000A    RTIER = 0030    RTM   = 0005    RUNKI = 0009
RUPTST= 005E    RWDUNL= 0005    R.AMT = 0001    R.BOP = 0008
```

```
R.DATA= 0040        R.DON = 0002        R.DRDY= 0010        R.END = 0010
R.ILL = 0004        R.JVOK= 0004        R.MK2 = 0008        R.PLOD= 0008
R.PLO0= 0010        R.PLO1= 0020        R.POST= 0020        R.STNM= 0002
R.STOP= 0004        R.STPC= 0001        R.TBJN= 0080        R.TSTD= 0040
R.VOK = 0080        R00H  = 0081        R00L  = 0080        R01H  = 0083
R01L  = 0082        R02H  = 0085        R02L  = 0084        R03H  = 0087
R03L  = 0086        R04H  = 0089        R04L  = 0088        R05H  = 008B
R05L  = 008A        R06H  = 008D        R06L  = 008C        R07H  = 008F
R07L  = 008E        R10H  = 0091        R10L  = 0090        R11H  = 0093
R11L  = 0092        R12H  = 0095        R12L  = 0094        R13H  = 0097
R13L  = 0096        R14H  = 0099        R14L  = 0098        R15H  = 009B
R15L  = 009A        R16H  = 009D        R16L  = 009C        R17H  = 009F
R17L  = 009E        R7.5  = 0010        SELCLR= 0000        SETATA= 00A1
SID   = 0080        SOD   = 0080        SOE   = 0040        SP    =%0008
SSCLK = 0040        SSTEP = 0005        STACK = 4FFF        STATRM= 4F20
STPCT   4F20        STRSP = 5000        SUNIT   4953        TAB4X5  4891
TADR00= 0080        TADR01= 0081        TADR02= 0082        TADR03= 0083
TADR04= 0084        TADR05= 0085        TADR06= 0086        TADR07= 0087
TADR10= 0088        TADR11= 0089        TADR12= 008A        TADR13= 008B
TAMT  = 0044        TASEL = 0080        TCMD  = 0040        TC.FWD= 0040
TC.INH= 0008        TC.LWR= 0004        TC.REV= 0020        TC.WRT= 0010
TEMP    4F99        TEMPX   494F        TEST1   4300        TEST2   4573
TIET1   4950        TIET2   4951        TIET3   4952        TIMTBL  4946
TMF   = 0099        TMRDY = 0040        TRANS   48E3        TRANS1  48E5
TRANS2  48ED        TRK     48F5        TRKA    4901        TRKB    491C
TRKC    4936        TRKENA= 00D2        TRNIN   48A1        TRNOUT  48A5
TRNTMP  48B0        TSET  = 2803        TSTEND= 2818        TSTS  = 0040
TST01A  445D        TST01B  4461        TST01C  437F        TST01F  439A
TST01G  43EB        TST01H  4423        TST01I  43AD        TST01K  44B3
TST01L  44C7        TST01M  44DA  G     TST01N  451C        TST01Q  4501
TST01R  44F8  G     TST01S  4511        TST01T  450B  G     TST01U  4533  G
TST01V  454F  G     TST01W  43B9        TST01X  4351        TST01Y  44B8
TST01Z  43C0        TST02A  4683        TST02B  4687        TST02C  45B0
TST02F  45CB        TST02G  4618        TST02H  4649        TST02I  45DE
TST02K  46BC        TST02L  46D3        TST02M  46EA  G     TST02N  4734
TST02Q  4715        TST02R  470C  G     TST02S  4729        TST02T  4723  G
TST02U  4751  G     TST02V  4773  G     TST02W  45EA        TST02X  457D
TST02Y  46C1        TST02Z  45F1        TUSEL0= 00D1        TUSEL1= 00D2
TU78  = 0010        T.ATH0= 0001        T.ATH1= 0002        T.BOT = 0004
T.EOT = 0002        T.FPT = 0001        T.NTHR= 0004        T.ONL = 0020
T.PES = 0008        T.PSBJ= 0020        T.PSOJ= 0008        T.PS1J= 0010
T.RDY = 0080        T.RDYO= 0040        T.RWD = 0010        T.SCLK= 0002
T4X5    47D2        UIBG  = 00A1        UNITMP  4956        UNITSL  4327
VALFC   4F98        VALTB   4F95        VELTST= 005B        WDR.P = 0010
WMCCTL= 00D3        WMCERR= 00DA        WMCSTA= 00D0        WRTCLK= 0000
WRTDAT= 00D3        W.ACRC= 0004        W.CRC = 0008        W.DIAG= 0002
W.DONN= 0040        W.ECC = 0010        W.ENAB= 0080        W.ERR = 0020
W.FMT = 0070        W.GCR = 0010        W.LEFT= 0004        W.ONES= 0020
W.RESI= 0002        W.REV = 0004        W.ROME= 0010        W.RST = 0001
W.SKIP= 000F        W.WRIT= 0008        W.XFER= 0020        X     =%000A
X.DONN= 0080        X.ENAB= 0040        X.PEPE= 0002        X.ROME= 0001
X.WCLK= 0001        Y     =%000B        .     = 4964
```

ERRORS DETECTED: 0

*ECC5.A78/PTP,ECC5=NLIST,PARAM,MACRO,LIST,ECC5
RUN-TIME: 4 7 0 SECONDS
CORE USED: 10K

     3   "MEMORY MAP" EXTENDED ADDRESS BIT
    19   READ MICRO CONTROLLER CONTROL & STATUS WORDS
   139   TRANSPORT/PORT COMMAND + STATUS WORDS
   380   COMMON ADDRESS SPACE REGISTERS
   444   MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
   485   MASSBUS CONTROL WORD DEFINITIONS
   557   WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
   709   KEYBOARD/DISPLAY CONTROL + STATUS WORD
   797   MISCELLANEOUS VARIABLE DEFINITIONS
   801   RAM ADDRESS DEFINITIONS
   819   LINKAGE TABLE ADDRESSES
   842   BIT DEFINITIONS
   856   RIM AND SIM INSTRUCTION BIT DEFINITIONS
   877   INTERRUPTABLE INSTRUCTION MACROS
   924   DIAGNOSTIC VARIABLES
   945   COMMON VARIABLE STORAGE
   959   MACRO CALL DEFINITIONS
  1047   TM78 MICRO-DIAGNOSTIC MACRO CALLS
  1101   8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
  1334   TEST 1 - CHECK CHARACTER MICRO INITIAL STATUS TEST
  1406   TEST 2 - CRC AND ACRC TEST
  1784   SUBROUTINE SETDDR
  1827   SUBROUTINE CLEAR ALL TU PORTS
  1872   SUBROUTINE   CLKSYS
  1885   PROGRAM VARIABLES

```
961                     ;*************************************************************************
962                     ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
963                     ;ERROR MACRO CALLS
964                     ;
965                     ;       1  -    REQUEST HOST CPU TO PRINT:
966                     ;               "BYTE/SCLK COUNT NUMBER = LLL"
967                     ;               WHERE:
968                     ;               LLL = THE VALUE STORED IN CAS REGISTER 5
969                     ;                     (THE BYTE COUNT REGISTER 16 BITS).
970                     ;
971                     ;       2  -    REQUEST HOST CPU TO PRINT:
972                     ;               DATA FORMAT = MM
973                     ;               SKIP COUNT  = NN
974                     ;               WHERE:
975                     ;               MM = DATA FORMAT FROM CAS REGISTER 2
976                     ;               NN = SKIP COUNT FROM CAS REGISTER 2
977                     ;
978                     ;       3  -    REQUEST HOST CPU TO PRINT:
979                     ;               BYTE-SCLK COUNT = LLL
980                     ;               DATA FORMAT = MM
981                     ;               SKIP COUNT  = NN
982                     ;               WHERE:
983                     ;               LLL = AS ABOVE
984                     ;               MM  = AS ABOVE
985                     ;               NN  = AS ABOVE
986                     ;
987                     ;       4  -    REQUEST HOST TO PRINT:
988                     ;               TRANSITION COUNT = LLL
989                     ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
990                     ;
991                     ;       5  -    REQUEST HOST CPU TO PRINT:
992                     ;               EXPECTED 18 BITS =      E EEEEEE
993                     ;               ACTUAL 18 BITS =        A AAAAAA
994                     ;
995                     ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
996                     ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
997                     ;               OF CAS REG 15 LOW BYTE.
998                     ;
999                     ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
1000                    ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
1001                    ;               17 LOW BYTE SIGN BIT.
1002                    ;
1003                    ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
1004                    ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
1005                    ;               AND/OR ACTUAL DATA.
1006                    ;
1007                    ;       7  -    REQUEST HOST CPU TO PRINT:
1008                    ;               "SUBGROUP NUMBER = LLL"
1009                    ;               WHERE:
1010                    ;               LLL = THE VALUE STORED IN CAS REGISTER 5
1011                    ;                     (THE BYTE COUNT REGISTER 16 BITS).
1012                    ;
1013                    ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
1014                    ;*************************************************************************
```

```
                        ;*************************************************************************
                        ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
                        ;TO CAUSE SOME HOST CPU ACTION.

                        ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
                        ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
                        ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
                        ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
                        ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
                        ;               - HOST RESPONSE CODE 31 OR 33
                        ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
                        ;               - HOST RESPONSE CODE 31 OR 33
                        ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
                        ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
                        ;                       BIT0 = 1 TEST PORT 0
                        ;                       BIT1 = 1 TEST PORT 1
                        ;                       BIT2 = 1 TEST PORT 2
                        ;                       BIT3 = 1 TEST PORT 3
                        ;*************************************************************************
                        ;*************************************************************************
                        ;DATA PATTERN CODES

                        ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
                        ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
                        ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
                        ;               18 BITS OF ALL 1'S
                        ;               18 BITS OF ALL 0'S
                        ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
                        ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
                        ;*************************************************************************
  4300                          .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  CCM1 - CHECK CHARACTER MICRO CONTROLLER PART #1
                                ;ID     CCM1-CHECK CHARACTER MICRO CONTROLLER
                                .SBTTL  TEST 1 - CHECK CHARACTER MICRO INITIAL STATUS TEST
4300                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*CCM - INITIAL STATUS CHECK
4300                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST CLEARS THE CHECK CHARACTER CONTROLLER AND VERIFIES THAT THE
                                ;*ACRC NOT OK AND CRC NOT OK ARE BOTH SET.
4300                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  SET NORMAL READ PATH CLOCK
                                ;*  PUT THE MACHINE IN GCR WRITE MODE
                                ;*  CALL SUBROUTINE CLEAR
                                ;*  ENABLE THE READ PATH CLOCK
                                ;*  SET PLO BYPASS MODE 2
                                ;*  ISSUE CLEAR ALL COMMAND
                                ;*  WAIT
                                ;*  STOP THE READ PATH
                                ;*  INPUT THE ACRC AND CRC NOT OK STATUS BITS
                                ;*  IF ACRC NOT OK=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*  IF CRC NOT OK=1
                                ;*  :  THEN-CONTINUE
                                ;*  :  ELSE-ERROR
                                ;*  ENDIF
                                ;*ENDTST
4300                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*CCM1 MICRO TEST 01
                                ;*CCM1 MICRO ERROR 01
                                ;*CCM1-CHECK CHARACTER CONTROLLER-INITIAL STATUS TEST
                                ;*M8952, M8953
                                ;*ACRC NOT OK = 0 AFTER CLEAR SHOULD = 1
                                ;*
                                ;*CCM1 MICRO TEST 01
                                ;*CCM1 MICRO ERROR 02
                                ;*CCM1-CHECK CHARACTER CONTROLLER-INITIAL STATUS TEST
                                ;*M8952, M8953
                                ;*CRC NOT OK = 0 AFTER CLEAR SHOULD = 1
4300                            S
 (1)                            ; ************************************************************************
```

```
1374  4300                                  TEST1:  TESTX   @1               ;INITIALIZE THE TEST
 (1)  4300  3E  01              7.0                    MVI     A,@1             ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28         18.0                    CALL    TSET             ;SETUP THE TEST
1375                                  ;%CCM1-CHECK CHARACTER CONTROLLER-INITIAL STATUS TEST
1376                                  ;&M8952, M8953, M8951
1377
1378  4305  3E  10              7.0  TST01X: MVI     A,RDCLK          ;SET NORMAL READ PATH CLOCK
1379  4307  D3  F0             10.0           OUT     CLKCTL
1380  4309  3E  18              7.0           MVI     A,W.GCR!W.WRITE ;PUT THE MACHINE IN GCR
1381  430B  D3  D3             10.0           OUT     WMCCTL           ;WRITE MODE
1382  430D  CD  8E  45         18.0           CALL    CLEAR            ;CALL SUBROUTINE CLEAR
1383  4310  3E  20              7.0           MVI     A,R.PLO1         ;SET PLO BYPASS MODE 2 AND
1384  4312  D3  09             10.0           OUT     RPCTL            ;ENABLE THE READ PATH CLOCK
1385  4314  3E  0D              7.0           MVI     A,RCLRT          ;ISSUE CLEAR ALL COMMAND
1386  4316  D3  0B             10.0           OUT     RCMD
1387  4318  00                  4.0           NOP                      ;
1388  4319  00                  4.0           NOP
1389  431A  00                  4.0           NOP
1390  431B  00                  4.0           NOP
1391  431C  00                  4.0           NOP
1392  431D  3E  21              7.0           MVI     A,R.PLO1!R.STPC ;STOP THE READ PATH
1393  431F  DB  1A             10.0           IN      ECCSTA           ;GET THE CCM STATUS
1394  4321  E6  10              7.0           ANI     E.ACRC           ;ACRC NOT OK=1?
1395  4323  C2  2B  43         10.0           JNZ     T01CN1           ;YES-CONTINUE
1396  4326                                    ERR     TST01X,T01CN1
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4326  CD  09  28         18.0                    CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)       0001                                       MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4329  01                                         .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  432A  00                                         .BYTE
 (1)  432B  CD  15  28         18.0           T01CN1::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  432E  DA  05  43         10.0                    JC      TST01X           ;LOOP ADDRESS IF LOOP SPECIFIED
1397                                  ;>ACRC NOT OK = 0 AFTER CLEAR SHOULD = 1
1398  4331  DB  1A             10.0           IN      ECCSTA
1399  4333  E6  80              7.0           ANI     E.CRC
1400  4335  C2  3D  43         10.0           JNZ     T01CN2
1401  4338                                    ERR     TST01X,T01CN2
 (1)                                        ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4338  CD  09  28         18.0                    CALL    ERLP             ;PROCESS ERROR - DO 2.3
 (1)       0002                                       MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  433B  02                                         .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  433C  00                                         .BYTE
 (1)  433D  CD  15  28         18.0           T01CN2::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4340  DA  05  43         10.0                    JC      TST01X           ;LOOP ADDRESS IF LOOP SPECIFIED
1402                                  ;>CRC NOT OK = 0 AFTER CLEAR SHOULD - 1
```

```
1404  4343                                      ENDTST  TST01X
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4343                                      REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4343  CD  06  28        18.0                            CALL    REQST
 (2)  4346  00                                                .BYTE                   ;DATA PATTERN NUMBER
 (2)  4347  00  00                                            .WORD                   ;SYSTEM "" COUNT
 (2)  4349  00  00                                            .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  434B  00                                                .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  434C  07                                                .BYTE   7               ;REQUEST CODE
 (1)  434D  3A  9A  4F        13.0              LDA     ITERA           ;GET ITERATION COUNT
 (1)  4350  3D                 4.0              DCR     A               ;DOWNCOUNT
 (1)  4351  32  9A  4F        13.0              STA     ITERA           ;SAVE COUNT
 (1)  4354  F2  05  43        10.0              JP      TST01X          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 2 - CRC AND ACRC TEST
4357                            ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*CCM CRC AND ACRC TEST
4357                            SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WRITES RECORDS FROM 1 - 128 BYTES AND CHECKS THAT THE CHECK
                                ;*CHARACTER MICRO-CONTROLLER WILL ACCURATELY RECOMPUTE THE ACRC AND CRC
                                ;*CHARACTERS.
4357                            SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT
                                ;*  CLEAR THE BYTE COUNT CONTROL BYTE (SOFTWARE)
                                ;*  BGNDO
                                ;*  :  CALL SUBROUTINE CLEAR
                                ;*  :  CLEAR THE TU PORT COMMAND REGISTER
                                ;*  :  ENABLE THE WRITE AND READ PATHS TO THE SELECTED TU PORT
                                ;*  :  CLEAR THE XMC TRACK ENABLE LINES
                                ;*  :  CLEAR THE TU PORT AMTIE LINES
                                ;*  :  SET NORMAL READ PATH CLOCKS
                                ;*  :  PUT THE MACHINE IN GCR MODE
                                ;*  :  SET THE TU PORT IN LOOP WRITE/READ MODE
                                ;*  :  SET THE READ PATH CONTROL REGISTER TO BYPASS 2, TIE JAM, AND PLO DISABLE
                                ;*  :  CLEAR THE READ CHANNEL TIE BUS
                                ;*  :  CLOCK THE READ CHANNEL FIFO'S
                                ;*  :  ISSUE READ PATH CLEAR ALL COMMAND
                                ;*  :  WAIT
                                ;*  :  SET THE READ PATH CONTROL REGISTER TO BYPASS 2, TIE JAM, PLO DISABLE AND STOP
                                ;*  :  INITIALIZE THE DDR CONTROL
                                ;*  :  SET NORMAL WRITE PATH CLOCKS
                                ;*  :  SET THE WMCCTL REGISTER TO RESTART
                                ;*  :  WAIT
                                ;*  :  CLEAR THE WMCCTL REGISTER
                                ;*  :  WAIT 1280 MICROSECONDS
                                ;*  :  SETUP SINGLE STEP WRITE CLOCK NORMAL READ CLOCK
                                ;*  :  SELECT THE BYTE, PAD AND ECODE COUNTERS
                                ;*  :  CLEAR THE BYTE, PAD AND ECODE COUNTERS
                                ;*  :  INCREMENT THE BYTE COUNT CONTROL BYTE
                                ;*  :  INITIALIZE THE DDR FOR A DATA TRANSFER
                                ;*  :  CALL OPERATIONAL MICROCODE MOD7 ROUTINE
                                ;*  :  CALCULATE THE MOD32 BITS
                                ;*  :  CALCULATE THE RESIDUAL CHARACTER COUNT
                                ;*  :  CALCULATE THE PAD CHARACTER COUNT
                                ;*  :  CALCULATE THE PAD CRC BIT
                                ;*  :  LOAD THE IMAGE MODE OF DATA TRANSFER
                                ;*  :  ENABLE ALL XMC TRACK ENABLE BITS
                                ;*  :  SET ALL THE AMTIE LINES
```

```
                                ;*  :  SET THE READ PATH CONTROL REGISTER TO BYPASS 1, PLO DISABLE AND STOP
                                ;*  :  CLEAR THE TIE BUS
                                ;*  :  CLEAR THE SCLK LATCH (SOFTWARE)
                                ;*  :  CLEAR THE XL WRITE CLOCK LATCH (SOFTWARE)
                                ;*  :  SET THE WRITE PATH CONTROL REGISTER TO WMC ENAB, XMC ENAB, WRITE, GCR
                                ;*  :  SET THE READ PATH CONTROL REGISTER TO READ AFTER WRITE GCR
                                ;*  :  BGNDO
                                ;*  :  :  SET THE LOOP COUNT TO 2
                                ;*  :  :  BGNDO
                                ;*  :  :  :  CALL SUBROUTINE CLKSYS
                                ;*  :  :  :  GET THE DBUS STATUS REGISTER
                                ;*  :  :  :  IF SCLOCK BIT=1
                                ;*  :  :  :  :  THEN-IF SCLK LATCH=1
                                ;*  :  :  :  :       :  THEN-CONTINUE
                                ;*  :  :  :  :       :  ELSE-CALL SUBROUTINE SETDDR
                                ;*  :  :  :  :       :       SET THE SCLK LATCH
                                ;*  :  :  :  :       ENDIF
                                ;*  :  :  :  :  ELSE-CLEAR THE SCLK LATCH
                                ;*  :  :  :  ENDIF
                                ;*  :  :  :  GET THE WMC STATUS
                                ;*  :  :  :  IF TRANSLATOR WRITE CLOCK BIT=1
                                ;*  :  :  :  :  THEN-IF XL WRITE CLOCK LATCH=1
                                ;*  :  :  :  :       :  THEN-CONTINUE
                                ;*  :  :  :  :       :  ELSE-CLEAR THE AMTIE LINES
                                ;*  :  :  :  :       :       SET THE XL WRITE CLOCK LATCH
                                ;*  :  :  :  :       ENDIF
                                ;*  :  :  :  :  ELSE-CONTINUE
                                ;*  :  :  :  ENDIF
                                ;*  :  :  :  DECREMENT THE LOOP COUNT
                                ;*  :  :  :  DO UNTIL THE LOOP COUNT=0
                                ;*  :  :  ENDDO
                                ;*  :  :  SINGLE STEP THE READ PATH
                                ;*  :  :  GET THE WRITE MICRO STATUS
                                ;*  :  :  DO UNTIL THE WMC NOT DONE AND XMC NOT DONE STATUS BITS=0
                                ;*  :  ENDDO
                                ;*  :  SET THE LOOP COUNT TO -2000
                                ;*  :  BGNDO
                                ;*  :  :  CALL SUBROUTINE CLKSYS
                                ;*  :  :  CALL SUBROUTINE CLKSYS
                                ;*  :  :  SINGLE STEP THE READ PATH
                                ;*  :  :  INCREMENT THE LOOP COUNT
                                ;*  :  :  DO UNTIL READ CHANNEL STATUS VALID=1 OR LOOP COUNT=0
                                ;*  :  ENDDO
                                ;*  :  IF LOOP COUNT=0
                                ;*  :  :  THEN-ERROR
                                ;*  :  :  ELSE-CONTINUE
                                ;*  :  ENDIF
                                ;*  :  IF ECC STATUS BIT CRC ERROR=1
                                ;*  :  :  THEN-ERROR
                                ;*  :  :  ELSE-CONTINUE
                                ;*  :  ENDIF
                                ;*  :  IF ECC STATUS BIT ACRC ERROR=1
                                ;*  :  :  THEN-ERROR
                                ;*  :  :  ELSE-CONTINUE
```

```
1505                                    ;*  :  ENDIF
1506                                    ;*  :  INCREMENT THE BYTE COUNT CONTROL BYTE
1507                                    ;*  :  DO UNTIL THE BYTE COUNT CONTROL BYTE=1
1508                                    ;*  ENDDO
1509                                    ;*ENDTST
1510  4357                              SE
 (1)                                    ; ****************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1511                                    ;*CCM1 MICRO TEST 02
1512                                    ;*CCM1 MICRO ERROR 03
1513                                    ;*CCM1-CHECK CHARACTER CONTROLLER-CRC AND ACRC TEST
1514                                    ;*M8952
1515                                    ;*OPERATOR ERROR NO TM78 UNITS SELECTED
1516                                    ;*FATAL ERROR - TEST ABORTED
1517                                    ;*
1518                                    ;*CCM1 MICRO TEST 02
1519                                    ;*CCM1 MICRO ERROR 04
1520                                    ;*CCM1-CHECK CHARACTER CONTROLLER-CRC AND ACRC TEST
1521                                    ;*M8952
1522                                    ;*READ CHANNEL ''STATUS VALID'' NOT SET AFTER XMC DONE
1523                                    ;*FATAL ERROR - TEST ABORTED
1524                                    ;*BYTE-SCLK COUNT = LLL
1525                                    ;*DATA FORMAT = MM
1526                                    :*SKIP COUNT = NN
1527                                    ;*
1528                                    ;*CCM1 MICRO TEST 02
1529                                    ;*CCM1 MICRO ERROR 05
1530                                    ;*CCM1-CHECK CHARACTER CONTROLLER-CRC AND ACRC TEST
1531                                    ;*M8952
1532                                    ;*CRC ERROR AFTER DATA TRANSFER
1533                                    ;*BYTE-SCLK COUNT = LLL
1534                                    ;*DATA FORMAT = MM
1535                                    ;*SKIP COUNT = NN
1536                                    ;*
1537                                    ;*CCM1 MICRO TEST 02
1538                                    ;*CCM1 MICRO ERROR 06
1539                                    ;*CCM1-CHECK CHARACTER CONTROLLER-CRC AND ACRC TEST
1540                                    ;*M8952
1541                                    ;*ACRC ERROR AFTER DATA TRANSFER
1542                                    ;*BYTE-SCLK COUNT = LLL
1543                                    ;*DATA FORMAT = MM
1544                                    ;*SKIP COUNT = NN
1545  4357                              S
 (1)                                    ; ****************************************************************************
1546  4357                              TST2:   TESTX   @2
 (1)  4357  3E  02            7.0                 MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4359  CD  03  28       18.0                 CALL    TSET            ;SETUP THE TEST
1547                                    ;%CCM1-CHECK CHARACTER CONTROLLER-CRC AND ACRC TEST
1548                                    ;&M8952
1549                                    ;REQUEST THE HOST FOR THE PORTS TO BE USED IN THIS TEST...VALUE WILL
1550                                    ;BE IN CAS REG 12 LOW BYTE ON RETURN FROM HOST REQUEST.
1551
1552  435C                                      REQ     @7,0,0,0
```

```
 (1)  435C  CD  06  28      18.0                          CALL    REQST
 (1)  435F  00                                            .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4360  00  00                                        .WORD   0               ;SYSTEM "0" COUNT
 (1)  4362  00  00                                        .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  4364  00                                            .BYTE                   ;DATA COMPARE FLAG IF =1
 (1)  4365  07                                            .BYTE   @7              ;REQUEST CODE
1553  4366                                        RIN     R12L
 (1)  4366  DB  94          10.0                          IN      R12L            ;READ R12L INTO AC
 (1)  4368  7F               4.0                          MOV     A,A             ;RETRY LINK
1554  4369  32  CF  45      13.0                  STA     UNITMP          ;SAVE THE UNIT MAP INFO FROM HOST
1555  436C  AF               4.0          T2LPB:   XRA     A
1556  436D  32  CD  45      13.0                  STA     TPBCNT          ;CLEAR THE TEST PASS/BYTE COUNT
1557  4370  CD  8E  45      18.0          T2LPA:   CALL    CLEAR
1558  4373  06  00           7.0                  MVI     B,0
1559  4375  3A  CF  45      13.0                  LDA     UNITMP          ;GET SELECTED UNIT MAP
1560  4378  E6  01           7.0                  ANI     $01             ;UNIT 0 PRESENT?
1561  437A  C2  A6  43      10.0                  JNZ     FOUND           ;YES-USE IT
1562  437D  04               4.0                  INR     B
1563  437E  3A  CF  45      13.0                  LDA     UNITMP          ;
1564  4381  E6  02           7.0                  ANI     $02             ;UNIT 1 PRESENT?
1565  4383  C2  A6  43      10.0                  JNZ     FOUND           ;YES-USE IT
1566  4386  04               4.0                  INR     B
1567  4387  3A  CF  45      13.0                  LDA     UNITMP          ;
1568  438A  E6  04           7.0                  ANI     $04             ;UNIT 2 PRESENT?
1569  438C  C2  A6  43      10.0                  JNZ     FOUND           ;YES-USE IT
1570  438F  04               4.0                  INR     B
1571  4390  3A  CF  45      13.0                  LDA     UNITMP
1572  4393  E6  08           7.0                  ANI     $08             ;UNIT 3 PRESENT
1573  4395  C2  A6  43      10.0                  JNZ     FOUND           ;YES-USE IT
1574  4398                                        ERR     EXIT,DUMMY
 (1)                                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4398  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0003                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  439B  03                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  439C  00                                          .BYTE
 (1)  439D  CD  15  28      18.0                  DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  43A0  DA  4F  45      10.0                          JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1575                                        ;>OPERATOR ERROR NO TM78 UNITS SELECTED
1576                                        ;>FATAL ERROR - TEST ABORTED
1577  43A3  C3  4F  45      10.0                  JMP     EXIT
1578                                        ;SET UP FOR THE DATA TRANSFER
1579  43A6  DB  E0           10.0          FOUND:   IN      INTSTA
1580  43A8  E6  80            7.0                  ANI     BIT7
1581  43AA  B0                4.0                  ORA     B
1582  43AB  D3  E0           10.0                  OUT     MBSEL
1583  43AD  AF                4.0                  XRA     A
1584  43AE  D3  40           10.0                  OUT     TCMD            ;CLEAR TU COMMAND REGISTER
1585  43B0  3E  14            7.0                  MVI     A,P.WPEN!P.RPEN ;ENABLE THE WRITE AND READ PATHS
1586  43B2  D3  4C           10.0                  OUT     PENAB
1587  43B4  AF                4.0                  XRA     A
1588  43B5  D3  D2           10.0                  OUT     TRKENA          ;CLEAR ALL TRACK ENABLE BITS
1589  43B7  D3  D2           10.0                  OUT     TRKENA
1590  43B9  D3  44           10.0                  OUT     TAMT            ;CLEAR THE AMTIE LINES
```

```
1592                                    ;RESTART READ MICROCONTROLLER ROM PROGRAM
1593  43BB  3E  10          7.0  TST2L:  MVI   A,RDCLK          ;SET NORMAL READ PATH CLOCKS
1594  43BD  D3  F0         10.0          OUT   CLKCTL
1595
1596  43BF  3E  10          7.0          MVI   A,W.GCR          ;SET MACHINE IN GCR MODE
1597  43C1  D3  D3         10.0          OUT   WMCCTL
1598
1599  43C3  3E  20          7.0          MVI   A,P.LWR          ;SET THE PORT CONTROL TO
1600  43C5  D3  48         10.0          OUT   PDIAG            ;LOOP WRITE/READ
1601
1602  43C7  3E  A8          7.0  TST02C: MVI   A,R.PLO1!R.PLOD!R.TBJN  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1603                                                           ;SET TIE BUS JAM AND PLO DISABLE
1604  43C9  D3  09         10.0          OUT   RPCTL
1605  43CB  AF              4.0          XRA   A
1606  43CC  D3  0A         10.0          OUT   RTIEB            ;CLEAR THE TIE BUS
1607  43CE  D3  08         10.0          OUT   RFIFOL           ;CLOCK THE FIFO'S
1608  43D0  3E  0D          7.0          MVI   A,RCLRT          ;ISSUE CLEAR ALL COMMAND
1609  43D2  D3  0B         10.0          OUT   RCMD
1610
1611  43D4  00              4.0          NOP                    ;WAIT
1612  43D5  00              4.0          NOP                    ;WAIT
1613  43D6  00              4.0          NOP
1614  43D7  00              4.0          NOP
1615  43D8  00              4.0          NOP
1616  43D9  3E  A9          7.0          MVI   A,R.PLO1!R.STPC!R.PLOD!R.TBJN
1617  43DB  D3  09         10.0          OUT   RPCTL
```

```
1619                                        ;RESTART WRITE MICROCONTROLLER ROM PROGRAM
1620
1621  43DD  3E  89            7.0           MVI     A,@211          ;
1622  43DF  D3  DB           10.0           OUT     DDRCTL
1623  43E1  AF                4.0           XRA     A               ;SET NORMAL WRITE PATH CLOCKS
1624  43E2  D3  F0           10.0           OUT     CLKCTL
1625  43E4  3E  01            7.0           MVI     A,W.RST         ;GET 'RST' BIT
1626  43E6  D3  D3           10.0           OUT     WMCCTL          ;RESTART WMC
1627  43E8  3E  15            7.0           MVI     A,SSTEP+RDCLK   ;SET UP SINGLE STEP WRITE CLOCK
1628  43EA  D3  F0           10.0           OUT     CLKCTL          ;AND +10% READ CLOCK
1629  43EC  32  C5  45       13.0           STA     CCTLWD          ;SAVE COPY
1630  43EF  AF                4.0           XRA     A               ;FINISH THE WMC RESTART
1631  43F0  D3  D3           10.0           OUT     WMCCTL
1632  43F2  3E  30            7.0           MVI     A,@60
1633  43F4  D3  D7           10.0           OUT     CNTCTL          ;SELECT BYTE COUNTER
1634  43F6  3E  70            7.0           MVI     A,@160
1635  43F8  D3  D7           10.0           OUT     CNTCTL          ;SELECT THE PAD COUNTER
1636  43FA  3E  B0            7.0           MVI     A,@260
1637  43FC  D3  D7           10.0           OUT     CNTCTL          ;SELECT ECODE COUNTER
1638
1639                                        ;CLEAR ERROR CODE COUNTER
1640  43FE                                  CLRECT                  ;CLEAR ECODE COUNTER
 (1)  43FE  AF                4.0                   XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  43FF  D3  D6           10.0                   OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  4401  D3  D6           10.0                   OUT     ERRCNT          ;CLEAR BITS 15-8
1641  4403                                  CLRBCT
 (1)  4403  AF                4.0                   XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  4404  D3  D4           10.0                   OUT     BYTCNT          ;CLEAR BITS 7-0
 (1)  4406  D3  D4           10.0                   OUT     BYTCNT          ;CLEAR BITS 15-8
1642  4408                                  CLRPCT
 (1)  4408  AF                4.0                   XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  4409  D3  D5           10.0                   OUT     PADCNT          ;CLEAR BITS 7-0
 (1)  440B  D3  D5           10.0                   OUT     PADCNT          ;CLEAR BITS 15-8
1643
1644                                        ;SET DDR CONTROL TO "OUT" FOR DATA INJECTION INTO BAL LOGIC
1645  440D  3A  CD  45        13.0           LDA     TPBCNT          ;GET THE BYTE COUNT
1646  4410  3C                4.0           INR     A
1647  4411  32  CD  45        13.0           STA     TPBCNT          ;SAVE THE NEW BYTE COUNT TO USE
1648  4414                                  ROUT    R05L
 (1)  4414  D3  8A           10.0                   OUT     R05L            ;WRITE AC INTO R05L
 (1)  4416  7F                4.0                   MOV     A,A             ;RETRY LINK
1649  4417  3E  88            7.0           MVI     A,@210          ;SET "OUT" BITS
1650  4419  D3  DB           10.0           OUT     DDRCTL          ;LOAD DDR CONTROL
1651
1652  441B  3A  CD  45        13.0           LDA     TPBCNT          ;GET THE BYTE COUNT TO USE
1653  441E  32  23  4F        13.0           STA     BYTEL           ;SAVE THE BYTE COUNT LOW VALUE
1654  4421  AF                4.0           XRA     A
1655  4422                                  ROUT    R05H
 (1)  4422  D3  8B           10.0                   OUT     R05H            ;WRITE AC INTO R05H
 (1)  4424  7F                4.0                   MOV     A,A             ;RETRY LINK
1656  4425                                  ROUT    R02L
 (1)  4425  D3  84           10.0                   OUT     R02L            ;WRITE AC INTO R02L
 (1)  4427  7F                4.0                   MOV     A,A             ;RETRY LINK
1657  4428  32  24  4F        13.0           STA     BYTEH           ;CLEAR THE BYTE COUNT HIGH VALUE
```

```
1658  442B  32  D2  45   13.0        STA   DPNUM          ;INITIALIZE THE DATA PATTERN
1659  442E  CD  46  00   18.0        CALL  LKMOD7         ;CALL MOD7 ROUTINE FROM CPR U-CODE
1660  4431  3A  CD  45   13.0        LDA   TPBCNT         ;GET THE BC AGAIN
1661  4434  3D              4.0      DCR   A              ;CALC MOD32
1662  4435  E6  1F              7.0  ANI   @37            ;SAVE LOW 5 BITS
1663  4437  57              4.0      MOV   D,A            ;TEMP SAVE
1664  4438  78              4.0      MOV   A,B            ;GET MOD7 REMAINDER FROM LKMOD7
1665  4439  0F              4.0      RRC                  ;POSITION THE MOD7 NUMBER
1666  443A  0F              4.0      RRC
1667  443B  0F              4.0      RRC
1668  443C  B2              4.0      ORA   D              ;MAKE THE RESID CHAR WORD
1669  443D  D3  D1           10.0     OUT   RESCHR         ;LOAD THE RESID CHAR WORD IN WMC
1670  443F  3A  CD  45   13.0        LDA   TPBCNT         ;GET THE BC AGAIN
1671  4442  90              4.0      SUB   B              ;CALC THE VALUE LOADED INTO THE BYTE COUNTER
1672  4443  3E  06              7.0  MVI   A,6
1673  4445  90              4.0      SUB   B              ;CALC THE PAD COUNTER VALUE
1674  4446  32  CC  45   13.0        STA   IMPADCT        ;SAVE THE PAD COUNT TO USE
1675  4449  3C              4.0      INR   A              ;ADD 1 MORE FOR THE INTEL 8253 CHIP
1676  444A  D3  D5           10.0     OUT   PADCNT         ;LOAD PAD COUNT 7-0
1677  444C  AF              4.0      XRA   A
1678  444D  D3  D5           10.0     OUT   PADCNT         ;LOAD PAD CCOUNTER 15-8
1679  444F  3A  CC  45   13.0        LDA   IMPADCT        ;GET PAD COUNT AGAIN
1680  4452  21  CD  45   10.0        LXI   H,TPBCNT       ;POINT TO BC TO USE
1681  4455  86              7.0      ADD   M              ;CALC # TO SEE IF PADCRC BIT NEEDED
1682  4456  3C              4.0      INR   A              ;IN THE DATACTL WORD
1683  4457  0F              4.0      RRC                  ;LOAD THE PADCRC BIT - IF =1 THEN PAD
1684  4458  E6  80              7.0  ANI   BIT7           ;SAVE ONLY THIS BIT
1685  445A  32  CE  45   13.0        STA   PDCRC          ;SAVE AS A FLAG
1686
1687                                ;LOAD DATA CONTROL WITH SELECTED FORMAT CODE
1688  445D  3E  50            7.0      MVI   A,$50          ;LOAD THE IMAGE MODE FORMAT CODE
1689  445F                             ROUT  R02H           ;ALSO SAVE IN CAS REG 02
 (1)  445F  D3  85           10.0                OUT   R02H       ;WRITE AC INTO R02H
 (1)  4461  7F               4.0                 MOV   A,A        ;RETRY LINK
1690  4462  47              4.0      MOV   B,A            ;SAVE THE FMT DATA
1691  4463  3A  CE  45   13.0        LDA   PDCRC          ;GET THE PAD CRC BIT
1692  4466  B0              4.0      ORA   B              ;ADD IN THE FORMAT NUMBER
1693  4467  D3  D0           10.0     OUT   DATACTL        ;LOAD DATA CONTROL WORD
1694
1695                                ;ENABLE WRITE MICROCONTROLLER   START OF DATA XFR - WRITE
1696  4469  3E  FF            7.0      MVI   A,@377
1697  446B  D3  D2           10.0      OUT   TRKENA         ;ENABLE THE TRACKS FROM
1698  446D  D3  D2           10.0      OUT   TRKENA         ;THE TRANSLATOR
1699  446F  D3  44           10.0      OUT   TAMT           ;SET THE AMTIE LINES
1700  4471  3E  21            7.0      MVI   A,P.LWR!P.AMTP ;SET THE PARITY AMTIE
1701  4473  D3  48           10.0      OUT   PDIAG          ;TO ONE
1702  4475  3E  19            7.0      MVI   A,R.PLOO!R.PLOD!R.STPC
1703  4477  D3  09           10.0      OUT   RPCTL
```

```
1705  4479  AF                    4.0           XRA     A
1706  447A  D3  0A                10.0          OUT     RTIEB           ;CLEAR THE TIE BUS
1707  447C  32  D1  45            13.0          STA     CLKON           ;CLEAR THE SCLK LATCH
1708  447F  32  D0  45            13.0          STA     XLCLK           ;CLEAR THE XL WRITE CLOCK LATCH
1709  4482  3E  D8                 7.0           MVI     A,W.ENAB!W.WRITE!W.GCR!X.ENAB  ;SELECT 'WMC EN' + 'WRITE'
1710  4484  D3  D3                10.0          OUT     WMCCTL          ;LOAD WRITE MICROCONTROLLER CONTROL
1711  4486  3E  07                 7.0           MVI     A,READG         ;LOAD THE GCR READ COMMAND
1712  4488  D3  0B                10.0          OUT     RCMD            ;START THE READ PATH
1713  448A  0E  02                 7.0   T2LOP1: MVI     C,@2            ;LOAD THE LOOP COUNT
1714  448C  CD  B9  45            18.0   T2LOP4: CALL    CLKSYS          ;SINGLE STEP THE MACHINE
1715  448F  DB  C0                10.0          IN      DBUSSTA         ;GET DATA BUS STATUS
1716  4491  E6  02                 7.0           ANI     T.SCLK          ;SCLOCK SET?
1717  4493  C2  9D  44            10.0          JNZ     1$              ;YES-GO PROCESS
1718  4496  AF                    4.0           XRA     A               ;NO-CLEAR THE SCLOCK LATCH
1719  4497  32  D1  45            13.0          STA     CLKON
1720  449A  C3  AC  44            10.0          JMP     2$              ;CONTINUE
1721  449D  3A  D1  45            13.0   1$:     LDA     CLKON           ;IF THE SCLOCK LATCH NOT=0
1722  44A0  A7                    4.0           ANA     A               ;
1723  44A1  C2  AC  44            10.0          JNZ     2$              ;THEN-CONTINUE
1724  44A4  CD  52  45            18.0          CALL    SETDDR          ;ELSE-LOAD DATA INTO THE DDR REGISTERS
1725  44A7  3E  FF                 7.0           MVI     A,@377          ;SET THE SCLOCK LATCH
1726  44A9  32  D1  45            13.0          STA     CLKON
1727  44AC  DB  D0                10.0   2$:     IN      WMCSTA          ;GET THE WMC STATUS
1728  44AE  E6  01                 7.0           ANI     X.WCLK          ;TRANSLATOR WRITE CLOCK=1
1729  44B0  CA  C5  44            10.0          JZ      3$              ;NO-CONTINUE
1730  44B3  3A  D0  45            13.0          LDA     XLCLK           ;FIRST OCCURANCE OF THE CLOCK?
1731  44B6  A7                    4.0           ANA     A               ;SET CONDITION BITS
1732  44B7  C2  C5  44            10.0          JNZ     3$              ;NO-THEN CONTINUE
1733  44BA  D3  44                10.0          OUT     TAMT            ;CLEAR THE AMTIE LINES
1734  44BC  3E  20                 7.0           MVI     A,P.LWR         ;CLEAR AMTIE PARITY
1735  44BE  D3  48                10.0          OUT     PDIAG
1736  44C0  3E  FF                 7.0           MVI     A,@377          ;SET THE TRANSLATOR CLOCK FLAG
1737  44C2  32  D0  45            13.0          STA     XLCLK
1738  44C5  0D                    4.0   3$:     DCR     C               ;DECREMENT THE LOOP COUNT
1739  44C6  C2  8C  44            10.0          JNZ     T2LOP4          ;CONTINUE UNTIL ZERO
1740  44C9  D3  0C                10.0          OUT     RINST           ;SINGLE STET THE READ PATH
1741  44CB  DB  D0                10.0          IN      WMCSTA          ;GET THE WRITE MICRO STATUS
1742  44CD  E6  C0                 7.0           ANI     $C0             ;XMC DONE?
1743  44CF  C2  8A  44            10.0          JNZ     T2LOP1          ;NO-CONTINUE
1744  44D2  3E  FF                 7.0           MVI     A,$FF           ;YES-SET THE AMTIES
1745  44D4  D3  44                10.0          OUT     TAMT            ;TO ONES
1746  44D6  3E  21                 7.0           MVI     A,P.LWR!P.AMTP  ;SET THE PARITY AMTIE
1747  44D8  D3  48                10.0          OUT     PDIAG           ;TO ONE
1748  44DA  21  30  F8            10.0          LXI     H,-2000         ;LOAD THE WATCHDOG LOOP COUNT
1749  44DD  CD  B9  45            18.0   T2LOP2: CALL    CLKSYS          ;CLOCK THE WRITE PATH
1750  44E0  CD  B9  45            18.0          CALL    CLKSYS          ;CLOCK THE WRITE PATH
1751  44E3  D3  0C                10.0          OUT     RINST           ;CLOCK THE READ PATH
1752  44E5  DB  01                10.0          IN      RPCHI           ;GET THE READ CHANNEL STATUS
1753  44E7  E6  40                 7.0           ANi     $40             ;STATUS VALID SET?
1754  44E9  C2  02  45            10.0          JNZ     T2LOP3          ;YES-GET OUT
1755  44EC  0E  01                 7.0           MVI     C,@1            ;SET UP A 16-BIT INCREMENT REGISTER
1756  44EE  06  00                 7.0           MVI     B,@0            ;
1757  44F0  09                    10.0          DAD     B               ;INCREMENT THE LOOP COUNT
1758  44F1  D2  DD  44            10.0          JNC     T2LOP2          ;CONTINUE UNTIL ZERO
```

```
1760  44F4                                        ERR     T2LPA,DUMMX,3
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44F4  CD  09  28        18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0004                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44F7  04                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44F8  03                                           .BYTE   3
 (1)  44F9  CD  15  28         8.0                DUMMX:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44FC  DA  70  43        10.0                        JC      T2LPA           ;LOOP ADDRESS IF LOOP SPECIFIED
1761                                      ;>READ CHANNEL "STATUS VALID" NOT SET AFTER XMC DONE
1762                                      ;<FATAL ERROR - TEST ABORTED!
1763  44FF  C3  4F  45        10.0                JMP     EXIT
1764  4502  DB  1A            10.0        T2LOP3: IN      ECCSTA          ;GET THE ECC STATUS
1765  4504  E6  80             7.0                ANI     E.CRC           ;CRC ERROR?
1766  4506  CA  0E  45        10.0                JZ      T2CN3           ;NO-CONTINUE
1767  4509                                        ERR     T2LPA,T2CN3,3
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4509  CD  09  28        18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0005                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  450C  05                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  450D  03                                           .BYTE   3
 (1)  450E  CD  15  28        18.0                T2CN3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4511  DA  70  43        10.0                        JC      T2LPA           ;LOOP ADDRESS IF LOOP SPECIFIED
1768                                      ;>CRC ERROR AFTER DATA TRANSFER
1769  4514  DB  1A            10.0                IN      ECCSTA          ;GET THE ECC STATUS
1770  4516  E6  10             7.0                ANI     E.ACRC          ;ACRC ERROR?
1771  4518  CA  20  45        10.0                JZ      T2CN4           ;NO-CONTINUE
1772  451B                                        ERR     T2LPA,T2CN4,3
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  451B  CD  09  28        18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0006                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  451E  06                                           .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  451F  03                                           .BYTE   3
 (1)  4520  CD  15  28        18.0                T2CN4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4523  DA  70  43        10.0                        JC      T2LPA           ;LOOP ADDRESS IF LOOP SPECIFIED
1773                                      ;>ACRC ERROR AFTER DATA TRANSFER
1774  4526                                        REQ     @7,0,0,0                ;DUMMY REQUEST TO KEEP THE HOST CPU ALIVE
 (1)  4526  CD  06  28        18.0                        CALL    REQST
 (1)  4529  00                                           .BYTE   0               ;DATA PATTERN NUMBER
 (1)  452A  00  00                                       .WORD   0               ;SYSTEM "0" COUNT
 (1)  452C  00  00                                       .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  452E  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (1)  452F  07                                           .BYTE   @7              ;REQUEST CODE
1775  4530  3A  CD  45        13.0                LDA     TPBCNT          ;GET THE BYTE COUNT
1776  4533  3C                 4.0                INR     A               ;INCREMENT
1777  4534  32  CD  45        13.0                STA     TPBCNT          ;SAVE THE NEW BYTE COUNT
1778  4537  A7                 4.0                ANA     A               ;SET THE CONDITION BITS
1779  4538  C2  70  43        10.0                JNZ     T2LPA           ;CONTINUE UNTIL A 256 BYTE RECORD IS WRITTEN
1780  453B                                        ENDTST  T2LPB
 (1)                                      ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  453B                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  453B  CD  06  28        18.0                        CALL    REQST
 (2)  453E  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  453F  00  00                                       .WORD                   ;SYSTEM "" COUNT
 (2)  4541  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
```

```
   (2)  4543   00                                                  .BYTE                   ;DATA COMPARE FLAG IF =1
   (2)  4544   07                                                  .BYTE   7               ;REQUESt CODE
   (1)  4545   3A   9A   4F      13.0                    LDA      ITERA                   ;GET ITERATION COUNT
   (1)  4548   3D                 4.0                    DCR      A                       ;DOWNCOUNT
   (1)  4549   32   9A   4F      13.0                    STA      ITERA                   ;SAVE COUNT
   (1)  454C   F2   6C   43      10.0                    JP       T2LPB                   ;DO TEST UNTIL TILL = 0
 1781
 1782  454F   C3   18   28      10.0  EXIT:             JMP      TSTEND
```

```
                                .SBTTL  SUBROUTINE SETDDR
                                $
                                ; ************************************************************************
                                ;GENDAT -- ROUTINE TO GENERATE DDR REG DATA FOR NEXT "SCLK" CYCLE.
                                ;       DATA IS LOADED IN DDR REGS AND A COPY IS STORED IN 'SDDRAC', +
                                ;       'SDDRCC' FOR REFERENCE.
                                $
                                ; ************************************************************************

4552 F5              12.0  SETDDR: PUSH    PSW             ;SAVE PSW + A
4553 C5              12.0          PUSH    B               ;SAVE B + C
4554 E5              12.0          PUSH    H               ;SAVE H + L
4555 3A D2 45        13.0          LDA     DPNUM           ;GET THE LAST PATTERN # USED
4558 3C               4.0          INR     A               ;POINT TO NEXT
4559 32 D2 45        13.0          STA     DPNUM           ;SAVE NUMBER
455C 32 C9 45        13.0          STA     SDDRAT          ;SAVE DDR 'A' IMAGE
455F AF               4.0          XRA     A               ;
4560 32 CA 45        13.0          STA     SDDRBT          ;SAVE DDR 'B' IMAGE
4563 32 CB 45        13.0          STA     SDDRCT          ;SAVE DDR 'C' IMAGE

4566 3A C9 45        13.0          LDA     SDDRAT          ;GET DDRA TEMP
4569 E6 3F            7.0          ANI     @77             ;STRIP 2 BITS OFF
456B 07               4.0          RLC
456C 07               4.0          RLC                     ;LEFT JUSTIFY
456D 47               4.0          MOV     B,A             ;TEMP SAVE
456E 3A CB 45        13.0          LDA     SDDRCT          ;GET DDRC TEMP
4571 E6 03            7.0          ANI     @3              ;SAVE 2 BITS
4573 B0               4.0          ORA     B
4574 32 C6 45'       13.0          STA     SDDRA           ;SAVE DDRA DESIRED
4577 D3 D8           10.0          OUT     DDRA            ;LOAD THE REAL DDRA

4579 3A C9 45        13.0          LDA     SDDRAT          ;GET DDRA AGAIN
457C E6 C0            7.0          ANI     @300            ;SAVE 2 BITS
457E 07               4.0          RLC
457F 07               4.0          RLC
4580 32 C7 45        13.0          STA     SDDRB           ;SAVE DDRB DATA
4583 D3 D9           10.0          OUT     DDRB            ;LOAD THE REAL DDRB

4585 3A C8 45        13.0          LDA     SDDRC
4588 D3 DA           10.0          OUT     DDRC
458A E1              10.0          POP     H               ;RESTORE REGS
458B C1              10.0          POP     B
458C F1              10.0          POP     PSW
458D C9              10.0          RET
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
458E                                    SSUB
 (1)                                    ; ************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
458E                                    SD
 (1)                                    ; ************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
458E                                    SP
 (1)                                    ; ************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNSUB
                                        ;*      SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*      CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*      CLEAR PORT SELECT FOR TRANSPORT
                                        ;*      CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*      CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*      CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
458E                                    S
 (1)                                    ; ************************************************************************

458E   F5                    12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
458F   C5                    12.0          PUSH    B
4590   06   00               7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
4592   DB   E0              10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
4594   E6   80               7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
4596   B0                     4.0          ORA     B               ;ADD IN THE SELECTED PORT #
4597   D3   E0              10.0          OUT     MBSEL           ;RESET TO THIS PORT
4599   3E   80               7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
459B   D3   40              10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
459D   AF                     4.0          XRA     A               ;CLEAR TU COMMAND A
459E   D3   40              10.0          OUT     TCMD            ;
45A0   3E   81               7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
45A2   D3   40              10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
45A4   3E   00               7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
45A6   D3   40              10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
45A8   AF                     4.0          XRA     A
45A9   D3   44              10.0          OUT     TAMT            ;CLEAR AMTIE WORD
45AB   D3   48              10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
45AD   D3   4C              10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
45AF   04                     4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
45B0   78                     4.0          MOV     A,B
45B1   FE   04                7.0          CPI     4               ;DONE?
45B3   C2   42   45          10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
45B6   C1                    10.0          POP     B               ;RESET B & C
45B7   F1                    10.0          POP     PSW             ;ALL DONE
45B8   C9                    10.0          RET                     ;EXIT
```

```
                                        .SBTTL  SUBROUTINE      CLKSYS
                                        $
 (1)                                    ; ************************************************************************
                                        ;CLKSYS -- ROUTINE TO CLOCK THE SYSTEM 1 TIME BY WRITING CLOCK CONTROL
                                        ;       'CLK' BIT TO A '1' THEN A '0'.
                                        $
 (1)                                    ; ************************************************************************

45B9  3A  C5  45        13.0  CLKSYS: LDA     CCTLWD          ;GET SOFTWARE CLOCK CONTROL IMAGE
45BC  F6  40             7.0          ORI     SSCLK           ;ADD IN 'CLK' BIT
45BE  D3  F0            10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL
45C0  E6  3F             7.0          ANI     @77             ;STRIP OFF CLOCK BIT
45C2  D3  F0            10.0          OUT     CLKCTL          ;LOAD CLOCK CONTROL WORD
45C4  C9                10.0          RET                     ;EXIT - CLOCK CYCLE COMPLETE
```

```
                                        .SBTTL  PROGRAM VARIABLES

                                        ;HERE IS THE NECESSARY STORAGE FOR TEST PARAMETERS AND DDR DATA STORAGE
45C5  00                                CCTLWD: .BYTE   0               ;IMAGE OF CLOCK CONTROL WORD
45C6  00                                SDDRA:  .BYTE   0               ;DDRA DATA ACTUALLY LOADED
45C7  00                                SDDRB:  .BYTE   0               ;DDRB DATA ACTUALLY LOADED
45C8  00                                SDDRC:  .BYTE   0               ;DDRC DATA ACTUALLY LOADED

45C9  00                                SDDRAT: .BYTE   0               ;DDRA DATA FROM PGM TABLE
45CA  00                                SDDRBT: .BYTE   0               ;DDRB DATA FROM PGM TABLE
45CB  00                                SDDRCT: .BYTE   0               ;DDRC DATA FROM PGM TABLE
45CC  00                                IMPADCT: .BYTE  0               ;SOFTWARE COPY OF THE PAD COUNT GIVEN TO THE WMC
45CD  00                                TPBCNT: .BYTE   0               ;TEST PASS BYTE COUNTER
45CE  00                                PDCRC:  .BYTE   0               ;PAD/CRC FLAG IF SET, WMC WILL ISSUE A PAD
45CF  00                                UNITMP: .BYTE   0               ;UNIT MAP
45D0  00                                XLCLK:  .BYTE   0               ;XL WRITE CLOCK FLAG
45D1  00                                CLKON:  .BYTE   0               ;SCLOCK ON
45D2  00                                DPNUM:  .BYTE   0               ;DATA PATTERN NUMBER
      0000                                      .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE    4F9B          ATTCD   4F97          AXNUM   4F91
B      =%0000          BADST  = 0090          BIT0   = 0001          BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008          BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080          BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00           BSAVE    4F9C          BYTCNT= 00D4          BYTEH   4F24
BYTEL    4F23          C      =%0001          CASCT   4F21          CASCTL= 00A0
CASSTA= 00AC           CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1
CBYTH  = 008B          CBYTL  = 008A          CCTLWD  45C5          CDG1H  = 0087
CDG1L  = 0086          CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095
CDG3L  = 0094          CDVTH  = 008D          CDVTL  = 008C          CHPTIE= 0028
CHOTIE= 0020           CH1TIE= 0021           CH2TIE= 0022           CH3TIE= 0023
CH4TIE= 0024           CH5TIE= 0025           CH6TIE= 0026           CH7TIE= 0027
CKLOP  = 2815          CLEAR    458E          CLKCTL= 00F0          CLKON   45D1
CLKSYS  45B9           CLOCK    4F26          CLRLP   4592          CMC0H  = 0099
CMC0L  = 0098          CMC1H  = 009B          CMC1L  = 009A          CMC2H  = 009D
CMC2L  = 009C          CMC3H  = 009F          CMC3L  = 009E          CMINH  = 0097
CMINL  = 0096          CNTCTL= 00D7           CRCWRD= 0018          CSAVE   4F9D
CSRLH  = 0091          CSRLL  = 0090          CTCH   = 0085          CTCL   = 0084
CTSTH  = 008F          CTSTL  = 008E          CXCTH  = 0081          CXCTL  = 0080
CXINH  = 0083          CXINL  = 0082          C.     = 0001          C.AVAI= 0080
C.DP   = 0008          C.DSE  = 0010          C.DTU  = 0003          C.DVA  = 0008
C.FAIL= 00FC           C.FMT  = 0070          C.FNCT= 003E          C.GO   = 0001
C.INTC= 00FE           C.MAIN= 0020           C.NSA  = 0080          C.RCT  = 00FC
C.SER  = 0080          C.SHR  = 0040          C.SKPC= 000F          C.TAPE= 0040
C.WCS  = 0002          D      =%0002          DATACT= 00D0          DBUS    4F28
DBUSCT= 00C0           DBUSST= 00C0           DDRA   = 00D8          DDRAIN= 0010
DDRB   = 00D9          DDRBIN= 0002           DDRC   = 00DA          DDRCIN= 0001
DDRCO  = 0088          DDRCTL= 00DB           DIAFLG  4F22          DIAGPG= 4300
DIAGRM= 4F90           DIARD  = 00DB          DONE1  = 0045          DONINT= 0010
DPNUM    45D2          DSAVE    4F9E          DSE    = 0006          DUMMX   44F9  G
DUMMY    439D  G       D.ATH0= 0001           D.ATH1= 0002          D.EOTD= 0010
D.LAGC= 0020           D.NOTW= 0040           D.NTHR= 0004          D.TACH= 0008
D.WR4  = 0080          E      =%0003          ECCBAD= 0042          ECCCOR= 0019
ECCOK  = 0041          ECCSTA= 001A           ECCTST= 000E          EDATA  = 0095
EOTCLR= 0003           ERFLG    4F93          ERLP   = 2809          ERLPA  = 280F
ERLPB  = 2812          ERLPE  = 280C          ERNUM   4F90          ERRCNT= 00D6
ESAVE    4F9F          EXIT     454F          E.ACRC= 0010          E.AMT  = 0020
E.CDP  = 0080          E.CRC  = 0080          E.PNTR= 0008          E.RPE  = 0040
E.STEC= 0001           E.TTEC= 0002           E.UNC  = 0004          FIFORD= 006A
FORMAT   4F25          FOUND    43A6          FWDTST= 0061          GCRID  = 0089
GCRSET= 0002           GOODTM= 0092           H      =%0004          HLSAVE  4FA0
IE     = 0008          IMPADC   45CC          INTSTA= 00E0          ITERA   4F9A
I.PWR  = 0020          I.RMPE= 0040           I5.5   = 0010          I6.5   = 0020
I7.5   = 0040          KCALL  = 005F          KCLR   = 007B          KDEP   = 003F
KENAB  = 0078          KEXAM  = 003E          KEYBRD= 00C8          KEY1   = 0078
KEY10  = 006D          KEY11  = 006E          KEY12  = 006F          KEY13  = 005C
KEY14  = 005D          KEY15  = 005E          KEY16  = 005F          KEY17  = 003C
KEY18  = 003D          KEY19  = 0C3E          KEY2   = 0079          KEY20  = 003F
KEY3   = 007A          KEY4   = 007B          KEY5   = 0074          KEY6   = 0075
KEY7   = 0076          KEY8   = 0077          KEY9   = 006C          KINTA  = 006F
KLDAD  = 003D          KN0    = 003C          KN1    = 005C          KN2    = 005D
KN3    = 005E          KN4    = 006C          KN5    = 006D          KN6    = 006E
```

```
KN7    = 0074          KN8    = 0075          KN9    = 0076          KU2    = 0079
KU3    = 007A          KU8    = 0077          L      =%0005          LBLANK= 000F
LCE    = 000B          LCH    = 000C          LCL    = 000D          LCP    = 000E
LC0    = 0000          LC1    = 0001          LC2    = 0002          LC3    = 0003
LC4    = 0004          LC5    = 0005          LC6    = 0006          LC7    = 0007
LC8    = 0008          LC9    = 0009          LDLEDA= 00CA          LDLEDB= 00CB
LDLEDC= 00CC          LDLEDD= 00CD          LDLEDE= 00CE          LDLEDF= 00CF
LKDIAG= 2800          LKKBD  = 004C          LKKEY  = 0049          LKLWMG= 0053
LKLWMP= 0055          LKLWPG= 0052          LKLWPP= 004F          LKMOD7= 0046
LKOPR  = 0046          LPFLG   4F94          LPNUM   4F92          M      =%0006
MBSEL  = 00E0          MB.A   = 0008          MB.B   = 0004          MEMTOP= 4FFF
MINUS  = 000A          MM     = 8000          MSE    = 0008          MSGN   = 0006
MTACLR= 0000          MT.ARA= 0020          MT.CPE= 0080          MT.DSE= 0001
MT.FWD= 0040          MT.INH= 0008          MT.LWR= 0004          MT.MOT= 0002
MT.NWT= 0080          MT.PEC= 0040          MT.PSB= 0004          MT.PS0= 0001
MT.PS1= 0002          MT.REV= 0020          MT.WRT= 0010          MT.Z   = 0008
M.ATA  = 0080          M.CAPE= 0020          M.CONT= 0080          M.DEM  = 0020
M.EBL  = 0010          M.EXC  = 0008          M.FAIL= 0008          M.ILR  = 0010
M.INIT= 0010          M.OCC  = 0020          M.ONLI= 0001          M.PE   = 0040
M.PORT= 0080          M.RDEN= 0002          M.RDPE= 0008          M.RUN  = 0004
M.SCLK= 0001          M.TRA  = 0040          M.UNIT= 0007          M.WCLK= 0040
M.WCLN= 0080          M.WREN= 0080          M5.5   = 0001          M6.5   = 0002
M7.5   = 0004          NOTCAP= 0088          OKAY   = 00FF          OPRRAM= 4300
OPSTRT= 0058          OPVER  = 0040          PADCNT= 00D5          PADCRC= 0080
PDCRC   45CE          PDIAG  = 0048          PEID   = 008A          PENAB  = 004C
PESET  = 0001          PL     = 00B1          PRDD   = 004C          PRENF  = 009C
PS     = 00B2          PSTAT  = 0048          PSW    =%0009          P.AMTP= 0001
P.BCTC= 0040          P.CMDP= 0020          P.INTE= 0080          P.LCS  = 0040
P.LWR  = 0020          P.RDP  = 0002          P.RPEN= 0004          P.RPST= 0002
P.RPOE= 0020          P.RP1E= 0010          P.RP2E= 0020          P.RP3E= 0010
P.SING= 0080          P.STAT= 0002          P.STPE= 0080          P.TACH= 0008
P.TUPR= 0010          P.WCSP= 0004          P.WDS  = 0040          P.WFLP= 0001
P.WPEN= 0010          P.WPOE= 0008          P.WP1E= 0004          P.WP2E= 0008
P.WP3E= 0004          P.5VOK= 0002          QUE    = 281B          QUEM   = 281E
RAMT   = 0010          RARA   = 0006          RARAI  = 0004          RCHBD0= 0048
RCHBD1= 0047          RCHOK  = 0046          RCHTST= 000C          RCLRT  = 000D
RCMD   = 000B          RCMLP  = 0003          RCONT  = 0080          RDATA  = 0017
RDCLK  = 0010          RDON   = 0011          READG  = 0007          REND   = 0014
REQST  = 2806          RESCHR= 00D1          REVTST= 0064          REWIND= 0004
RFIFOL= 0008          RGCLK  = 0002          RGCRI  = 0003          RIBG   = 0001
RILL   = 0012          RINST  = 000C          RMCTST= 0008          RMK2   = 0013
RNOP   = 0000          RPATH  = 0001          RPBAD  = 0044          RPCHI  = 0001
RPCLK  = 0003          RPCTL  = 0009          RPEI   = 0002          RPFAIL= 0000
RPF1   = 009D          RPF2   = 009E          RPOK   = 0043          RPOSTN= 0016
RPSTA  = 0015          RRCMT  = 000A          RSTAT  = 0002          RTIEB  = 000A
RTIER  = 0030          RTM    = 0005          RUNKI  = 0009          RUPTST= 005E
RWDUNL= 0005          R.AMT  = 0001          R.BOP  = 0008          R.DATA= 0040
R.DON  = 0002          R.DRDY= 0010          R.END  = 0010          R.ILL  = 0004
R.JVOK= 0004          R.MK2  = 0008          R.PLOD= 0008          R.PLOO= 0010
R.PLO1= 0020          R.POST= 0020          R.STNM= 0002          R.STOP= 0004
R.STPC= 0001          R.TBJN= 0080          R.TSTD= 0040          R.VOK  = 0080
R00H   = 0081          R00L   = 0080          R01H   = 0083          R01L   = 0082
R02H   = 0085          R02L   = 0084          R03H   = 0087          R03L   = 0086
R04H   = 0089          R04L   = 0088          R05H   = 008B          R05L   = 008A
```

```
R06H  = 008D            R06L  = 008C            R07H  = 008F            R07L  = 008E
R10H  = 0091            R10L  = 0090            R11H  = 0093            R11L  = 0092
R12H  = 0095            R12L  = 0094            R13H  = 0097            R13L  = 0096
R14H  = 0099            R14L  = 0098            R15H  = 009B            R15L  = 009A
R16H  = 009D            R16L  = 009C            R17H  = 009F            R17L  = 009E
R7.S  = 0010            SDDRA   45C6            SDDRAT  45C9            SDDRB   45C7
SDDRBT  45CA            SDDRC   45C8            SDDRCT  45CB            SELCLR= 0000
SETATA= 00A1            SETDDR  4552            SID   = 0080            SOD   = 0080
SOE   = 0040            SP    =%0008            SSCLK = 0040            SSTEP = 0005
STACK = 4FFF            STATRM= 4F20            STPCT   4F20            STRSP = 5000
TADR00= 0080            TADR01= 0081            TADR02= 0082            TADR03= 0083
TADR04= 0084            TADR05= 0085            TADR06= 0086            TADR07= 0087
TADR10= 0088            TADR11= 0089            TADR12= 008A            TADR13= 008B
TAMT  = 0044            TASEL = 0080            TCMD  = 0040            TC.FWD= 0040
TC.INH= 0008            TC.LWR= 0004            TC.REV= 0020            TC.WRT= 0010
TEMP    4F99            TEST1   4300            TMF   = 0099            TMRDY = 00[illegible]
TPBCNT  45CD            TRKENA= 00D2            TSET  = 2803            TSTEND= 2818
TSTS  = 0040            TST01X  4305            TST02C  43C7            TST2    4357
TST2L   43BB            TUSEL0= 00D1            TUSEL1= 00D2            TU78  = 0010
T.ATH0= 0001            T.ATH1= 0002            T.BOT = 0004            T.EOT = 0002
T.FPT = 0001            T.NTHR= 0004            T.ONL = 0020            T.PES = 0008
T.PSBJ= 0020            T.PSOJ= 0008            T.PS1J= 0010            T.RDY = 0080
T.RDYO= 0040            T.RWD = 0010            T.SCLK= 0002            T01CN1  432B  G
T01CN2  433D  G         T2CN3   450E  G         T2CN4   4520  G         T2LOP1  448A
T2LOP2  44DD            T2LOP3  4502            T2LOP4  448C            T2LPA   4370
T2LPB   436C            UIBG  = 00A1            UNITMP  45CF            VALFC   4F98
VALTB   4F95            VELTST= 005B            WDR.P = 0010            WMCCTL= 00D3
WMCERR= 00DA            WMCSTA= 00D0            WRTCLK= 0000            WRTDAT= 00D3
W.ACRC= 0004            W.CRC = 0008            W.DIAG= 0002            W.DONN= 0040
W.ECC = 0010            W.ENAB= 0080            W.ERR = 0020            W.FMT = 0070
W.GCR = 0010            W.LEFT= 0004            W.ONES= 0020            W.RESI= 0002
W.REV = 0004            W.ROME= 0010            W.RST = 0001            W.SKIP= 000F
W.WRIT= 0008            W.XFER= 0020            X     =%000A            XLCLK   45D0
X.DONN= 0080            X.ENAB= 0040            X.PEPE= 0002            X.ROME= 0001
X.WCLK= 0001            Y     =%000B            .     = 45D3


ERRORS DETECTED:  0

*CCM1.A78/PTP,CCM1=NLIST,PARAM,MACRO,LIST,CCM1
RUN-TIME: 4 4 0 SECONDS
CORE USED:  10K
```

3 'MEMORY MAP'' EXTENDED ADDRESS BIT
19 READ MICRO CONTROLLER CONTROL & STATUS WORDS
139 TRANSPORT/PORT COMMAND + STATUS WORDS
380 COMMON ADDRESS SPACE REGISTERS
444 MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485 MASSBUS CONTROL WORD DEFINITIONS
557 WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709 KEYBOARD/DISPLAY CONTROL + STATUS WORD
797 MISCELLANEOUS VARIABLE DEFINITIONS
801 RAM ADDRESS DEFINITIONS
819 LINKAGE TABLE ADDRESSES
842 BIT DEFINITIONS
856 RIM AND SIM INSTRUCTION BIT DEFINITIONS
877 INTERRUPTABLE INSTRUCTION MACROS
924 DIAGNOSTIC VARIABLES
945 COMMON VARIABLE STORAGE
959 MACRO CALL DEFINITIONS
1047 TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101 8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1333 TEST 1 - WMC BYTE ASSEMBLY - READ FORWARD
1772 TEST 2 - WMC BYTE ASSEMBLY - READ REVERSE
2613 SUBROUTINE CLEAR ALL TU PORTS
2658 SUBROUTINE 4X5 TRANSLATE A SUBGROUP
2770 TABLE 4 X 5 TRANSLATION
2810 SUBROUTINE VARIABLES
2833 SUBROUTINE CLEAR ECC
2844 SUBROUTINE CALCUALTE ECC CHARACTER
2882 SUBROUTINE POLYNOMIAL BIT TRANSLATION

```
                ;*************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;                "BYTE/SCLK COUNT NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                MM = DATA FORMAT FROM CAS REGISTER 2
                ;                NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;                BYTE-SCLK COUNT = LLL
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                LLL = AS ABOVE
                ;                MM  = AS ABOVE
                ;                NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;                TRANSITION COUNT = LLL
                ;                WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;                EXPECTED 18 BITS =      E EEEEEE
                ;                ACTUAL 18 BITS =        A AAAAAA
                ;
                ;                WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;                BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;                OF CAS REG 15 LOW BYTE.
                ;
                ;                ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;                AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;                17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;                TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;                AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;                "SUBGROUP NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;*************************************************************************
```

```
1016                    ;*****************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ;*****************************************************************
1035                    ;*****************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ;*****************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  WMC6 - BYTE ASSEMBLY OF READ PATH DATA
                                .SBTTL  TEST 1 - WMC BYTE ASSEMBLY - READ FORWARD
                                ;ID     WMC6-WRITE MICRO CONTROLLER PART #6
4300                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*WMC BYTE ASSEMBLY FROM READ FORWARD DATA
4300                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST WILL INJECT DATA THROUGH THE READ PATH FIFO'S AND CLOCK THE
                                ;*SYSTEM UNTIL AN 'SCLK' APPEARS TO SIGNAL 18 BITS OF DATA READY FOR
                                ;*TRANSMISSION ACROSS THE MASSBUS. EACH 'SCLK' TIME WILL HAVE THE 'DDR'
                                ;*REGISTERS SETUP WITH THE ASSEMBLED DATA AS COMMANDED BY THE PACKING
                                ;*MODE SELECTED IN THE TEST. ALL PACKING MODES ARE TESTED (I.E. 11-NORM
                                ;*TO 10-HI DEN COMPAT) IN A READ FORWARD DATA TRANSFER. THE TEST WILL
                                ;*CLOCK THE SYSTEM ONLY UNTIL THE EXPECTED AMOUNT OF 'SCLK' DATA HAS
                                ;*BEEN TESTED TO VERIFY THAT THE SELECTED PACKING MODE IS WORKING AS
                                ;*EXPECTED. THE READ PATH WILL BE FILLED WITH 8 SUBGROUPS WORTH OF DATA
                                ;*REGUARDLESS OF THE AMOUNT OF EXPECTED DATA GROUPS TO BE TESTED (USUALLY
                                ;*4 SUBGROUPS OF DATA WILL BE SUFFICIENT TO TEST A PACKING MODE). THE EXTRA
                                ;*SUBGROUPS OF DATA INJECTED WILL ELIMINATE THE NECESSITY OF CALCULATING
                                ;*THE 'SPECIAL' CHARACTERS FROM THE TAPE IN GCR MODE (I.E. CRC, ACRC, &
                                ;*RESID CHARACTERS). EVERY 4 BYTES OF DATA TO BE INJECTED WILL BE TURNED
                                ;*INTO A 5 BYTE SUBGROUP TO SIMULATE DATA FROM THE 'TAPE'. THE 'ECC'
                                ;*WILL BE CALCULATED FOR EACH 7 BYTE OF DATA INJECTED AND ALSO INJECTED
                                ;*AT THE PROPER TIME INTO THE FIFO'S.
4300                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  REQUEST A TU PORT FROM 'HOST' CPU
                                ;*  IF NO TU PORT SELECTED FROM 'HOST' CPU
                                ;*  :  THEN-REPORT OPERATOR ERROR
                                ;*  :  ELSE-PROCEED
                                ;*  ENDIF
                                ;*  BGNDO
                                ;*  :  CLEAR ALL PORTS
                                ;*  :  INIT THE SYSTEM
                                ;*  :  SET READ PATH TO NORMAL CLOCK SPEED
                                ;*  :  INIT COUNTERS + CONTROLS
                                ;*  :  INIT PACKING MODE TO 'IMAGE'
                                ;*  :  LOAD BYTE COUNTER + PAD COUNTER
                                ;*  :  RESTART WRITE MICROCONTROLLER
                                ;*  :  CLOCK WMC THROUGH RESTART CODE
                                ;*  :  START 'GCR' READ DATA XFR
                                ;*  :  CALCULATE ALL NECESSARY ECC CHARACTERS FOR INJECTION
                                ;*  :  SET READ PATH PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
                                ;*  :  CLOCK THE FIFO'S
                                ;*  :  ISSUE 'CLEAR ALL' TO READ PATH
                                ;*  :  WAIT
```

```
                                ;*  :  STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
                                ;*  :  CONVERT SUBGROUPS TO 4X5 & LOAD INTO FIFO
                                ;*  :  SETUP A WATCHDOG TIMER
                                ;*  :  CLOCK SYSTEM TO XFR DATA FROM FIFO TO WMC BYTE ASSY LOGIC
                                ;*  :  IF WATCHDOG TIMER=0
                                ;*  :  :  THEN-REPORT NO 'SCLK'
                                ;*  :  :  ELSE-CONT
                                ;*  :  ENDIF
                                ;*  :  CLOCK SYSTEM TILL SCLK=0...DATA READY TO BE TESTED
                                ;*  :  GET 'SCLK' 18 BIT DATA FROM BYTE ASSY INTO 'DDR' REGISTER
                                ;*  :  IF 18 BIT DATA SAME AS EXPECTED
                                ;*  :  :  THEN-PROCEED
                                ;*  :  :  ELSE-REPORT BYTE ASSY ERROR + WHAT WAS ASSEMBLED
                                ;*  :  ENDIF
                                ;*  :  POINT TO NEXT SET OF EXPECTED DATA
                                ;*  :  IF ALL DONE WITH CURRENT XFR
                                ;*  :  :  THEN-UPDATE TO NEXT PACKING MODE
                                ;*  :  :     IF ALL MODES DONE
                                ;*  :  :     :  THEN-EXIT
                                ;*  :  :     :  ELSE-CONTINUE
                                ;*  :  :     ENDIF
                                ;*  :  :  ELSE-CONTINUE
                                ;*  :  ENDIF
                                ;*  :  DO UNTIL ALL 'SCLK' CHECKED FOR SELECTED XFR
                                ;*  ENDDO
                                ;*ENDTST
1403  4300                      SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1404                            ;*WMC6 MICRO TEST 01
1405                            ;*WMC6 MICRO ERROR 01
1406                            ;*WMC BYTE ASSY FOR READ FORWARD XFR
1407                            ;*M8959 (WMC) & ALL OTHER BOARDS
1408                            ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
1409                            ;*FATAL ERROR - TEST ABORTED
1410                            ;*
1411                            ;*WMC6 MICRO TEST 01
1412                            ;*WMC6 MICRO ERROR 02
1413                            ;*WMC BYTE ASSY FOR READ FORWARD XFR
1414                            ;*M8959 (WMC) & ALL OTHER BOARDS
1415                            ;*TIMEOUT WHILE WAITING FOR 'SCLK TO SET IN DBUSSTA
1416                            ;*AFTER ISSUEING A READ FORWARD COMMAND TO THE TM78
1417                            ;*
1418                            ;*WMC6 MICRO TEST 01
1419                            ;*WMC6 MICRO ERROR 03
1420                            ;*WMC BYTE ASSY FOR READ FORWARD XFR
1421                            ;*M8959 (WMC) & ALL OTHER BOARDS
1422                            ;*TIMEOUT WHILE WAITING FOR 'SCLK' TO CLEAR AFTER SETTING
1423                            ;*DURING A READ FORWARD COMMAND
1424                            ;*FATAL ERROR - TEST ABORTED
1425                            ;*
1426                            ;*WMC6 MICRO TEST 01
1427                            ;*WMC6 MICRO ERROR 04
```

```
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 7-0
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*WMC6 MICRO TEST 01
                                        ;*WMC6 MICRO ERROR 05
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 15-8
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*WMC6 MICRO TEST 01
                                        ;*WMC6 MICRO ERROR 06
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 17-16
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
1449  4300                              $
 (1)                                    ; ************************************************************************************

1450
1451  4300                              TEST1:  TESTX   @1
 (1)  4300  3E  01              7.0                     MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28         18.0                     CALL    TSET            ;SETUP THE TEST
1452                                    ;%WMC BYTE ASSY FOR READ FORWARD XFR
1453                                    ;&M8959 (WMC) & ALL OTHER BOARDS
1454                                    ;REQUEST THE 'HOST' CPU TO SELECT A PORT FOR TESTING
1455
1456  4305                                      REQ     7
 (1)  4305  CD  06  28         18.0                      CALL    REQST
 (1)  4308  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (1)  4309  00  00                                       .WORD                   ;SYSTEM '"' COUNT
 (1)  430B  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (1)  430E  07                                           .BYTE   7               ;REQUEST CODE
1457  430F                                RIN     R12L                    ;GET THE SELECTED PORT FROM 'HOST'
 (1)  430F  DB  94             10.0                      IN      R12L            ;READ R12L INTO AC
 (1)  4311  7F                  4.0                      MOV     A,A             ;RETRY LINK
1458  4312  32  2D  4A          13.0              STA     UNITMP                  ;SAVE IT
1459  4315  A7                   4.0              ANA     A                       ;SET THE CONDITION BITS
1460  4316  C2  27  43          10.0              JNZ     FOUND                   ;SEE IF ANY WERE SELECTED IN 'HOST'
1461  4319                              ERR     EXIT,NOUNIT
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28         18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                        MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                          .BYTE
 (1)  431E  CD  15  28         18.0              NOUNIT::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  6F  45         10.0                      JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1462                                    ;>OPERATOR ERROR - NO TU78 UNIT SELECTED
1463                                    ;>FATAL ERROR - TEST ABORTED
```

```
1464                                        ;*REQUEST THE 'HOST' CPU TO SELECT A PORT FOR TESTING
1465  4324  C3  6F  45      10.0            JMP     EXIT
1466                                        ;HERE IF THE HOST CPU INDICATED A PORT TO USE
1467
1468  4327  CD  7C  49      18.0  FOUND:    CALL    CLEAR           ;INIT ALL THE PORTS IN THE PORT BOARD
1469  432A  3A  2D  4A      13.0            LDA     UNITMP          ;GET THE RESPONSE FROM 'HOST' CPU
1470  432D  06  00           7.0            MVI     B,0             ;INIT 'B'
1471  432F  E6  01           7.0            ANI     1               ;IS UNIT 0 SELECTED?
1472  4331  C2  47  43      10.0            JNZ     FPORT           ;YUP - FOUND THE PORT TO USE
1473  4334  04               4.0            INR     B               ;NO - COUNT THE PORT
1474  4335  3A  2D  4A      13.0            LDA     UNITMP          ;GET RESPONSE AGAIN
1475  4338  E6  02           7.0            ANI     2               ;PORT 1 SELECTED?
1476  433A  C2  47  43      10.0            JNZ     FPORT           ;YUP
1477  433D  04               4.0            INR     B               ;COUNT THE PORT
1478  433E  3A  2D  4A      13.0            LDA     UNITMP          ;GET RESPONSE
1479  4341  E6  04           7.0            ANI     4               ;PORT 2 SELECTED?
1480  4343  C2  47  43      10.0            JNZ     FPORT           ;YUP
1481  4346  04               4.0            INR     B               ;NO - ASSUME PORT 3
1482
1483  4347  78               4.0  FPORT:    MOV     A,B             ;PUT UNIT # IN AC
1484  4348  32  2E  4A      13.0            STA     SUNIT           ;SAVE THE UNIT UNDER TEST
1485  434B  DB  E0          10.0            IN      INTSTA          ;GET THE PORT SELECT STATUS
1486  434D  E6  80           7.0            ANI     BIT7
1487  434F  B0               4.0            ORA     B               ;ADD IN THE SELECTED PORT #
1488  4350  D3  E0          10.0            OUT     MBSEL           ;SELECT THE DESIRED PORT TO USE
1489  4352  3E  60           7.0            MVI     A,P.LWR!P.LCS   ;GET THE LOOP-COMMAND-STATUS BIT
1490  4354  D3  48          10.0            OUT     PDIAG           ;SELECT IT
1491  4356  AF               4.0            XRA     A
1492  4357  D3  44          10.0            OUT     TAMT            ;CLEAR THE AMTIE LINES
1493  4359  D3  D2          10.0            OUT     TRKENA          ;CLEAR THE TRACK ENABLE BITS
1494  435B  D3  D2          10.0            OUT     TRKENA
1495  435D  3E  05           7.0            MVI     A,5
1496  435F  32  2C  4A      13.0            STA     FMTNUM          ;INIT THE PACKING MODE TO IMAGE MODE
1497  4362  3E  04           7.0            MVI     A,P.RPEN        ;SET READ PATH ENABLE
1498  4364  D3  4C          10.0            OUT     PENAB           ;NOW
1499  4366  3E  10           7.0            MVI     A,RDCLK         ;SETUP THE READ AND WRITE CLOCK TO 'NORMAL'
1500  4368  D3  F0          10.0            OUT     CLKCTL
1501  436A  AF               4.0            XRA     A               ;SETUP TO READ FWD 1ST
1502  436B  32  2F  4A      13.0            STA     REVERSE         ;=0 FOR READ FWD, =-1 FOR READ REV
1503  436E  D3  C0          10.0            OUT     DBUSCTL         ;CLEAR THE DATA BUS CONTROL WORD
1504                                    ;INIT THE COUNTER CONTROL AND COUNTERS
1505
1506  4370                            T1LOOP: INICNT
 (1)  4370  3E  30           7.0                    MVI     A,@060          ;INIT THE
 (1)  4372  D3  D7          10.0                    OUT     CNTCTL          ;BYTE COUNTER
 (1)  4374  3E  70           7.0                    MVI     A,@160          ;INIT THE
 (1)  4376  D3  D7          10.0                    OUT     CNTCTL          ;PAD COUNTER
 (1)  4378  3E  B0           7.0                    MVI     A,@260          ;INIT THE
 (1)  437A  D3  D7          10.0                    OUT     CNTCTL          ;ERROR COUNTER
1507  437C                                CLRECT
 (1)  437C  AF               4.0                    XRA     A               ;CLEAR THE ACCUMULATOR
 (1)  437D  D3  D6          10.0                    OUT     ERRCNT          ;CLEAR BITS 7-0
 (1)  437F  D3  D6          10.0                    OUT     ERRCNT          ;CLEAR BITS 15-8
1508  4381  3A  2C  4A      13.0            LDA     FMTNUM                  ;GET THE SELECTED PACKING MODE #
```

```
1509  4384  87              4.0          ADD   A              ;DOUBLE THE COUNT - 2 ADDR PER ENTRY IN TABLE
1510  4385  5F              4.0          MOV   E,A
1511  4386  16  00          7.0          MVI   D,0
1512  4388  21  AF  48      10.0  T1RF:  LXI   H,EDFWD        ;POINT TO READ FORWARD
1513  438B  19              10.0         DAD   D              ;GET OFFSET TO EXPECTED DATA
1514  438C  7E              7.0          MOV   A,M            ;GET LOW BYTE OF ADDRESS
1515  438D  32  D2  48      13.0         STA   EXPADR         ;SAVE IT
1516  4390  23              6.0          INX   H              ;POINT TO THE HIGH BYTE
1517  4391  7E              7.0          MOV   A,M            ;GET IT
1518  4392  32  D3  48      13.0         STA   EXPADR+1       ;SAVE HIGH BYTE OF EXPECTED DATA ADDRESS
1519  4395  3A  2C  4A      13.0         LDA   FMTNUM         ;GET THE PACKING MODE # AGAIN
1520  4398  5F              4.0          MOV   E,A
1521  4399  16  00          7.0          MVI   D,0
1522  439B  21  CB  48      10.0         LXI   H,ESCLKT       ;POINT TO TABLE OF EXPECTED SCLKS/MODE
1523  439E  19              10.0         DAD   D              ;POINT TO TABLE OFFSET #
1524  439F  7E              7.0          MOV   A,M            ;GET THE EXPECTED # SCLKS/MODE SELECTED
1525  43A0  32  2B  4A      13.0         STA   ESCLK          ;SAVE THE NUMBER FOR THIS PASS
1526  43A3  3A  2C  4A      13.0         LDA   FMTNUM         ;GET THE PACKING MODE # AGAIN
1527  43A6  07              4.0          RLC
1528  43A7  07              4.0          RLC
1529  43A8  07              4.0          RLC
1530  43A9  07              4.0          RLC
1531  43AA  E6  70          7.0          ANI   $70            ;POSITION THE FMT #
1532  43AC  D3  D0          10.0         OUT   DATACTL        ;LOAD THE DATA CONTROL WORD
1533  43AE                               ROUT  R02H           ;SAVE FOR ERROR PRINTING
 (1)  43AE  D3  85          10.0                OUT   R02H            ;WRITE AC INTO R02H
 (1)  43B0  7F              4.0                 MOV   A,A             ;RETRY LINK
1534  43B1  3E  9B          7.0          MVI   A,@233         ;SET THE DDR CONTROL TO 'IN'
1535  43B3  D3  DB          10.0         OUT   DDRCTL         ;TO READ THE 'DDR'
1536  43B5  3E  00          7.0          MVI   A,@0           ;GET THE RESIDUAL CHAR WORD
1537  43B7  D3  D1          10.0         OUT   RESCHR         ;LOAD IT
1538  43B9  3E  32          7.0          MVI   A,50           ;SETUP THE BYTE COUNTER FOR 8 DATA GROUPS
1539  43BB  D3  D4          10.0         OUT   BYTCNT
1540  43BD  AF              4.0          XRA   A
1541  43BE  D3  D4          10.0         OUT   BYTCNT
1542  43C0  3E  06          7.0          MVI   A,6            ;NEED 6 PAD CHARACTERS FOR THE XFR
1543  43C2  D3  D5          10.0         OUT   PADCNT         ;LOAD IT
1544  43C4  AF              4.0          XRA   A
1545  43C5  D3  D5          10.0         OUT   PADCNT
1546  43C7                               ROUT  R02L           ;CLEAR OUT THE REST OF REG 2 IN CAS
 (1)  43C7  D3  84          10.0                OUT   R02L            ;WRITE AC INTO R02L
 (1)  43C9  7F              4.0                 MOV   A,A             ;RETRY LINK
1547  43CA                               ROUT  R05L           ;CLEAR THE BYTE/SCLK COUNTER FOR REPORTS
 (1)  43CA  D3  8A          10.0                OUT   R05L            ;WRITE AC INTO R05L
 (1)  43CC  7F              4.0                 MOV   A,A             ;RETRY LINK
1548  43CD  32  2A  4A      13.0         STA   SCLKCT         ;INIT THE SCLK DETECTED COUNTER
1549
1550                         ;RESTART THE WMC AND SET WMC TO SINGLE STEP MODE
1551
1552  43D0  3E  01          7.0          MVI   A,W.RST        ;GET THE WMC RESTART BIT
1553  43D2  D3  D3          10.0         OUT   WMCCTL         ;SET RESTART
1554  43D4  3E  15          7.0          MVI   A,SSTEP!RDCLK  ;GET SINGLE STEP FOR WMC
1555  43D6  D3  F0          10.0         OUT   CLKCTL         ;SET SINGLE STEP MODE
155.  43D8  32  30  4A      13.0         STA   CCTLWD         ;SAVE THE WORD
```

```
1557  43DB  CD  81  4A   18.0          CALL   CLKSYS          ;CLOCK THE SYSTEM TO CAUSE RESTART TO
1558  43DE  CD  81  4A   18.0          CALL   CLKSYS          ;TO TAKE HOLD
1559  43E1  AF             4.0          XRA    A
1560  43E2  D3  D3        10.0          OUT    WMCCTL          ;FINISH THE RESTART CYCLE
1561
1562                                ;CLOCK THE WMC THRU ITS RESTART AREA
1563
1564  43E4  06  B4         7.0          MVI    B,180           ;GET A LOOP COUNT
1565  43E6  CD  81  4A   18.0  CLKRS:  CALL   CLKSYS          ;CLOCK THE SYSTEM
1566  43E9  05             4.0          DCR    B               ;DECREMENT THE LOOP COUNTER
1567  43EA  C2  E6  43   10.0          JNZ    CLKRS           ;CLOCK TILL COUNT RUN OUT
1568
1569  43ED  3E  90         7.0          MVI    A,W.GCR.W.ENAB  ;SETUP FOR A GCR DATA XFR
1570  43EF  D3  D3        10.0  T1ST:   OUT    WMCCTL          ;LOAD THE WORD IN WMC
1571
1572                                ;CALCULATE THE 1ST DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
1573
1574  43F1  21  8F  48    10.0  C1EC1:  LXI    H,DG1           ;POINT THE THE 1ST DATA GROUP DATA TABLE
1575  43F4  06  07         7.0          MVI    B,7             ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
1576  43F6  CD  42  4A   18.0          CALL   CLECC           ;INIT THE ECC CHARACTER
1577
1578  43F9  7E             7.0  C1EC1L: MOV    A,M             ;GET A DATA BYTE FROM THE TABLE
1579  43FA  CD  47  4A   18.0          CALL   ECC             ;CALC ECC WITH THIS CHARACTER
1580  43FD  23             6.0          INX    H               ;POINT TO THE NEXT DATA BYTE IN TABLE
1581  43FE  05             4.0          DCR    B               ;DECREMENT THE COUNTER OF DATA BYTES
1582  43FF  C2  F9  43    10.0          JNZ    C1EC1L          ;STAY HERE TILL ALL DATA BYTES USED
1583  4402  3A  6E  4A    13.0          LDA    ECCCHR          ;GET THE CALCULATED ECC CHARACTER
1584  4405  32  96  48    13.0          STA    DG1ECC          ;SAVE THE CALCULATED ECC CHARACTER
1585
1586                                ;CALCULATE THE 2ND DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
1587
1588  4408  21  97  48    10.0  C1EC2:  LXI    H,DG2           ;POINT THE THE 2ND DATA GROUP DATA TABLE
1589  440B  06  07         7.0          MVI    B,7             ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
1590  440D  CD  42  4A   18.0          CALL   CLECC           ;INIT THE ECC CHARACTER
1591
1592  4410  7E             7.0  C1EC2L: MOV    A,M             ;GET A DATA BYTE FROM THE TABLE
1593  4411  CD  47  4A   18.0          CALL   ECC             ;CALC ECC WITH THIS CHARACTER
1594  4414  23             6.0          INX    H               ;POINT TO THE NEXT DATA BYTE IN TABLE
1595  4415  05             4.0          DCR    B               ;DECREMENT THE COUNTER OF DATA BYTES
1596  4416  C2  10  44    10.0          JNZ    C1EC2L          ;STAY HERE TILL ALL DATA BYTES USED
1597  4419  3A  6E  4A    13.0          LDA    ECCCHR          ;GET THE CALCULATED ECC CHARACTER
1598  441C  32  9E  48    13.0          STA    DG2ECC          ;SAVE THE CALCULATED ECC CHARACTER
1599  441F  3E  A8         7.0  T1PLO:  MVI    A,R.PLO1!R.TBJN!R.PLOD  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1600                                                                  ;SET TIE BUS JAM AND PLO DISABLE
1601  4421  D3  09        10.0          OUT    RPCTL
1602  4423  AF             4.0          XRA    A
1603  4424  D3  0A        10.0          OUT    RTIEB           ;CLEAR THE TIE BUS
1604  4426  D3  08        10.0          OUT    RFIFOL          ;CLOCK THE FIFO'S
1605  4428  3E  0D         7.0          MVI    A,RCLRT         ;ISSUE CLEAR ALL COMMAND
1606  442A  D3  0B        10.0          OUT    RCMD
1607
1608  442C  00             4.0          NOP                    ;WAIT
1609  442D  00             4.0          NOP                    ;WAIT
1610  442E  00             4.0          NOP
```

```
1611  442F  00          4.0          NOP
1612  4430  00          4.0          NOP
1613
1614  4431  3E  A9      7.0          MVI     A,R.PLO1.R.STPC.R.PLOD.R.TBJN   ;STOP THE READ PATH
1615                                                                         ;SET TIE BUS JAM AND PLO DISABLE
1616  4433  D3  09     10.0          OUT     RPCTL
1617
1618  4435  0E  03      7.0          MVI     C,@3            ;INIT THE LOOP COUNT FOR 6 DATA GROUPS
1619
1620  4437  21  8F 48  10.0  T1SG1:  LXI     H,DG1           ;POINT TO THE 1ST DATA GROUP FOR TRANSLATION
1621  443A  CD  43 48  18.0          CALL    CONVRT          ;CONVERT SUBGROUP - 4X5
1622  443D  CD  58 48  18.0          CALL    LOADIT          ;INJECT THE DATA INTO THE FIFO
1623
1624  4440  21  93 48  10.0  T1SG2:  LXI     H,SG2           ;POINT TO THE 2ND SUBGROUP OF DATA
1625  4443  CD  43 48  18.0          CALL    CONVRT          ;MAKE IT 4X5
1626  4446  CD  58 48  18.0          CALL    LOADIT          ;INJECT IT INTO FIFO
1627
1628  4449  21  97 48  10.0  T1SG3:  LXI     H,DG2           ;POINT TO THE 2ND DATA GROUP - SUBGROUP 3
1629  444C  CD  43 48  18.0          CALL    CONVRT          ;MAKE IT 4X5
1630  444F  CD  58 48  18.0          CALL    LOADIT          ;INJECT THE DATA INTO FIFO
1631
1632  4452  21  9B 48  10.0  T1SG4:  LXI     H,SG4           ;POINT TO THE 4TH SUBGROUP OF DATA
1633  4455  CD  43 48  18.0          CALL    CONVRT          ;MAKE IT 4X5
1634  4458  CD  58 48  18.0          CALL    LOADIT          ;INJECT THE DATA INTO FIFO
1635
1636  445B  0D          4.0          DCR     C               ;DECREMENT THE LOOP COUNTER
1637  445C  C2  37 44  10.0          JNZ     T1SG1           ;KEEP INJECTING DATA TILL 6 GROUPS IN FIFO
1638
1639  445F  3E  0B      7.0          MVI     A,DIARD         ;LOAD THE DIAGNOSTIC READ
1640  4461  D3  0B     10.0          OUT     RCMD            ;COMMAND
1641  4463  11  01 00  10.0  T1CLKA: LXI     D,1             ;SET WATCH DOG INCREMENT
1642  4466  21  68 C5  10.0          LXI     H,-15000        ;SET WATCH DOG COUNT TO 15000
1643  4469  D3  0C     10.0  T1CLK:  OUT     RINST           ;STEP THE READ PATH
1644  446B  CD  81 4A  18.0          CALL    CLKSYS          ;CLOCK THE WMC
1645  446E  DB  C0     10.0          IN      DBUSSTA         ;GET THE DATABUS STATUS
1646  4470  E6  02      7.0          ANI     T.SCLK          ;SEE IF SCLK UP - DATA READY
1647  4472  C2  87 44  10.0          JNZ     T1SCLK          ;JUMP IF SCLK UP
1648  4475  19         10.0          DAD     D
1649  4476  D2  69 44  10.0          JNC     T1CLK           ;STAY IN LOOP UNLESS TIMEOUT
1650
```

```
1652  4479                                          ERR     T1LOOP,DUMM1,3
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4479  CD   09   28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  447C  02                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  447D  03                                          .BYTE   3
 (1)  447E  CD   15   28        18.0               DUMM1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4481  DA   70   43        10.0                       JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1653                                    ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO SET IN DBUSSTA
1654                                    ;>AFTER ISSUEING A READ FORWARD COMMAND TO THE TM78
1655                                    ;<FATAL ERROR - TEST ABORTED
1656  4484  C3   6F   45        10.0            JMP     EXIT
1657
1658                                    ;HERE TO CLOCK THE WMC SO THE 'SCLK' SIGNAL WILL GO =0
1659                                    ;THIS IS NECESSARY BEFORE STROBING THE DDR REGISTERS...THEY ARE ONLY
1660                                    ;VALID AFTER 'SCLK' HAS TRANSITIONED TO ZERO INDICATING THE DATA IS
1661                                    ;ON THE MASSBUS
1662
1663  4487  06   FF              7.0  T1SCLK: MVI     B,$FF           ;SETUP A DROP DEAD COUNT
1664  4489  D3   0C             10.0  T1SCKL: OUT     RINST           ;CLOCK THE READ PATH ONCE
1665  448B  CD   81   4A        18.0          CALL    CLKSYS          ;CLOCK THE WMC TO CLEAR 'SCLK'
1666  448E  DB   C0             10.0          IN      DBUSSTA         ;SEE IF 'SCLK' IS GONE
1667  4490  E6   02              7.0          ANI     T.SCLK          ;SAVE ONLY THE 'SCLK' BIT IN STATUS WORD
1668  4492  CA   A7   44        10.0          JZ      T1SCK0          ;JUMP IF 'SCLK' = 0
1669  4495  05                   4.0          DCR     B               ;DECREMENT THE COUNTER
1670  4496  C2   89   44        10.0          JNZ     T1SCKL          ;LOOP BACK TILL COUNTER =0
1671  4499                                    ERR     T1LOOP,DUMM2,3
 (1)                                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4499  CD   09   28        18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0003                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  449C  03                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  449D  03                                          .BYTE   3
 (1)  449E  CD   15   28        18.0               DUMM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44A1  DA   70   43        10.0                       JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1672                                    ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO CLEAR AFTER SETTING
1673                                    ;>DURING A READ FORWARD COMMAND
1674                                    ;<FATAL ERROR - TEST ABORTED
1675  44A4  C3   6F   45        10.0            JMP     EXIT
1676                                    ;HERE WHEN 'SCLK' WAS SET - 18 BITS OF DATA HAS BEEN ASSEMBLED FROM
1677                                    ;THE BYTE ASSEMBLY LOGIC. DATA CAME FROM THE READ PATH (FIFO).
1678
1679  44A7  DB   D8             10.0  T1SCK0: IN      DDRA            ;GET THE DDR A DATA
1680  44A9  32   27   4A        13.0          STA     SDDRA           ;SAVE IT
1681  44AC                                    ROUT    R16L            ;SAVE FOR PRINTING
 (1)  44AC  D3   9C             10.0                       OUT     R16L            ;WRITE AC INTO R16L
 (1)  44AE  7F                   4.0                       MOV     A,A             ;RETRY LINK
1682  44AF  DB   D9             10.0          IN      DDRB            ;GET DDR B DATA
1683  44B1  32   28   4A        13.0          STA     SDDRB           ;SAVE IT
1684  44B4                                    ROUT    R16H            ;SAVE FOR ERROR REPORTING
 (1)  44B4  D3   9D             10.0                       OUT     R16H            ;WRITE AC INTO R16H
 (1)  44B6  7F                   4.0                       MOV     A,A             ;RETRY LINK
1685  44B7  DB   DA             10.0          IN      DDRC            ;GET DDR C DATA
1686  44B9  E6   03              7.0          ANI     3
1687  44BB  32   29   4A        13.0          STA     SDDRC           ;SAVE IT
```

```
1688  44BE  F6  80        7.0          ORI   BIT7          ;ADD IN THE PRINT FLAG
1689  44C0                             ROUT  R17L          ;SAVE IT FOR PRINTING
 (1)  44C0  D3  9E       10.0                OUT   R17L           ;WRITE AC INTO R17L
 (1)  44C2  7F            4.0                MOV   A,A            ;RETRY LINK
1690
1691  44C3  3A  2A  4A    13.0          LDA   SCLKCT        ;GET THE CURRENT SCLK COUNT
1692  44C6  3C            4.0          INR   A             ;COUNT THIS SCLK
1693  44C7  32  2A  4A    13.0          STA   SCLKCT        ;SAVE THE NEW COUNT
1694  44CA                             ROUT  R05L          ;ALSO IN CAS FOR ERROR REPORTING
 (1)  44CA  D3  8A       10.0                OUT   R05L           ;WRITE AC INTO R05L
 (1)  44CC  7F            4.0                MOV   A,A            ;RETRY LINK
1695  44CD  AF            4.0          XRA   A
1696  44CE                             ROUT  R05H
 (1)  44CE  D3  8B       10.0                OUT   R05H           ;WRITE AC INTO R05H
 (1)  44D0  7F            4.0                MOV   A,A            ;RETRY LINK
1697
1698                                 ;CHECK THE DATA IN DDR A - BITS 7-0 OF THE 18 BITS ASSEMBLED
1699
1700  44D1  2A  D2  48    16.0  T1CKA:  LHLD  EXPADR           ;POINT TO THE TABLE OF EXPECTED DATA
1701  44D4  7E            7.0          MOV   A,M             ;GET BITS 7-0
1702  44D5                             ROUT  R14L            ;SAVE FOR ERROR REPORTING
 (1)  44D5  D3  98       10.0                OUT   R14L           ;WRITE AC INTO R14L
 (1)  44D7  7F            4.0                MOV   A,A            ;RETRY LINK
1703  44D8  23            6.0          INX   H               ;POINT TO NEXT BYTE
1704  44D9  7E            7.0          MOV   A,M             ;GET BITS 15-8
1705  44DA                             ROUT  R14H            ;SAVE IT
 (1)  44DA  D3  99       10.0                OUT   R14H           ;WRITE AC INTO R14H
 (1)  44DC  7F            4.0                MOV   A,A            ;RETRY LINK
1706  44DD  23            6.0          INX   H
1707  44DE  7E            7.0          MOV   A,M             ;GET BITS 17-16
1708  44DF  F6  80        7.0          ORI   BIT7            ;ADD IN THE PRINT FLAG
1709  44E1                             ROUT  R15L            ;SAVE FOR ERROR REPORT
 (1)  44E1  D3  9A       10.0                OUT   R15L           ;WRITE AC INTO R15L
 (1)  44E3  7F            4.0                MOV   A,A            ;RETRY LINK
1710  44E4  3A  27  4A    13.0          LDA   SDDRA           ;GET THE DDR A DATA
1711  44E7                             ROUT  ADATA           ;SAVE IN CASE OF ERROR DETECTED
 (1)  44E7  D3  94       10.0                OUT   ADATA          ;WRITE AC INTO ADATA
 (1)  44E9  7F            4.0                MOV   A,A            ;RETRY LINK
1712  44EA  2A  D2  48    16.0          LHLD  EXPADR          ;POINT TO THE EXPECTED DATA TABLE
1713  44ED  BE            7.0          CMP   M               ;ACTUAL DATA = EXPECTED?
1714  44EE  CA  FA  44    10.0          JZ    T1CKAC          ;JUMP IF OK
1715  44F1  7E            7.0          MOV   A,M             ;GET THE EXPECTED DATA
1716  44F2                             ROUT  EDATA           ;SAVE FOR ERROR REPORT
 (1)  44F2  D3  95       10.0                OUT   EDATA          ;WRITE AC INTO EDATA
 (1)  44F4  7F            4.0                MOV   A,A            ;RETRY LINK
1717  44F5                             ERRB  T1LOOP,T1CKAC,@10
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44F5  CD  12  28    18.0                CALL  ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0004                              MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44F8  04                                .BYTE MSGN            ;MESSAGE NUMBER ID
 (1)  44F9  08                                .BYTE @10             ;PRINT ROUTINE NUMBER
 (1)  44FA  CD  15  28    18.0          T1CKAC::    CALL  CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44FD  DA  70  43    10.0                JC    T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1718                             ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 7-0
```

```
1719  4500  23                    6.0  T1CKB:  INX   H                ;POINT TO THE EXPECTED DATA FOR DDR B
1720  4501  3A  28  4A            13.0         LDA   SDDRB            ;GET THE DDR B DATA
1721  4504                               ROUT  ADATA                  ;SAVE FOR ERROR REPORT
 (1)  4504  D3  94                10.0         OUT   ADATA                ;WRITE AC INTO ADATA
 (1)  4506  7F                     4.0         MOV   A,A                  ;RETRY LINK
1722  4507  BE                     7.0         CMP   M                  ;ACTUAL = EXPECTED DATA?
1723  4508  CA  14  45            10.0         JZ    T1CKBC             ;JUMP IF OK
1724  450B  7E                     7.0         MOV   A,M                ;GET THE EXPECTED DATA
1725  450C                               ROUT  EDATA                  ;SAVE FOR ERROR REPORT
 (1)  450C  D3  95                10.0         OUT   EDATA                ;WRITE AC INTO EDATA
 (1)  450E  7F                     4.0         MOV   A,A                  ;RETRY LINK
1726  450F                               ERRB  T1LOOP,T1CKBC,@10
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  450F  CD  12  28            18.0                CALL  ERLPB            ;PROCESS ERROR - DO 2.3
 (1)       0005                                       MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4512  05                                        .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  4513  08                                        .BYTE @10              ;PRINT ROUTINE NUMBER
 (1)  4514  CD  15  28            18.0         T1CKBC::    CALL  CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4517  DA  70  43            10.0                JC    T1LOOP           ;LOOP ADDRESS IF LOOP SPECIFIED
1727                          ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 15-8
1728
1729  451A  23                    6.0  T1CKC:  INX   H                ;POINT TO EXPECTED DDR C DATA
1730  451B  3A  29  4A            13.0         LDA   SDDRC            ;GET DDR C DATA
1731  451E                               ROUT  ADATA                  ;SAVE FOR ERROR REPORT
 (1)  451E  D3  94                10.0         OUT   ADATA                ;WRITE AC INTO ADATA
 (1)  4520  7F                     4.0         MOV   A,A                  ;RETRY LINK
1732  4521  BE                     7.0         CMP   M                  ;ACTUAL DATA = EXPECTED DDRC DATA?
1733  4522  CA  2E  45            10.0         JZ    T1CKCC             ;JUMP IF OK
1734  4525  7E                     7.0         MOV   A,M                ;GET THE EXPECTED DATA
1735  4526                               ROUT  EDATA                  ;SAVE FOR ERROR REPORT
 (1)  4526  D3  95                10.0         OUT   EDATA                ;WRITE AC INTO EDATA
 (1)  4528  7F                     4.0         MOV   A,A                  ;RETRY LINK
1736  4529                               ERRB  T1LOOP,T1CKCC,@10
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4529  CD  12  28            18.0                CALL  ERLPB            ;PROCESS ERROR - DO 2.3
 (1)       0006                                       MSGN  =     MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  452C  06                                        .BYTE MSGN             ;MESSAGE NUMBER ID
 (1)  452D  08                                        .BYTE @10              ;PRINT ROUTINE NUMBER
 (1)  452E  CD  15  28            18.0         T1CKCC::    CALL  CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4531  DA  70  43            10.0                JC    T1LOOP           ;LOOP ADDRESS IF LOOP SPECIFIED
1737                          ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 17-16
1738
1739  4534  2A  D2  48            16.0         LHLD  EXPADR           ;GET THE CURRENT EXPECTED DATA ADDR.
1740  4537  23                     6.0         INX   H
1741  4538  23                     6.0         INX   H                ;UPDATE THE EXPECTED ADDRESS POINTER
1742  4539  23                     6.0         INX   H                ;TO THE DATA
1743  453A  22  D2  48            16.0         SHLD  EXPADR           ;SAVE THE 'NEW' EXPECTED DATA TABLE ADDR.
1744  453D  3A  2A  4A            13.0         LDA   SCLKCT           ;GET THE SCLK COUNT
1745  4540  21  2B  4A            10.0         LXI   H,ESCLK          ;POINT TO THE EXPECTED SCLK COUNT
1746  4543  BE                     7.0         CMP   M                ;DONE WITH THIS TEST PASS?
1747  4544  C2  63  44            10.0         JNZ   T1CLKA           ;JUMP IF STILL EXPECTING MORE SCLKS
1748
1749                          ;UPDATE THE PACKING MODE NUMBER FOR THE NEXT PASS
1750
```

```
1751  4547  3A  2C  4A      13.0          LDA     FMTNUM          ;GET THE CURRENT NUMBER
1752  454A  3C                4.0          INR     A
1753  454B  32  2C  4A      13.0          STA     FMTNUM          ;SAVE THE NEW NUMBER
1754  454E  FE  05            7.0          CPI     5               ;READY TO DO MODE #6??
1755  4550  C2  5B  45       10.0          JNZ     T1PM6           ;SEE IF TIME TO DO MODE #0
1756  4553  3E  06            7.0          MVI     A,6             ;YUP (ALREADY DID #5 ON 1ST PASS)
1757  4555  32  2C  4A      13.0          STA     FMTNUM          ;SAVE IT
1758  4558  C3  70  43       10.0          JMP     T1LOOP          ;DO MODE 6
1759  455B  FE  06            7.0  T1PM6:  CPI     6               ;READY TO START AT MODE 0
1760  455D  CA  68  45       10.0          JZ      T1PM0           ;YUP
1761  4560  FE  07            7.0          CPI     7               ;DONE WITH ALL THE PACKING MODES?
1762  4562  CA  6F  45       10.0          JZ      EXIT            ;YUP
1763  4565  C3  70  43       10.0          JMP     T1LOOP          ;DO THE NEXT PACKING MODE
1764
1765  4568  AF                4.0  T1PM0:  XRA     A               ;START AT MODE 0
1766  4569  32  2C  4A      13.0          STA     FMTNUM          ;SAVE THE PACKING MODE #
1767  456C  C3  70  43       10.0          JMP     T1LOOP          ;DO THE MODE
1768
1769  456F                          EXIT:   ENDTST  FOUND
 (1)                          ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  456F                            REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  456F  CD  06  28       18.0                            CALL    REQST
 (2)  4572  00                                               .BYTE                   ;DATA PATTERN NUMBER
 (2)  4573  00  00                                           .WORD                   ;SYSTEM '"' COUN
 (2)  4575  00  00                                           .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4577  00                                               .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4578  07                                               .BYTE   7               ;REQUEST CODE
 (1)  4579  3A  9A  4F       13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  457C  3D                4.0          DCR     A               ;DOWNCOUNT
 (1)  457D  32  9A  4F       13.0          STA     ITERA           ;SAVE COUNT
 (1)  4580  F2  27  43       10.0          JP      FOUND           ;DO TEST UNTIL TILL = 0
1770  4583  CD  7C  49       18.0          CALL    CLEAR           ;CLEAR ALL TU PORTS
```

```
                                .SBTTL  TEST 2 - WMC BYTE ASSEMBLY - READ REVERSE
                        4586    ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*WMC BYTE ASSEMBLY FROM READ REVERSE DATA
                        4586    SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST WILL INJECT DATA THROUGH THE READ PATH FIFO'S AND CLOCK THE
                                ;*SYSTEM UNTIL AN 'SCLK' APPEARS TO SIGNAL 18 BITS OF DATA READY FOR
                                ;*TRANSMISSION ACROSS THE MASSBUS. EACH 'SCLK' TIME WILL HAVE THE 'DDR'
                                ;*REGISTERS SETUP WITH THE ASSEMBLED DATA AS COMMANDED BY THE PACKING
                                ;*MODE SELECTED IN THE TEST. ALL PACKING MODES ARE TESTED (I.E. 11-NORM
                                ;*TO 10-HI DEN COMPAT) IN A READ REVERSE DATA TRANSFER. THE TEST WILL
                                ;*CLOCK THE SYSTEM ONLY UNTIL THE EXPECTED AMOUNT OF 'SCLK' DATA HAS
                                ;*BEEN TESTED TO VERIFY THAT THE SELECTED PACKING MODE IS WORKING AS
                                ;*EXPECTED. THE READ PATH WILL BE FILLED WITH 8 SUBGROUPS WORTH OF DATA
                                ;*REGUARDLESS OF THE AMOUNT OF EXPECTED DATA GROUPS TO BE TESTED (USUALLY
                                ;*4 SUBGROUPS OF DATA WILL BE SUFFICIENT TO TEST A PACKING MODE). THE
                                ;*FIRST TWO GROUPS OF DATA ARE SIMULATED 'CRC' DATA GROUP AND THE 'RESID'
                                ;*DATA GROUP. THE RESIDUAL DATA GROUP WILL CONTAIN NO DATA CHARACTERS,
                                ;*JUST THE 6 PAD CHARACTERS AND A FAKE ACRC & FAKE ECC CHARACTER (ALL 0'S).
                                ;*THE CRC GROUP WILL CONTAIN A RESIDUAL CHARACTER THAT INDICATED NO DATA
                                ;*BYTES IN THE RESIDUAL DATA GROUP. EVERY 4 CHARACTER BYTES INJECTED WILL
                                ;*BE TURNED INTO A 5 BYTE SUBGROUP TO SIMULATE DATA FROM THE 'TAPE'. THE 'ECC'
                                ;*WILL BE CALCULATED FOR EACH 7 BYTE OF DATA INJECTED AND ALSO INJECTED
                                ;*AT THE PROPER TIME INTO THE FIFO'S. ALL DATA INJECTED INTO THE FIFO'S
                                ;*WILL BE REVERSED TO SIMULATE A READ REVERSE DATA TRANSFER.
                        4586    SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  REQUEST A TU PORT FROM 'HOST' CPU
                                ;*  IF NO TU PORT SELECTED FROM 'HOST' CPU
                                ;*  :  THEN-REPORT OPERATOR ERROR
                                ;*  :  ELSE-PROCEED
                                ;*  ENDIF
                                ;*  BGNDO
                                ;*  :  CLEAR ALL PORTS
                                ;*  :  INIT THE SYSTEM
                                ;*  :  SET READ PATH TO NORMAL CLOCK SPEED
                                ;*  :  INIT COUNTERS + CONTROLS
                                ;*  :  INIT PACKING MODE TO 'IMAGE'
                                ;*  :  LOAD BYTE COUNTER + PAD COUNTER
                                ;*  :  RESTART WRITE MICROCONTROLLER
                                ;*  :  CLOCK WMC THROUGH RESTART CODE
                                ;*  :  START 'GCR' READ DATA XFR
                                ;*  :  CALCULATE ALL NECESSARY ECC CHARACTERS FOR INJECTION
                                ;*  :  SET READ PATH PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
                                ;*  :  CLOCK THE FIFO'S
                                ;*  :  ISSUE 'CLEAR ALL' TO READ PATH
```

```
                                ;*  :  WAIT
                                ;*  :  STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
                                ;*  :  CONVERT SUBGROUPS TO 4X5 & LOAD INTO FIFO
                                ;*  :  SETUP A WATCHDOG TIMER
                                ;*  :  CLOCK SYSTEM TO XFR DATA FROM FIFO TO WMC BYTE ASSY LOGIC
                                ;*  :  IF WATCHDOG TIMER=0
                                ;*  :  :  THEN-REPORT NO 'SCLK'
                                ;*  :  :  ELSE-CONT
                                ;*  :  ENDIF
                                ;*  :  CLOCK SYSTEM TILL SCLK=0...DATA READY TO BE TESTED
                                ;*  :  GET 'SCLK' 18 BIT DATA FROM BYTE ASSY INTO 'DDR' REGISTER
                                ;*  :  IF 18 BIT DATA SAME AS EXPECTED
                                ;*  :  :  THEN-PROCEED
                                ;*  :  :  ELSE-REPORT BYTE ASSY ERROR + WHAT WAS ASSEMBLED
                                ;*  :  ENDIF
                                ;*  :  POINT TO NEXT SET OF EXPECTED DATA
                                ;*  :  IF ALL DONE WITH CURRENT XFR
                                ;*  :  :  THEN-UPDATE TO NEXT PACKING MODE
                                ;*  :  :     IF ALL MODES DONE
                                ;*  :  :     :  THEN-EXIT
                                ;*  :  :     :  ELSE-CONTINUE
                                ;*  :  :     ENDIF
                                ;*  :  :  ELSE-CONTINUE
                                ;*  :  ENDIF
                                ;*  :  DO UNTIL ALL 'SCLK' CHECKED FOR SELECTED XFR
                                ;*  ENDDO
                                ;*ENDTST
1844  4586                      SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1845                            ;*WMC6 MICRO TEST 02
1846                            ;*WMC6 MICRO ERROR 07
1847                            ;*WMC BYTE ASSY FOR READ REVERSE XFR
1848                            ;*M8959 (WMC) & ALL OTHER BOARDS
1849                            ;*TIMEOUT WHILE WAITING FOR 'SCLK' TO SET IN DBUSSTA
1850                            ;*AFTER ISSUEING A READ REVERSE COMMAND TO THE TM78
1851                            ;*FATAL ERROR - TEST ABORTED
1852                            ;*
1853                            ;*WMC6 MICRO TEST 02
1854                            ;*WMC6 MICRO ERROR 10
1855                            ;*WMC BYTE ASSY FOR READ REVERSE XFR
1856                            ;*M8959 (WMC) & ALL OTHER BOARDS
1857                            ;*TIMEOUT WHILE WAITING FOR 'SCLK' TO CLEAR AFTER IT SET
1858                            ;*DURING A READ REVERSE XFR
1859                            ;*FATAL ERROR - TEST ABORTED
1860                            ;*
1861                            ;*WMC6 MICRO TEST 02
1862                            ;*WMC6 MICRO ERROR 11
1863                            ;*WMC BYTE ASSY FOR READ REVERSE XFR
1864                            ;*M8959 (WMC) & ALL OTHER BOARDS
1865                            ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 7-0
1866                            ;*ACTUAL = NNNN
1867                            ;*EXPECTED = NNNN
```

```
                                        ;*
                                        ;*WMC6 MICRO TEST 02
                                        ;*WMC6 MICRO ERROR 12
                                        ;*WMC BYTE ASSY FOR READ REVERSE XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 15-8
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*WMC6 MICRO TEST 02
                                        ;*WMC6 MICRO ERROR 13
                                        ;*WMC BYTE ASSY FOR READ REVERSE XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 17-16
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
1884  4586                              S
 (1)                                    ; *****************************************************************************

1886  4586                              TEST2:  TESTX   @2
 (1)  4586  3E  02            7.0                       MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4588  CD  03  28       18.0                       CALL    TSET            ;SETUP THE TEST
                                        ;%WMC BYTE ASSY FOR READ REVERSE XFR
                                        ;&M8959 (WMC) & ALL OTHER BOARDS
                                        ·SETUP THE NECESSARY FLAGS FOR READ REVERSE

1891  458B  3A  2E  4A          13.0  T2LP:   LDA     SUNIT           ;GET THE SELECTED UNIT
1892  458E  47                   4.0          MOV     B,A
1893  458F  DB  E0               10.0          IN      INTSTA          ;GET THE PORT SELECT STATUS
1894  4591  E6  80                7.0          ANI     BIT7
1895  4593  B0                   4.0          ORA     B               ;ADD IN THE SELECTED PORT #
1896  4594  D3  EC               10.0          OUT     MBSEL           ;SELECT THE DESIRED PORT TO USE
1897  4596  3E  60                7.0          MVI     A,P.LWR!P.LCS   ;GET THE LOOP-COMMAND-STATUS BIT
1898  4598  D3  48               10.0          OUT     PDIAG           ;SELECT IT
1899  459A  AF                   4.0          XRA     A
1900  459B  D3  44               10.0          OUT     TAMT            ;CLEAR THE AMTIE LINES
1901  459D  D3  D2               10.0          OUT     TRKENA          ;CLEAR THE TRACK ENABLE BITS
1902  459F  D3  D2               10.0          OUT     TRKENA
1903  45A1  3E  05                7.0          MVI     A,5
1904  45A3  32  2C  4A           13.0          STA     FMTNUM          ;INIT THE PACKING MODE TO IMAGE MODE
1905  45A6  3E  04                7.0          MVI     A,P.RPEN        ;SET READ PATH ENABLE
1906  45A8  D3  4C               10.0          OUT     PENAB           ;NOW
1907  45AA  3E  10                7.0          MVI     A,RDCLK         ;SETUP THE READ AND WRITE CLOCK TO 'NORMAL'
1908  45AC  D3  F0               10.0          OUT     CLKCTL
1909  45AE  3E  FF                7.0          MVI     A,$FF
1910  45B0  32  2F  4A           13.0          STA     REVERSE         ;SET TO DO A READ REVERSE XFR
1911  45B3  AF                   4.0          XRA     A
1912  45B4  D3  C0               10.0          OUT     DBUSCTL         ;INIT THE DATA BUS CONTROL

                                        ;INIT THE COUNTER CONTROL AND COUNTERS

1916  45B6                              T2LOOP: INICNT
 (1)  45B6  3E  30            7.0                       MVI     A,@060          ;INIT THE
 (1)  45B8  D3  D7           10.0                       OUT     CNTCTL          ;BYTE COUNTER
```

```
   (1)  45BA  3E  70          7.0                          MVI   A,@160          ;INIT THE
   (1)  45BC  D3  D7         10.0                          OUT   CNTCTL          ;PAD COUNTER
   (1)  45BE  3E  B0          7.0                          MVI   A,@260          ;INIT THE
   (1)  45C0  D3  D7         10.0                          OUT   CNTCTL          ;ERROR COUNTER
1917    45C2                                CLRECT
   (1)  45C2  AF              4.0                          XRA   A               ;CLEAR THE ACCUMULATOR
   (1)  45C3  D3  D6         10.0                          OUT   ERRCNT          ;CLEAR BITS 7-0
   (1)  45C5  D3  D6         10.0                          OUT   ERRCNT          ;CLEAR BITS 15-8
1918    45C7  3A  2C  4A      13.0                LDA   FMTNUM                   ;GET THE SELECTED PACKING MODE #
1919    45CA  87               4.0                ADD   A                       ;DOUBLE THE COUNT - 2 ADDR PER ENTRY IN TABLE
1920    45CB  5F               4.0                MOV   E,A
1921    45CC  16  00           7.0                MVI   D,0
1922    45CE  21  BD  48      10.0          T2RR: LXI   H,EDREV                 ;POINT TO READ REVERSE
1923    45D1  19              10.0                DAD   D                       ;GET OFFSET TO EXPECTED DATA
1924    45D2  7E               7.0                MOV   A,M                     ;GET LOW BYTE OF ADDRESS
1925    45D3  32  D2  48      13.0                STA   EXPADR                  ;SAVE IT
1926    45D6  23               6.0                INX   H                       ;POINT TO THE HIGH BYTE
1927    45D7  7E               7.0                MOV   A,M                     ;GET IT
1928    45D8  32  D3  48      13.0                STA   EXPADR+1                ;SAVE HIGH BYTE OF EXPECTED DATA ADDRESS
1929    45DB  3A  2C  4A      13.0                LDA   FMTNUM                  ;GET THE PACKING MODE # AGAIN
1930    45DE  5F               4.0                MOV   E,A
1931    45DF  16  00           7.0                MVI   D,0
1932    45E1  21  CB  48      10.0                LXI   H,ESCLKT                ;POINT TO TABLE OF EXPECTED SCLKS/MODE
1933    45E4  19              10.0                DAD   D                       ;POINT TO TABLE OFFSET #
1934    45E5  7E               7.0                MOV   A,M                     ;GET THE EXPECTED # SCLKS/MODE SELECTED
1935    45E6  32  2B  4A      13.0                STA   ESCLK                   ;SAVE THE NUMBER FOR THIS PASS
1936    45E9  3A  2C  4A      13.0                LDA   FMTNUM                  ;GET THE PACKING MODE # AGAIN
1937    45EC  07               4.0                RLC
1938    45ED  07               4.0                RLC
1939    45EE  07               4.0                RLC
1940    45EF  07               4.0                RLC
1941    45F0  E6  70           7.0                ANI   $70                     ;POSITION THE FMT #
1942    45F2  D3  D0          10.0                OUT   DATACTL                 ;LOAD THE DATA CONTROL WORD
1943    45F4                                      ROUT  R02H                    ;SAVE FOR ERROR PRINTING
   (1)  45F4  D3  85          10.0                      OUT   R02H                    ;WRITE AC INTO R02H
   (1)  45F6  7F              4.0                       MOV   A,A                     ;RETRY LINK
1944    45F7  3E  9B           7.0                MVI   A,@233                  ;SET THE DDR CONTROL TO 'IN'
1945    45F9  D3  DB          10.0                OUT   DDRCTL                  ;TO READ THE 'DDR'
1946    45FB  3E  00           7.0                MVI   A,@0                    ;GET THE RESIDUAL CHAR WORD
1947    45FD  D3  D1          10.0                OUT   RESCHR                  ;LOAD IT
1948    45FF  3E  32           7.0                MVI   A,50                    ;SETUP THE BYTE COUNTER FOR 8 DATA GROUPS
1949    4601  D3  D4          10.0                OUT   BYTCNT
1950    4603  AF               4.0                XRA   A
1951    4604  D3  D4          10.0                OUT   BYTCNT
1952    4606  3E  06           7.0                MVI   A,6                     ;NEED 6 PAD CHARACTERS FOR THE XFR
1953    4608  D3  D5          10.0                OUT   PADCNT                  ;LOAD IT
1954    460A  AF               4.0                XRA   A
1955    460B  D3  D5          10.0                OUT   PADCNT
1956    460D                                      ROUT  R02L                    ;CLEAR OUT THE REST OF REG 2 IN CAS
   (1)  460D  D3  84          10.0                      OUT   R02L                    ;WRITE AC INTO R02L
   (1)  460F  7F              4.0                       MOV   A,A                     ;RETRY LINK
1957    4610                                      ROUT  R05L                    ;CLEAR THE BYTE/SCLK COUNTER FOR REPORTS
   (1)  4610  D3  8A          10.0                      OUT   R05L                    ;WRITE AC INTO R05L
   (1)  4612  7F              4.0                       MOV   A,A                     ;RETRY LINK
```

```
1958  4613  32  2A  4A   13.0          STA     SCLKCT          ;INIT THE SCLK DETECTED COUNTER
1959
1960                                ;RESTART THE WMC AND SET WMC TO SINGLE STEP MODE
1961
1962  4616  3E  01       7.0          MVI     A,W.RST         ;GET THE WMC RESTART BIT
1963  4618  D3  D3      10.0          OUT     WMCCTL          ;SET RESTART
1964  461A  3E  15       7.0          MVI     A,SSTEP!RDCLK   ;GET SINGLE STEP FOR WMC
1965  461C  D3  F0      10.0          OUT     CLKCTL          ;SET SINGLE STEP MODE
1966  461E  32  30  4A   13.0          STA     CCTLWD          ;SAVE THE WORD
1967  4621  CD  81  4A   18.0          CALL    CLKSYS          ;SET THE RESTART
1968  4624  CD  81  4A   18.0          CALL    CLKSYS
1969  4627  AF          4.0          XRA     A
1970  4628  D3  D3      10.0          OUT     WMCCTL          ;FINISH THE RESTART CYCLE
1971
1972                                ;CLOCK THE WMC THRU ITS RESTART AREA
1973
1974  462A  06  B4       7.0          MVI     B,180           ;GET A LOOP COUNT
1975  462C  CD  81  4A   18.0  CLKRS2: CALL    CLKSYS          ;CLOCK THE SYSTEM
1976  462F  05           4.0          DCR     B               ;DECREMENT THE LOOP COUNTER
1977  4630  C2  2C  46   10.0          JNZ     CLKRS2          ;CLOCK TILL COUNT RUN OUT
1978
1979  4633  3E  94       7.0          MVI     A,W.GCR!W.ENAB!W.REV    ;SETUP FOR A GCR DATA XFR
1980  4635  D3  D3      10.0  T2ST:   OUT     WMCCTL          ;LOAD THE WORD IN WMC
1981                                ;CALCULATE THE CRC DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
1982
1983  4637  21  7F  46   10.0  C2EC1:  LXI     H,CRCG          ;POINT THE THE 1ST DATA GROUP DATA TABLE
1984  463A  06  07       7.0          MVI     B,7             ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
1985  463C  CD  42  4A   18.0          CALL    CLECC           ;INIT THE ECC CHARACTER
1986
1987  463F  7E           7.0  C2EC1L: MOV     A,M             ;GET A DATA BYTE FROM THE TABLE
1988  4640  CD  47  4A   18.0          CALL    ECC             ;CALC ECC WITH THIS CHARACTER
1989  4643  23           6.0          INX     H               ;POINT TO THE NEXT DATA BYTE IN TABLE
1990  4644  05           4.0          DCR     B               ;DECREMENT THE COUNTER OF DATA BYTES
1991  4645  C2  3F  46   10.0          JNZ     C2EC1L          ;STAY HERE TILL ALL DATA BYTES USED
1992  4648  3A  6E  4A   13.0          LDA     ECCCHR          ;GET THE CALCULATED ECC CHARACTER
1993  464B  32  86  46   13.0          STA     CRCGEC          ;SAVE THE CALCULATED ECC CHARACTER
1994
1995                                ;CALCULATE THE 1ST DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
1996
1997  464E  21  9F  48   10.0  C2EC2:  LXI     H,RDG1          ;POINT THE THE 1ST DATA GROUP DATA TABLE
1998  4651  06  07       7.0          MVI     B,7             ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
1999  4653  CD  42  4A   18.0          CALL    CLECC           ;INIT THE ECC CHARACTER
2000
2001  4656  7E           7.0  C2EC2L: MOV     A,M             ;GET A DATA BYTE FROM THE TABLE
2002  4657  CD  47  4A   18.0          CALL    ECC             ;CALC ECC WITH THIS CHARACTER
2003  465A  23           6.0          INX     H               ;POINT TO THE NEXT DATA BYTE IN TABLE
2004  465B  05           4.0          DCR     B               ;DECREMENT THE COUNTER OF DATA BYTES
2005  465C  C2  56  46   10.0          JNZ     C2EC2L          ;STAY HERE TILL ALL DATA BYTES USED
2006  465F  3A  6E  4A   13.0          LDA     ECCCHR          ;GET THE CALCULATED ECC CHARACTER
2007  4662  32  A6  48   13.0          STA     RDG1EC          ;SAVE THE CALCULATED ECC CHARACTER
2008                                ;CALCULATE THE 2ND DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
2009
2010  4665  21  A7  48   10.0  C2EC3:  LXI     H,RDG2          ;POINT THE THE 2ND DATA GROUP DATA TABLE
2011  4668  06  07       7.0          MVI     B,7             ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
```

```
2012  466A  CD  42  4A      18.0          CALL    CLECC           ;INIT THE ECC CHARACTER
2013
2014  466D  7E                 7.0  C2EC3L: MOV     A,M             ;GET A DATA BYTE FROM THE TABLE
2015  466E  CD  47  4A        18.0          CALL    ECC             ;CALC ECC WITH THIS CHARACTER
2016  4671  23                 6.0          INX     H               ;POINT TO THE NEXT DATA BYTE IN TABLE
2017  4672  05                 4.0          DCR     B               ;DECREMENT THE COUNTER OF DATA BYTES
2018  4673  C2  6D  46        10.0          JNZ     C2EC3L          ;STAY HERE TILL ALL DATA BYTES USED
2019  4676  3A  6E  4A        13.0          LDA     ECCCHR          ;GET THE CALCULATED ECC CHARACTER
2020  4679  32  AE  48        13.0          STA     RDG2EC          ;SAVE THE CALCULATED ECC CHARACTER
2021  467C  C3  87  46        10.0          JMP     T2PLO
2022
2023  467F                           RESG:
2024  467F  00                       CRCG:   .BYTE   0               ;CRC GROUP
2025  4680  00                               .BYTE   0               ;CRC GROUP 'CRC' CHARACTER
2026  4681  00                               .BYTE   0               ;CRC GROUP 'CRC' CHAR
2027  4682  00                               .BYTE   0               ;CRC GROUP 'CRC' CHAR
2028
2029  4683                           RESG2:
2030  4683  00                       CRCG2:  .BYTE   0
2031  4684  00                               .BYTE   0
2032  4685  00                               .BYTE   0               ;RESIDUAL CHARACTER
2033  4686  00                       CRCGEC: .BYTE   0
2034  4687  3E  A8                7.0  T2PLO:  MVI     A,R.PLO1!R.TBJN.R.PLOD  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
2035                                                                           ;SET TIE BUS JAM AND PLO DISABLE
2036  4689  D3  09                10.0          OUT     RPCTL
2037  468B  AF                   4.0          XRA     A
2038  468C  D3  0A                10.0          OUT     RTIEB           ;CLEAR THE TIE BUS
2039  468E  D3  08                10.0          OUT     RFIFOL          ;CLOCK THE FIFO'S
2040  4690  3E  0D                 7.0          MVI     A,RCLRT         ;ISSUE CLEAR ALL COMMAND
2041  4692  D3  0B                10.0          OUT     RCMD
2042
2043  4694  00                   4.0          NOP                     ;WAIT
2044  4695  00                   4.0          NOP                     ;WAIT
2045  4696  00                   4.0          NOP
2046  4697  00                   4.0          NOP
2047  4698  00                   4.0          NOP
2048
2049  4699  3E  A9                7.0          MVI     A,R.PLO1!R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
2050                                                                           ;SET TIE BUS JAM AND PLO DISABLE
2051  469B  D3  09                10.0          OUT     RPCTL
2052
2053  469D  21  7F  46          10.0  T2SG1:  LXI     H,CRCG          ;POINT TO THE CRC DATA GROUP
2054  46A0  CD  43  48          18.0          CALL    CONVRT          ;CONVERT IT TO 4X5
2055  46A3  CD  58  48          18.0          CALL    LOADIT          ;PUT IT IN THE FIFO
2056
2057  46A6  21  83  46          10.0  T2SG2:  LXI     H,CRCG2         ;POINT TO 2ND HALF OF CRC GROUP
2058  46A9  CD  43  48          18.0          CALL    CONVRT          ;CHANGE IT
2059  46AC  CD  58  48          18.0          CALL    LOADIT          ;AND PUT IT IN THE FIFO
2060
2061  46AF  21  7F  46          10.0  T2SG3:  LXI     H,RESG          ;POINT TO THE RESIDUAL DATA GROUP
2062  46B2  CD  43  48          18.0          CALL    CONVRT          ;CHANGE IT
2063  46B5  CD  58  48          18.0          CALL    LOADIT          ;AND PUT IT IN THE FIFO
2064
2065  46B8  21  83  46          10.0  T2SG4:  LXI     H,RESG2         ;POINT TO THE 2ND HALF OF RESIDUAL DATA
```

```
2066  46BB  CD  43  48      18.0            CALL    CONVRT          ;CHANGE IT
2067  46BE  CD  58  48      18.0            CALL    LOADIT          ;AND PUT IT IN THE FIFO
2068
2069  46C1  21  35  4A      10.0  T2EMK:    LXI     H,TRNOUT        ;SETUP TO LOAD THE END MARK SIGNAL
2070  46C4  06  05           7.0            MVI     B,5
2071  46C6  3E  FF           7.0  T2EMKL:    MVI     A,$FF
2072  46C8  77               7.0            MOV     M,A             ;SAVE A CHAR
2073  46C9  23               6.0            INX     H               ;POINT TO PARITY BIT STORAGE
2074  46CA  3E  01           7.0            MVI     A,1
2075  46CC  77               7.0            MOV     M,A
2076  46CD  23               6.0            INX     H
2077  46CE  05               4.0            DCR     B               ;DECREMENT LOOP COUNTER
2078  46CF  C2  C6  46      10.0            JNZ     T2EMKL          ;LOOP TILL END MARK IN 4X5 STORAGE AREA
2079  46D2  CD  58  48      18.0            CALL    LOADIT          ;LOAD THE END MARK INTO FIFO
2080  46D5  21  A3  48      10.0  T2SGD2:    LXI     H,RSG2          ;POINT TO THE 2ND SUBGROUP OF DATA
2081  46D8  CD  43  48      18.0            CALL    CONVRT          ;MAKE IT 4X5
2082  46DB  CD  58  48      18.0            CALL    LOADIT          ;INJECT IT INTO FIFO
2083
2084  46DE  21  9F  48      10.0  T2SGD1:    LXI     H,RDG1          ;POINT TO THE 1ST DATA GROUP FOR TRANSLATION
2085  46E1  CD  43  48      18.0            CALL    CONVRT          ;CONVERT SUBGROUP - 4X5
2086  46E4  CD  58  48      18.0            CALL    LOADIT          ;INJECT THE DATA INTO THE FIFO
2087
2088  46E7  21  AB  48      10.0  T2SGD4:    LXI     H,RSG4          ;POINT TO THE 4TH SUBGROUP OF DATA
2089  46EA  CD  43  48      18.0            CALL    CONVRT          ;MAKE IT 4X5
2090  46ED  CD  58  48      18.0            CALL    LOADIT          ;INJECT THE DATA INTO FIFO
2091
2092  46F0  21  A7  48      10.0  T2SGD3:    LXI     H,RDG2          ;POINT TO THE 2ND DATA GROUP - SUBGROUP 3
2093  46F3  CD  43  48      18.0            CALL    CONVRT          ;MAKE IT 4X5
2094  46F6  CD  58  48      18.0            CALL    LOADIT          ;INJECT THE DATA INTO FIFO
2095  46F9  CD  58  48      18.0            CALL    LOADIT          ;INJECT THE DATA INTO FIFO
2096  46FC  CD  58  48      18.0            CALL    LOADIT          ;INJECT THE DATA INTO FIFO
2097  46FF  AF               4.0            XRA     A               ;CLEAR THE LOAD MORE FLAG
2098  4700  32  41  4A      13.0            STA     MOREFG          ;SO PGM WILL LOAD ANOTHER GROUP ON 1ST
2099                                                                ;SCLK
2100
2101  4703  3E  0B           7.0            MVI     A,DIARD         ;LOAD THE DIAGNOSTIC READ
2102  4705  D3  0B          10.0            OUT     RCMD            ;COMMAND
2103
2104  4707  11  01  00      10.0  T2CLKA:    LXI     D,1             ;SET WATCH DOG INCREMENT
2105  470A  21  68  C5      10.0            LXI     H,-15000        ;SET WATCH DOG COUNT TO 15000
2106  470D  D3  0C          10.0  T2CLK:    OUT     RINST           ;STEP THE READ PATH
2107  470F  CD  81  4A      18.0            CALL    CLKSYS          ;CLOCK THE WMC
2108  4712  DB  C0          10.0            IN      DBUSSTA         ;GET THE DATABUS STATUS
2109  4714  E6  02           7.0            ANI     T.SCLK          ;SEE IF SCLK UP - DATA READY
2110  4716  C2  2B  47      10.0            JNZ     T2SCLK          ;JUMP IF SCLK UP
2111  4719  19              10.0            DAD     D
2112  471A  D2  0D  47      10.0            JNC     T2CLK           ;STAY IN LOOP UNLESS TIMEOUT
2113
```

```
2115  471D                                      ERR     T2LOOP,DUMM3,3
 (1)                                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  471D   CD   09   28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0007                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4720   07                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4721   03                                          .BYTE   3
 (1)  4722   CD   15   28        18.0            DUMM3:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4725   DA   B6   45        10.0                    JC      T2LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2116                                    ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO SET IN DBUSSTA
2117                                    ;>AFTER ISSUEING A READ REVERSE COMMAND TO THE TM78
2118                                    ;<FATAL ERROR - TEST ABORTED
2119  4728   C3   6F   45        10.0            JMP     EXIT
2120
2121                                    ;HERE TO CLOCK THE WMC SO THE 'SCLK' SIGNAL WILL GO =0
2122                                    ;THIS IS NECESSARY BEFORE STROBING THE DDR REGISTERS...THEY ARE ONLY
2123                                    ;VALID AFTER 'SCLK' HAS TRANSITIONED TO ZERO INDICATING THE DATA IS
2124                                    ;ON THE MASSBUS
2125
2126  472B   11   01   00        10.0  T2SCLK: LXI     D,1             ;SETUP THE DROP DEAD LOOP COUNTER
2127  472E   21   68   C5        10.0          LXI     H,-15000        ;SO IT WON'T HANG FOREVER
2128  4731   3A   41   4A        13.0          LDA     MOREFG          ;SEE IF NEED MORE DATA
2129  4734   A7                     4.0          ANA     A               ;WILL ONLY LOAD ANOTHER GROUP IF 1ST SCLK
2130  4735   C2   43   47        10.0          JNZ     T2SCKL          ;JUMP OVER IF NOT 1ST TIME
2131  4738   CD   58   48        18.0          CALL    LOADIT
2132  473B   CD   58   48        18.0          CALL    LOADIT          ;PUT IN ANOTHER DATA GROUP - DUMMY DATA
2133  473E   3E   FF                7.0          MVI     A,$FF
2134  4740   32   41   4A        13.0          STA     MOREFG          ;SET THE DATA FLAG
2135  4743   D3   0C                10.0  T2SCKL: OUT     RINST           ;CLOCK THE READ PATH ONCE
2136  4745   CD   81   4A        18.0          CALL    CLKSYS          ;CLOCK THE WMC TO CLEAR 'SCLK'
2137  4748   DB   C0                10.0          IN      DBUSSTA         ;SEE IF 'SCLK' IS GONE
2138  474A   E6   02                 7.0          ANI     T.SCLK          ;SAVE ONLY THE 'SCLK' BIT IN STATUS WORD
2139  474C   CA   61   47        10.0          JZ      T2SCKO          ;JUMP IF EVER GOES =0
2140  474F   19                     10.0          DAD     D               ;DECREMENT THE LOOP COUNTER
2141  4750   D2   43   47        10 0          JNC     T2SCKL          ;& LOOP TILL DEAD
2142  4753                                    ERR     T2LOOP,DUMM4,3
 (1)                                            ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4753   CD   09   28        18.0                    CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0008                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4756   08                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4757   03                                          .BYTE   3
 (1)  4758   CD   15   28        18.0            DUMM4:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  475B   DA   B6   45        10.0                    JC      T2LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2143                                    ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO CLEAR AFTER IT SET
2144                                    ;>DURING A READ REVERSE XFR
2145                                    ;<FATAL ERROR - TEST ABORTED
2146  475E   C3   29   48        10.0          JMP     EXIT2
2147                                    ;HERE WHEN 'SCLK' CLEARED - 18 BITS OF DATA HAS BEEN ASSEMBLED FROM
2148                                    ;THE BYTE ASSEMBLY LOGIC. DATA CAME FROM THE READ PATH (FIFO).
2149
2150  4761   DB   D8                10.0  T2SCKO: IN      DDRA            ;GET THE DDR A DATA
2151  4763   32   27   4A        13.0          STA     SDDRA           ;SAVE IT
2152  4766                                  ROUT    R16L            ;SAVE FOR PRINTING
 (1)  4766   D3   9C                10.0                    OUT     R16L            ;WRITE AC INTO R16L
 (1)  4768   7F                     4.0                    MOV     A,A             ;RETRY LINK
```

```
 2153  4769  DB  D9        10.0          IN      DDRB            ;GET DDR B DATA
 2154  476B  32  28  4A    13.0          STA     SDDRB           ;SAVE IT
 2155  476E                              ROUT    R16H            ;SAVE FOR ERROR REPORTING
  (1)  476E  D3  9D        10.0                  OUT     R16H            ;WRITE AC INTO R16H
  (1)  4770  7F            4.0                   MOV     A,A             ;RETRY LINK
 2156  4771  DB  DA        10.0          IN      DDRC            ;GET DDR C DATA
 2157  4773  E6  03        7.0           ANI     3
 2158  4775  32  29  4A    13.0          STA     SDDRC           ;SAVE IT
 2159  4778  F6  80        7.0           ORI     BIT7            ;ADD IN THE PRINT FLAG
 2160  477A                              ROUT    R17L            ;SAVE IT FOR PRINTING
  (1)  477A  D3  9E        10.0                  OUT     R17L            ;WRITE AC INTO R17L
  (1)  477C  7F            4.0                   MOV     A,A             ;RETRY LINK
 2161
 2162  477D  3A  2A  4A    13.0          LDA     SCLKCT          ;GET THE CURRENT SCLK COUNT
 2163  4780  3C            4.0           INR     A               ;COUNT THIS SCLK
 2164  4781  32  2A  4A    13.0          STA     SCLKCT          ;SAVE THE NEW COUNT
 2165  4784                              ROUT    R05L            ;ALSO IN CAS FOR ERROR REPORTING
  (1)  4784  D3  8A        10.0                  OUT     R05L            ;WRITE AC INTO R05L
  (1)  4786  7F            4.0                   MOV     A,A             ;RETRY LINK
 2166  4787  AF            4.0           XRA     A
 2167  4788                              ROUT    R05H
  (1)  4788  D3  8B        10.0                  OUT     R05H            ;WRITE AC INTO R05H
  (1)  478A  7F            4.0                   MOV     A,A             ;RETRY LINK
 2168
 2169                                   ;CHECK THE DATA IN DDR A - BITS 7-0 OF THE 18 BITS ASSEMBLED
 2170
 2171  478B  2A  D2  48    16.0  T2CKA:  LHLD    EXPADR          ;POINT TO THE TABLE OF EXPECTED DATA
 2172  478E  7E            7.0           MOV     A,M             ;GET BITS 7-0
 2173  478F                              ROUT    R14L            ;SAVE FOR ERROR REPORTING
  (1)  478F  D3  98        10.0                  OUT     R14L            ;WRITE AC INTO R14L
  (1)  4791  7F            4.0                   MOV     A,A             ;RETRY LINK
 2174  4792  23            6.0           INX     H               ;POINT TO NEXT BYTE
 2175  4793  7E            7.0           MOV     A,M             ;GET BITS 15-8
 2176  4794                              ROUT    R14H            ;SAVE IT
  (1)  4794  D3  99        10.0                  OUT     R14H            ;WRITE AC INTO R14H
  (1)  4796  7F            4.0                   MOV     A,A             ;RETRY LINK
 2177  4797  23            6.0           INX     H
 2178  4798  7E            7.0           MOV     A,M             ;GET BITS 17-16
 2179  4799  F6  80        7.0           ORI     BIT7            ;ADD IN THE PRINT FLAG
 2180  479B                              ROUT    R15L            ;SAVE FOR ERROR REPORT
  (1)  479B  D3  9A        10.0                  OUT     R15L            ;WRITE AC INTO R15L
  (1)  479D  7F            4.0                   MOV     A,A             ;RETRY LINK
 2181  479E  3A  27  4A    13.0          LDA     SDDRA           ;GET THE DDR A DATA
 2182  47A1                              ROUT    ADATA           ;SAVE IN CASE OF ERROR DETECTED
  (1)  47A1  D3  94        10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  47A3  7F            4.0                   MOV     A,A             ;RETRY LINK
 2183  47A4  2A  D2  48    16.0          LHLD    EXPADR          ;POINT TO THE EXPECTED DATA TABLE
 2184  47A7  BE            7.0           CMP     M               ;ACTUAL DATA = EXPECTED?
 2185  47A8  CA  B4  47    10.0          JZ      T2CKAC          ;JUMP IF OK
 2186  47AB  7E            7.0           MOV     A,M             ;GET THE EXPECTED DATA
```

```
2188  47AC                                          ROUT    EDATA           ;SAVE FOR ERROR REPORT
 (1)  47AC   D3   95            10.0                      OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  47AE   7F                  4.0                      MOV     A,A             ;RETRY LINK
2189  47AF                                          ERRB    T2LOOP,T2CKAC,@10
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  47AF   CD   12   28       18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0009                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47B2   09                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47B3   08                                         .BYTE   @10             ;PRINT ROUTINE NUMBER
 (1)  47B4   CD   15   28       18.0              T2CKAC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47B7   DA   B6   45       10.0                      JC      T2LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2190                                        ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 7-0
2191  47BA   23                  6.0        T2CKB:  INX     H               ;POINT TO THE EXPECTED DATA FOR DDR B
2192  47BB   3A   28   4A        13.0                LDA     SDDRB           ;GET THE DDR B DATA
2193  47BE                                          ROUT    ADATA           ;SAVE FOR ERROR REPORT
 (1)  47BE   D3   94            10.0                      OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  47C0   7F                  4.0                      MOV     A,A             ;RETRY LINK
2194  47C1   BE                  7.0                CMP     M               ;ACTUAL = EXPECTED DATA?
2195  47C2   CA   CE   47       10.0                JZ      T2CKBC          ;JUMP IF OK
2196  47C5   7E                  7.0                MOV     A,M             ;GET THE EXPECTED DATA
2197  47C6                                          ROUT    EDATA           ;SAVE FOR ERROR REPORT
 (1)  47C6   D3   95            10.0                      OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  47C8   7F                  4.0                      MOV     A,A             ;RETRY LINK
2198  47C9                                          ERRB    T2LOOP,T2CKBC,@10
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  47C9   CD   12   28       18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000A                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47CC   0A                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47CD   08                                         .BYTE   @10             ;PRINT ROUTINE NUMBER
 (1)  47CE   CD   15   28       18.0              T2CKBC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47D1   DA   B6   45       10.0                      JC      T2LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2199                                        ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 15-8
2200
2201  47D4   23                  6.0        T2CKC:  INX     H               ;POINT TO EXPECTED DDR C DATA
2202  47D5   3A   29   4A        13.0                LDA     SDDRC           ;GET DDR C DATA
2203  47D8                                          ROUT    ADATA           ;SAVE FOR ERROR REPORT
 (1)  47D8   D3   94            10.0                      OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  47DA   7F                  4.0                      MOV     A,A             ;RETRY LINK
2204  47DB   BE                  7.0                CMP     M               ;ACTUAL DATA = EXPECTED DDRC DATA?
2205  47DC   CA   E8   47       10.0                JZ      T2CKCC          ;JUMP IF OK
2206  47DF   7E                  7.0                MOV     A,M             ;GET THE EXPECTED DATA
2207  47E0                                          ROUT    EDATA           ;SAVE FOR ERROR REPORT
 (1)  47E0   D3   95            10.0                      OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  47E2   7F                  4.0                      MOV     A,A             ;RETRY LINK
2208  47E3                                          ERRB    T2LOOP,T2CKCC,@10
 (1)                                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  47E3   CD   12   28       18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000B                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47E6   0B                                         .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  47E7   08                                         .BYTE   @10             ;PRINT ROUTINE NUMBER
 (1)  47E8   CD   15   28       18.0              T2CKCC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47EB   DA   B6   45       10.0                      JC      T2LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
2209                                        ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 17-16
2210
```

```
 2211  47EE  2A  D2  48        16.0          LHLD    EXPADR          ;GET THE CURRENT EXPECTED DATA ADDR.
 2212  47F1  23                   6.0          INX     H
 2213  47F2  23                   6.0          INX     H               ;UPDATE THE EXPECTED ADDRESS POINTER
 2214  47F3  23                   6.0          INX     H               ;TO THE DATA
 2215  47F4  22  D2  48        16.0          SHLD    EXPADR          ;SAVE THE 'NEW' EXPECTED DATA TABLE ADDR.
 2216  47F7  3A  2A  4A        13.0          LDA     SCLKCT          ;GET THE SCLK COUNT
 2217  47FA  21  2B  4A        10.0          LXI     H,ESCLK         ;POINT TO THE EXPECTED SCLK COUNT
 2218  47FD  BE                   7.0          CMP     M               ;DONE WITH THIS TEST PASS?
 2219  47FE  C2  07  47        10.0          JNZ     T2CLKA          ;JUMP IF STILL EXPECTING MORE SCLKS
 2220
 2221                                  ;UPDATE THE PACKING MODE NUMBER FOR THE NEXT PASS
 2222
 2223  4801  3A  2C  4A        13.0          LDA     FMTNUM          ;GET THE CURRENT NUMBER
 2224  4804  3C                   4.0          INR     A
 2225  4805  32  2C  4A        13.0          STA     FMTNUM          ;SAVE THE NEW NUMBER
 2226  4808  FE  05                7.0          CPI     5               ;READY TO DO MODE #6??
 2227  480A  C2  15  48        10.0          JNZ     T2PM6           ;SEE IF TIME TO DO MODE #0
 2228  480D  3E  06                7.0          MVI     A,6             ;YUP (ALREADY DID #5 ON 1ST PASS)
 2229  480F  32  2C  4A        13.0          STA     FMTNUM          ;SAVE IT
 2230  4812  C3  B6  45        10.0          JMP     T2LOOP          ;DO MODE 6
 2231  4815  FE  06                7.0  T2PM6:  CPI     6               ;READY TO START AT MODE 0
 2232  4817  CA  22  48        10.0          JZ      T2PM0           ;YUP
 2233  481A  FE  07                7.0          CPI     7               ;DONE WITH ALL THE PACKING MODES?
 2234  481C  CA  29  48        10.0          JZ      EXIT2           ;YUP
 2235  481F  C3  B6  45        10.0          JMP     T2LOOP          ;DO THE NEXT PACKING MODE
 2236
 2237  4822  AF                   4.0  T2PM0:  XRA     A               ;START AT MODE 0
 2238  4823  32  2C  4A        13.0          STA     FMTNUM          ;SAVE THE PACKING MODE #
 2239  4826  C3  B6  45        10.0          JMP     T2LOOP          ;DO THE MODE
 2240
 2241  4829                         EXIT2:  ENDTST  T2LP
  (1)                                 ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  4829                                   REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
  (2)  4829  CD  06  28        18.0                                CALL    REQST
  (2)  482C  00                                                    .BYTE                   ;DATA PATTERN NUMBER
  (2)  482D  00  00                                                .WORD                   ;SYSTEM '"' COUNT
  (2)  482F  00  00                                                .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  4831  00                                                    .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  4832  07                                                    .BYTE   7               ;REQUEST CODE
  (1)  4833  3A  9A  4F        13.0          LDA     ITERA           ;GET ITERATION COUNT
  (1)  4836  3D                   4.0          DCR     A               ;DOWNCOUNT
  (1)  4837  32  9A  4F        13.0          STA     ITERA           ;SAVE COUNT
  (1)  483A  F2  8B  45        10.0          JP      T2LP            ;DO TEST UNTIL TILL = 0
 2242  483D  CD  7C  49        18.0          CALL    CLEAR           ;CLEAR ALL TU PORTS
 2243  4840  C3  18  28        10.0          JMP     TSTEND          ;ALL DONE!
 2244                                  ;HERE IS THE SUBROUTINES NECESSARY FOR CONVERTING SUBGROUPS AND LOADING
 2245                                  ;THEM INTO THE FIFO
 2246
 2247  4843                         $
  (1)                                 ;*****************************************************************************
 2248                                  ;*CONVRT -- THIS SUBROUTINE WILL GET A SUBGROUP OF DATA AS POINTED TO BY
 2249                                  ;*      THE H & L REGISTERS AND CAUSE THEM TO BE TRANSLATED (4X5). THE
 2250                                  ;*      TRANSLATED DATA WILL BE IN A TABLE CALLED 'TRNOUT'.
 2251  4843                         $
```

```
 (1)                                    ; ***************************************************************************
2252
2253  4843  D5                  12.0  CONVRT: PUSH    D               ;SAVE D & E
2254  4844  11  31  4A          10.0          LXI     D,TRNIN         ;POINT TO THE TEMP STORAGE FOR DATA BYTES
2255  4847  06  04               7.0          MVI     B,4             ;SETUP A LOOP COUNTER FOR 4 BYTES
2256
2257  4849  7E                   7.0  CNV1:   MOV     A,M             ;GET A BYTE OF DATA
2258  484A  EB                   4.0          XCHG                    ;POINT TO THE STORAGE
2259  484B  77                   7.0          MOV     M,A             ;SAVE DATA BYTE IN TEMP STORAGE
2260  484C  EB                   4.0          XCHG                    ;RESET POINTER
2261  484D  23                   6.0          INX     H               ;POINT TO NEXT DATA BYTE
2262  484E  13                   6.0          INX     D               ;POINT TO NEXT TEMP STORAGE ADDRESS
2263  484F  05                   4.0          DCR     B               ;UPDATE THE LOOP COUNT
2264  4850  C2  49  48          10.0          JNZ     CNV1            ;LOOP TILL 4 BYTES IN TEMP STORAGE
2265  4853  CD  A7  49          18.0          CALL    T4X5            ;CONVERT THE 'SUBGROUP' OF DATA
2266  4856  D1                  10.0          POP     D
2267  4857  C9                  10.0          RET
2268  4858                              S
 (1)                                    ; ***************************************************************************
2269                                    ;*LOADIT -- THIS SUBROUTINE WILL TAKE THE TRANSLATED SUBGROUP (4X5)
2270                                    ;*      DATA FROM TABLE 'TRNOUT' AND LOAD THE DATA INTO THE READ
2271                                    ;*      PATH FIFO.
2272  4858                              S
 (1)                                    ; ***************************************************************************
2273
2274  4858  3A  2F  4A          13.0  LOADIT: LDA     REVERSE         ;SEE IF A READ REVERSE XFR
2275  485B  A7                   4.0          ANA     A
2276  485C  C2  76  48          10.0          JNZ     RLOAD           ;REVERSE LOAD INTO FIFO
2277  485F  21  35  4A          10.0  FLOAD:  LXI     H,TRNOUT        ;POINT TO THE TRANSLATED DATA TABLE
2278  4862  06  05               7.0          MVI     B,5             ;GET THE LOOP COUNT FOR SUBGROUP
2279
2280  4864  7E                   7.0  LDFIFO: MOV     A,M             ;GET A BYTE OF DATA
2281  4865  D3  40              10.0          OUT     TCMD            ;STORE IT IN THE CMD REG FOR LOADING
2282  4867  23                   6.0          INX     H               ;POINT TO THE PARITY BIT
2283  4868  7E                   7.0          MOV     A,M             ;GET THE PARITY BIT
2284  4869  07                   4.0          RLC
2285  486A  F6  60               7.0          ORI     P.LWR!P.LCS     ;ADD IN THE OTHER NECESSARY BITS
2286  486C  D3  48              10.0          OUT     PDIAG           ;LOAD INTO CONTROL REG FOR LOADING
2287  486E  D3  08              10.0          OUT     RFIFOL          ;STROBE THE DATA INTO FIFO
2288  4870  23                   6.0          INX     H               ;POINT TO THE NEXT SET OF DATA
2289  4871  05                   4.0          DCR     B               ;DECREMENT THE LOOP COUNTER
2290  4872  C2  64  48          10.0          JNZ     LDFIFO          ;LOOP TILL 5 SETS OF DATA LOADED INTO FIFO
2291  4875  C9                  10.0          RET
2292
2293  4876  21  3D  4A          10.0  RLOAD:  LXI     H,TRNOUT+8      ;POINT TO LAST ENTRY OF SUBGROUP
2294  4879  06  05               7.0          MVI     B,5             ;GET THE LOOP COUNT NEEDED
2295
2296  487B  7E                   7.0  RLFIFO: MOV     A,M             ;GET A BYTE OF DATA
2297  487C  D3  40              10.0          OUT     TCMD            ;STORE IT IN THE CMD REG FOR LOADING
2298  487E  23                   6.0          INX     H               ;POINT TO THE PARITY BIT
2299  487F  7E                   7.0          MOV     A,M             ;GET THE PARITY BIT
2300  4880  07                   4.0          RLC
2301  4881  F6  60               7.0          ORI     P.LWR!P.LCS     ;ADD IN THE OTHER NECESSARY BITS
2302  4883  D3  48              10.0          OUT     PDIAG           ;LOAD INTO CONTROL REG FOR LOADING
```

```
2303  4885  D3  08        10.0        OUT   RFIFOL        ;STROBE THE DATA INTO FIFO
2304  4887  2B             6.0        DCX   H             ;BACK THE POINTER UP TO THE
2305  4888  2B             6.0        DCX   H             ;BEGINING OF THE PREVIOUS
2306  4889  2B             6.0        DCX   H             ;ENTRY PAIR
2307  488A  05             4.0        DCR   B             ;DECREMENT THE LOOP COUNTER
2308  488B  C2  7B  48      10.0       JNZ   RLFIFO        ;LOOP TILL 5 SETS OF DATA LOADED INTO FIFO
2309  488E  C9             10.0       RET
2310                                  ;READ PATH BYTE ASSEMBLY DATA BYTES TO BE INJECTED INTO THE READ PATH
2311                                  ;FIFO - 4 SUBGROUPS WORTH OF DATA - ALL DATA TRANSFERS ARE 'READ FORWARD'
2312
2313  488F  01                        DG1:    .BYTE  @1     ;1ST BYTE IN THE 1ST SUBGROUP
2314  4890  02                                .BYTE  @2     ;1ST SUBGROUP HAS 4 DATA BYTES IN IT
2315  4891  03                                .BYTE  @3
2316  4892  04                                .BYTE  @4     ;LAST BYTE IN THE 1ST SUBGROUP
2317
2318  4893  05                        SG2:    .BYTE  @5     ;1ST BYTE IN 2ND SUBGROUP
2319  4894  06                                .BYTE  @6     ;2ND SUBGROUP HAS 3 BYTES IN IT TO FINISH THE DATA
2320  4895  07                                .BYTE  @7     ;DATA GROUP OF 7 DATA BYTES
2321  4896  00                        DG1ECC: .BYTE  0      ;STORAGE FOR THE 1ST ECC BYTE (CALCULATED)
2322
2323  4897  08                        DG2:    .BYTE  @10    ;1ST BYTE OF THE 3RD SUBGROUP
2324  4898  09                                .BYTE  @11    ;3RD SUBGROUP HAS 4 DATA BYTES
2325  4899  0A                                .BYTE  @12
2326  489A  0B                                .BYTE  @13    ;LAST DATA BYTE IN 3RD SUBGROUP
2327
2328  489B  0C                        SG4:    .BYTE  @14    ;1ST BYTE IN THE 4TH SUBGROUP
2329  489C  0D                                .BYTE  @15    ;THIS SUBGROUP HAS ONLY 3 DATA BYTES
2330  489D  0E                                .BYTE  @16    ;LAST BYTE IN 4TH SUBGROUP
2331  489E  00                        DG2ECC: .BYTE  0      ;2ND DATA GROUP ECC BYTE STORAGE (CALCULATED)
2332
2333                                  ;READ PATH BYTE ASSEMBLY DATA BYTES TO BE INJECTED INTO THE READ PATH
2334                                  ;FIFO - 4 SUBGROUPS WORTH OF DATA - ALL DATA TRANSFERS ARE 'READ REVERSE'
2335
2336  489F  07                        RDG1:   .BYTE  @7     ;1ST BYTE IN THE 1ST SUBGROUP
2337  48A0  06                                .BYTE  @6     ;1ST SUBGROUP HAS 4 DATA BYTES IN IT
2338  48A1  05                                .BYTE  @5
2339  48A2  04                                .BYTE  @4     ;LAST BYTE IN THE 1ST SUBGROUP
2340
2341  48A3  03                        RSG2:   .BYTE  @3     ;1ST BYTE IN 2ND SUBGROUP
2342  48A4  02                                .BYTE  @2     ;2ND SUBGROUP HAS 3 BYTES IN IT TO FINISH THE DATA
2343  48A5  01                                .BYTE  @1     ;DATA GROUP OF 7 DATA BYTES
2344  48A6  00                        RDG1EC: .BYTE  0      ;STORAGE FOR THE 1ST ECC BYTE (CALCULATED)
2345
2346  48A7  0E                        RDG2:   .BYTE  @16    ;1ST BYTE OF THE 3RD SUBGROUP
2347  48A8  0D                                .BYTE  @15    ;3RD SUBGROUP HAS 4 DATA BYTES
2348  48A9  0C                                .BYTE  @14
2349  48AA  0B                                .BYTE  @13    ;LAST DATA BYTE IN 3RD SUBGROUP
2350
2351  48AB  0A                        RSG4:   .BYTE  @12    ;1ST BYTE IN THE 4TH SUBGROUP
2352  48AC  09                                .BYTE  @11    ;THIS SUBGROUP HAS ONLY 3 DATA BYTES
2353  48AD  08                                .BYTE  @10    ;LAST BYTE IN 4TH SUBGROUP
2354  48AE  00                        RDG2EC: .BYTE  0      ;2ND DATA GROUP ECC BYTE STORAGE (CALCULATED)
2355                                  ;THIS IS THE TABLE OF EXPECTED DATA ADDRESSES FOR SELECTED PACKING MODES
2356                                  ;THE DDR REGISTERS WILL BE EXPECTED TO CONTAIN THE DATA IN THESE TABLES
```

```
                                        ;ON EVERY OCCURANCE OF 'SCLK'


                                        ;READ FORWARD EXPECTED DATA TABLES

48AF   D4   48                          EDFWD:  .ADDR   FD11N    ;FWD DATA FOR 11-NORMAL MODE #0
48B1   E9   48                                  .ADDR   FD15N    ;FWD DATA FOR 15-NORMAL MODE #1
48B3   FE   48                                  .ADDR   FD10C    ;FWD DATA FOR 10-COMPAT MODE #2
48B5   22   49                                  .ADDR   FD10D    ;FWD DATA FOR 10-DUMP MODE #3
48B7   3A   49                                  .ADDR   FD10HC   ;FWD DATA FOR 10-HI DEN COMPAT MODE #4
48B9   52   49                                  .ADDR   FDIMAG   ;FWD DATA FOR IMAGE MODE #5
48BB   3A   49                                  .ADDR   FD10HD   ;FWD DATA FOR 10-HI DEN DUMP MODE #6

                                        ;READ REVERSE EXPECTED DATA TABLES

48BD   E9   48                          EDREV:  .ADDR   RD11N    ;REV DATA FOR 11-NORMAL MODE #0
48BF   D4   48                                  .ADDR   RD15N    ;REV DATA FOR 15-NORMAL MODE #1
48C1   10   49                                  .ADDR   RD10C    ;REV DATA FOR 10-COMPAT MODE #2
48C3   2E   49                                  .ADDR   RD10D    ;REV DATA FOR 10-DUMP MODE #3
48C5   46   49                                  .ADDR   RD10HC   ;REV DATA FOR 10-HI DEN COMPAT MODE #4
48C7   52   49                                  .ADDR   RDIMAG   ;REV DATA FOR IMAGE MODE #5
48C9   46   49                                  .ADDR   RD10HD   ;REV DATA FOR 10-HI DEN DUMP MODE #6

                                        ;THIS IS THE TABLE OF EXPECTED SCLK ASSEMBLIES FOR THE SELECTED PACKING
                                        ;MODES

48CB   07                               ESCLKT: .BYTE   7        ;MODE #0 - 11-NORMAL 7 SCLKS
48CC   07                                       .BYTE   7        ;        1 - 15-NORMAL 7 SCLKS
48CD   06                                       .BYTE   6        ;        2 - 10-COMPAT 6 SCLKS (3 36 BIT WORDS)
48CE   04                                       .BYTE   4        ;        3 - 10-DUMP 4 SCLKS (2 36 BIT WORDS)
48CF   04                                       .BYTE   4        ;        4 - 10-HDC 4 SCLKS (2 36 BIT WORDS)
48D0   0E                                       .BYTE   14       ;        5 - IMAGE 14 SCLKS (14 8 BIT WORDS)
48D1   04                                       .BYTE   4        ;        6 - 10-HDD 4 SCLKS (2 36 BIT WORDS)

48D2   00   00                          EXPADR: .ADDR   0        ;ADDRESS OF THE  EXPECTED 'SCLK' DATA TABLE
                                        ;TABLE OF EXPECTED DATA FOR 11-NORMAL BYTE ASSY FOR A READ FWD FUNCTION
                                        ;EVERY 'SCLK' ASSY TIME HAS 3 ENTRYS IN THE TABLE TO REPRESENT THE
                                        ;CONTENTS OF THE 'DDR' REGISTERS (DDR A THRU DDR C).

                                        ;THIS IS PACKING MODE #0 - READ FORWARD ... ALSO 15-NORMAL IF READ REV.

48D4                                    RD15N:
48D4   04                               FD11N:  .BYTE   @4       ;EXPECTED DATA -        DDR A (XFR HAS 7 16.)
48D5   08                                       .BYTE   @10      ;1ST SCLK -             DDR B (BIT WORDS OR)
48D6   00                                       .BYTE   @0       ;                       DDR C (14 DATA BYTES)
48D7   0C                                       .BYTE   @14      ;2ND SCLK               DDR A (DATA IS 1 TO 16(8))
48D8   10                                       .BYTE   @20      ;                       DDR B
48D9   00                                       .BYTE   @0       ;                       DDR C
48DA   14                                       .BYTE   @24      ;3RD SCLK
48DB   18                                       .BYTE   @30
48DC   00                                       .BYTE   @0
48DD   1C                                       .BYTE   @34      ;4TH SCLK
48DE   20                                       .BYTE   @40
48DF   00                                       .BYTE   @0
```

```
2411  48E0  24                        .BYTE   @44     ;5TH SCLK
2412  48E1  28                        .BYTE   @50
2413  48E2  00                        .BYTE   @0
2414  48E3  2C                        .BYTE   @54     ;6TH SCLK
2415  48E4  30                        .BYTE   @60
2416  48E5  00                        .BYTE   @0
2417  48E6  34                        .BYTE   @64     ;7TH & LAST SCLK
2418  48E7  38                        .BYTE   @70
2419  48E8  00                        .BYTE   @0
2420                          ;TABLE OF EXPECTED DATA FOR 15-NORMAL BYTE ASSY FOR A READ FWD FUNCTION
2421                          ;EVERY 'SCLK' ASSY TIME HAS 3 ENTRYS IN THE TABLE TO REPRESENT THE
2422                          ;CONTENTS OF THE 'DDR' REGISTERS (DDR A THRU DDR C).
2423
2424                          ;THIS IS PACKING MODE #1 - READ FORWARD ... ALSO 11-NORMAL IF READ REV.
2425
2426  48E9                    RD11N:
2427  48E9  08                FD15N:  .BYTE   @10     ;EXPECTED DATA -        DDR A (XFR HAS 7 16.)
2428  48EA  04                        .BYTE   @4      ;1ST SCLK -             DDR B (BIT WORDS OR)
2429  48EB  00                        .BYTE   @0      ;                       DDR C (14 DATA BYTES)
2430  48EC  10                        .BYTE   @20     ;2ND SCLK               DDR A (DATA IS 1 TO 16(8))
2431  48ED  0C                        .BYTE   @14     ;                       DDR B
2432  48EE  00                        .BYTE   @0      ;                       DDR C
2433  48EF  18                        .BYTE   @30     ;3RD SCLK
2434  48F0  14                        .BYTE   @24
2435  48F1  00                        .BYTE   @0
2436  48F2  20                        .BYTE   @40     ;4TH SCLK
2437  48F3  1C                        .BYTE   @34
2438  48F4  00                        .BYTE   @0
2439  48F5  28                        .BYTE   @50     ;5TH SCLK
2440  48F6  24                        .BYTE   @44
2441  48F7  00                        .BYTE   @0
2442  48F8  30                        .BYTE   @60     ;6TH SCLK
2443  48F9  2C                        .BYTE   @54
2444  48FA  00                        .BYTE   @0
2445  48FB  38                        .BYTE   @70     ;7TH & LAST SCLK
2446  48FC  34                        .BYTE   @64
2447  48FD  00                        .BYTE   @0
2448                          ;TABLE OF EXPECTED DATA FOR 10-COMPATABILITY BYTE ASSY FOR A READ FWD FUNCTION
2449                          ;EVERY 'SCLK' ASSY TIME HAS 3 ENTRYS IN THE TABLE TO REPRESENT THE
2450                          ;CONTENTS OF THE 'DDR' REGISTERS (DDR A THRU DDR C).
2451
2452                          ;THIS IS PACKING MODE #2 - READ FORWARD
2453
2454  48FE  08                FD10C:  .BYTE   @10     ;1ST SCLK       DDR A (3 COMPLETE 36. BIT)
2455  48FF  04                        .BYTE   @4      ;               DDR B (WORDS OR 12 BYTES IN THE)
2456  4900  00                        .BYTE   @0      ;               DDR C (XFR. DATA = 1 TO 14(8))
2457  4901  00                        .BYTE   @0      ;2ND SCLK       DDR A
2458  4902  C1                        .BYTE   @301    ;               DDR B
2459  4903  00                        .BYTE   @0      ;               DDR C
2460  4904  18                        .BYTE   @30     ;3RD SCLK
2461  4905  14                        .BYTE   @24
2462  4906  00                        .BYTE   @0
2463  4907  00                        .BYTE   @0      ;4TH SCLK
2464  4908  C2                        .BYTE   @302
```

```
2465  4909  01                        .BYTE   @1
2466  490A  28                        .BYTE   @50     ;5TH SCLK
2467  490B  24                        .BYTE   @44
2468  490C  00                        .BYTE   0
2469  490D  00                        .BYTE   @0      ;6TH SCLK (LAST)
2470  490E  C3                        .BYTE   @303
2471  490F  02                        .BYTE   @2
2472
2473                          ;TABLE OF DATA FOR A REVERSE READ DATA TRANSFER
2474
2475  4910  40                  RD10C:  .BYTE   @100    ;1ST SCLK       DDR A (3 COMPLETE 36. BIT)
2476  4911  80                          .BYTE   @200    ;               DDR B (WORDS OR 12 BYTES IN THE)
2477  4912  00                          .BYTE   @0      ;               DDR C (XFR. DATA = 14 TO 1(8))
2478  4913  0C                          .BYTE   @14     ;2ND SCLK       DDR A
2479  4914  10                          .BYTE   @20     ;               DDR B
2480  4915  00                          .BYTE   @0      ;               DDR C
2481  4916  40                          .BYTE   @100    ;3RD SCLK
2482  4917  81                          .BYTE   @201
2483  4918  01                          .BYTE   @1
2484  4919  1C                          .BYTE   @34     ;4TH SCLK
2485  491A  20                          .BYTE   @40
2486  491B  00                          .BYTE   @0
2487  491C  40                          .BYTE   @100    ;5TH SCLK
2488  491D  82                          .BYTE   @202
2489  491E  02                          .BYTE   2
2490  491F  2C                          .BYTE   @54     ;6TH SCLK (LAST)
2491  4920  30                          .BYTE   @60
2492  4921  00                          .BYTE   @0
2493                          ;TABLE OF EXPECTED DATA FOR 10-DUMP BYTE ASSY FOR A READ FWD FUNCTION
2494                          ;EVERY 'SCLK' ASSY TIME HAS 3 ENTRYS IN THE TAB_E TO REPRESENT THE
2495                          ;CONTENTS OF THE 'DDR' REGISTERS (DDR A THRU DDR C).
2496
2497                          ;THIS IS PACKING MODE #3 - READ FORWARD
2498
2499  4922  08                  FD10D:  .BYTE   @10     ;1ST SCLK       DDR A (2 COMPLETE 36. BIT)
2500  4923  04                          .BYTE   @4      ;               DDR B (WORDS OR 10 BYTES IN THE)
2501  4924  00                          .BYTE   @0      ;               DDR C (XFR. DATA = 1 TO 12(8))
2502  4925  14                          .BYTE   @24     ;2ND SCLK       DDR A
2503  4926  C1                          .BYTE   @301    ;               DDR B
2504  4927  00                          .BYTE   @0      ;               DDR C
2505  4928  1C                          .BYTE   @34     ;3RD SCLK
2506  4929  18                          .BYTE   @30
2507  492A  00                          .BYTE   @0
2508  492B  68                          .BYTE   @150    ;4TH SCLK
2509  492C  02                          .BYTE   @2
2510  492D  02                          .BYTE   @2
2511
2512                          ;READ REVERSE TRANSFER EXPECTED DATA TABLE
2513
2514  492E  84                  RD10D:  .BYTE   @204    ;1ST SCLK       DDR A (2 COMPLETE 36. BIT)
2515  492F  C0                          .BYTE   @300    ;               DDR B (WORDS OR 10 BYTES IN THE)
2516  4930  00                          .BYTE   @0      ;               DDR C (XFR. DATA = 12 TO 1(8))
2517  4931  10                          .BYTE   @20     ;2ND SCLK       DDR A
2518  4932  14                          .BYTE   @24     ;               DDR B
```

```
2519  4933  00                          .BYTE   @0      ;                       DDR C
2520  4934  D8                          .BYTE   @330    ;3RD SCLK
2521  4935  01                          .BYTE   @1
2522  4936  02                          .BYTE   @2
2523  4937  24                          .BYTE   @44     ;4TH SCLK
2524  4938  28                          .BYTE   @50
2525  4939  00                          .BYTE   @0
2526                              ;TABLE OF EXPECTED DATA FOR 10-HIGH DENSITY BYTE ASSY FOR A READ FWD FUNCTION
2527                              ;EVERY 'SCLK' ASSY TIME HAS 3 ENTRYS IN THE TABLE TO REPRESENT THE
2528                              ;CONTENTS OF THE 'DDR' REGISTERS (DDR A THRU DDR C).
2529
2530                              ;THIS IS PACKING MODE #4 (HDC) - READ FORWARD
2531                              ;ALSO PACKING MODE #6 (HDD) - READ FORWARD
2532
2533  493A                        FD10HC:
2534  493A  08                   FD10HD: .BYTE   @10     ;1ST SCLK       DDR A (2 COMPLETE 36. BIT)
2535  493B  04                           .BYTE   @4      ;               DDR B (WORDS OR 9 DATA BYTES)
2536  493C  00                           .BYTE   @0      ;               DDR C (IN XFR. DATA = 1 THRU)
2537  493D  00                           .BYTE   @0      ;2ND SCLK       DDR A (11(8))
2538  493E  C1                           .BYTE   @301    ;               DDR B
2539  493F  00                           .BYTE   @0      ;               DDR C
2540  4940  81                           .BYTE   @201    ;3RD SCLK
2541  4941  41                           .BYTE   @101
2542  4942  01                           .BYTE   @1
2543  4943  27                           .BYTE   @47     ;4TH & LAST SCLK
2544  4944  20                           .BYTE   @40
2545  4945  00                           .BYTE   @0
2546
2547                              ;THIS IS PACKING MODE #4 (HDC) - READ REVERSE TABLE OF EXPECTED DATA
2548                              ;ALSO PACKING MODE #6 (HDD) - READ REVERSE DATA TABLE
2549
2550  4946                        RD10HC:
2551  4946  07                   RD10HD: .BYTE   @7      ;1ST SCLK       DDR A (2 COMPLETE 36. BIT)
2552  4947  08                           .BYTE   @10     ;               DDR B (WORDS OR 9 DATA BYTES)
2553  4948  00                           .BYTE   @0      ;               DDR C (IN XFR. DATA = 11 THRU)
2554  4949  00                           .BYTE   @0      ;2ND SCLK       DDR A (1(8))
2555  494A  41                           .BYTE   @101    ;               DDR B
2556  494B  01                           .BYTE   @1      ;               DDR C
2557  494C  80                           .BYTE   @200    ;3RD SCLK
2558  494D  C1                           .BYTE   @301
2559  494E  01                           .BYTE   @1
2560  494F  20                           .BYTE   @40     ;4TH & LAST SCLK
2561  4950  24                           .BYTE   @44
2562  4951  00                           .BYTE   @0
2563                              ;TABLE OF EXPECTED DATA FOR IMAGE BYTE ASSY FOR A READ FWD FUNCTION
2564                              ;EVERY 'SCLK' ASSY TIME HAS 3 ENTRYS IN THE TABLE TO REPRESENT THE
2565                              ;CONTENTS OF THE 'DDR' REGISTERS (DDR A THRU DDR C).
2566
2567                              ;THIS IS PACKING MODE #5 - READ FORWARD AND READ REVERSE
2568
2569  4952                        RDIMAG:
2570  4952  04                   FDIMAG: .BYTE   @4      ;1ST SCLK       DDR A
2571  4953  00                           .BYTE   @0      ;               DDR B
2572  4954  00                           .BYTE   @0      ;               DDR C
```

```
2573  4955  08                .BYTE   @10     ;SCLK #2        DDR A
2574  4956  00                .BYTE   @0      ;               DDR B
2575  4957  00                .BYTE   @0      ;               DDR C
2576  4958  0C                .BYTE   @14     ;SCLK #3        DDR A
2577  4959  00                .BYTE   @0      ;               DDR B
2578  495A  00                .BYTE   @0      ;               DDR C
2579  495B  10                .BYTE   @20     ;SCLK #4        DDR A
2580  495C  00                .BYTE   @0      ;               DDR B
2581  495D  00                .BYTE   @0      ;               DDR C
2582  495E  14                .BYTE   @24     ;SCLK #5        DDR A
2583  495F  00                .BYTE   @0      ;               DDR B
2584  4960  00                .BYTE   @0      ;               DDR C
2585  4961  18                .BYTE   @30     ;SCLK #6        DDR A
2586  4962  00                .BYTE   @0      ;               DDR B
2587  4963  00                .BYTE   @0      ;               DDR C
2588  4964  1C                .BYTE   @34     ;SCLK #7        DDR A
2589  4965  00                .BYTE   @0      ;               DDR B
2590  4966  00                .BYTE   @0      ;               DDR C
2591  4967  20                .BYTE   @40     ;SCLK #10       DDR A
2592  4968  00                .BYTE   @0      ;               DDR B
2593  4969  00                .BYTE   @0      ;               DDR C
2594  496A  24                .BYTE   @44     ;SCLK #11       DDR A
2595  496B  00                .BYTE   @0      ;               DDR B
2596  496C  00                .BYTE   @0      ;               DDR C
2597  496D  28                .BYTE   @50     ;SCLK #12       DDR A
2598  496E  00                .BYTE   @0      ;               DDR B
2599  496F  00                .BYTE   @0      ;               DDR C
2600  4970  2C                .BYTE   @54     ;SCLK #13       DDR A
2601  4971  C0                .BYTE   @0      ;               DDR B
2602  4972  00                .BYTE   @0      ;               DDR C
2603  4973  30                .BYTE   @60     ;SCLK #14       DDR A
2604  4974  00                .BYTE   @0      ;               DDR B
2605  4975  00                .BYTE   @0      ;               DDR C
2606  4976  34                .BYTE   @64     ;SCLK #15       DDR A
2607  4977  00                .BYTE   @0      ;               DDR B
2608  4978  00                .BYTE   @0      ;               DDR C
2609  4979  38                .BYTE   @70     ;SCLK #16       DDR A
2610  497A  00                .BYTE   @0      ;               DDR B
2611  497B  00                .BYTE   @0      ;               DDR C
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS

                                        SSUB
                                        ;*************************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*CLEAR ALL TU PORTS
                                        SD
                                        ;*************************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                        ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                        ;*AND LOOP MODES.
                                        SP
                                        ;*************************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                        ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                        ;*     CLEAR PORT SELECT FOR TRANSPORT
                                        ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                        ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                        ;*     CLEAR PORT AMTIE WORD
                                        ;*ENDSUB
                                        S
                                        ;*************************************************************************
497C  F5                      12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
497D  C5                      12.0          PUSH    B
497E  06  00                   7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
4980  DB  E0                  10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
4982  E6  80                   7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
4984  B0                       4.0          ORA     B               ;ADD IN THE SELECTED PORT #
4985  D3  E0                  10.0          OUT     MBSEL           ;RESET TO THIS PORT
4987  3E  80                   7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
4989  D3  40                  10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
498B  AF                       4.0          XRA     A               ;CLEAR TU COMMAND A
498C  D3  40                  10.0          OUT     TCMD            ;
498E  3E  81                   7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
4990  D3  40                  10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
4992  3E  00                   7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
4994  D3  40                  10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
4996  AF                       4.0          XRA     A
4997  D3  44                  10.0          OUT     TAMT            ;CLEAR AMTIE WORD
4999  D3  48                  10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
499B  D3  4C                  10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
499D  04                       4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
499E  78                       4.0          MOV     A,B
499F  FE  04                   7.0          CPI     4               ;DONE?
49A1  C2  80  49              10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
49A4  C1                      10.0          POP     B               ;RESET B & C
49A5  F1                      10.0          POP     PSW             ;ALL DONE
49A6  C9                      10.0          RET                     ;EXIT
```

```
                                        .SBTTL  SUBROUTINE 4X5 TRANSLATE A SUBGROUP
49A7                                    SSUB
 (1)                                    ; ************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
                                        ;*4X5 TRANSLATE A SUBGROUP
49A7                                    SD
 (1)                                    ; ************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO 4X5 TRANSLATE A 4 BYTE DATA SUBGROUP INTO
                                        ;*A 5 BYTE SUBGROUP AS RECORDED ON TAPE OR READ FROM TAPE.
49A7                                    SP
 (1)                                    ; ************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNSUB
                                        ;*    GET THE 4 (INPUT) DATA BYTES TO BE TRANSLATED
                                        ;*    CLEAR THE TRANSLATED (OUTPUT) BUFFER (OUTPUT BUFFER IS 9X5)
                                        ;*    SET LOOP COUNT TO 9
                                        ;*    BGNDO
                                        ;*    :   CALCULATE ODD PARITY FOR THE FOUR (INPUT) DATA BYTES (INPUT BUFFER IS NOW 9X4
                                        ;*    :   ROTATE EACH (5) ENTRY OF THE TRANSLATED (OUTPUT) BUFFER RIGHT
                                        ;*    :   GET THE LEAST SIGNIFICANT BIT OF EACH (INPUT) DATA BYTE AND
                                        ;*    :       FORM A TABLE INDEX (MS BIT OF INDEX = LS BIT OF THE FIRST
                                        ;*    :       BYTE OF THE INPUT DATA, ETC.)
                                        ;*    :   REMOVE THE LS BIT OF EACH INPUT DATA BYTE (THE ONE JUST USED
                                        ;*    :       TO BUILD THE INDEX)
                                        ;*    :   USE THE 4 BIT INDEX TO GET THE TRANSLATION VALUE
                                        ;*    :   PUT ONE BIT OF THE TRANSLATION VALUE, AS THE MOST SIGNIFICANT
                                        ;*    :       BIT EACH OF THE 5 TRANSLATED DATA BYTES (LS BIT OF THE
                                        ;*    :       TRANSLATION VALUE REPLACES MS BIT OF THE FIRST TRANSLATED
                                        ;*    :       DATA BYTE.
                                        ;*    :   DECREMENT THE LOOP COUNT
                                        ;*    :   DO UNTIL THE LOOP COUNT = 0
                                        ;*    ENDDO
                                        ;*ENDSUB
49A7                                    S
 (1)                                    ; ************************************************************************
                                        
49A7   F5              12.0  T4X5:      PUSH    PSW             ;SAVE REGISTER A + PSW
49A8   C5              12.0             PUSH    B               ;SAVE REGISTER B + C
49A9   D5              12.0             PUSH    D               ;SAVE REGISTER D + E
49AA   E5              12.0             PUSH    H               ;SAVE REGISTER H + L
                                        ;FOUR DATA BYTES ARE NOW IS THE BUFFER TABLE
                                        ;CLEAR THE TRANSLATED DATA TABLE.
49AB   06   0A          7.0             MVI     B,10            ;SET UP LOOP COUNT
49AD   11   35   4A    10.0             LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
49B0   AF               4.0             XRA     A               ;CLEAR A
49B1   12               7.0  D4X5:      STAX    D               ;STORE IN THE TABLE
49B2   13               6.0             INX     D               ;UPDATE TABLE POINTER
49B3   05               4.0             DCR     B               ;DECREMENT LOOP COUNT
49B4   C2   B1   49    10.0             JNZ     D4X5            ;DO UNTIL LOOP COUNT=0
49B7   0E   09          7.0             MVI     C,9             ;SET UP TRACK COUNT
```

```
2702  49B9  06  04          7.0  B4X5:   MVI   B,4            ;SET UP BIT COUNT
2703  49BB  11  31  4A      10.0          LXI   D,TRNIN
2704  49BE  AF               4.0          XRA   A              ;CLEAR THE GROUP POSITION COUNT
2705  49BF  32  3F  4A      13.0          STA   GP4X5
2706  49C2  1A               7.0  C4X5:   LDAX  D              ;GET A DATA BYTE
2707  49C3  A7               4.0          ANA   A              ;SET CONDITION BITS
2708  49C4  E2  C8  49      10.0          JPO   PO4X5          ;ODD PARITY LEAVE CARRY CLEAR
2709  49C7  37               4.0          STC                  ;EVEN PARITY SET CARRY
2710  49C8  1F               4.0  PO4X5:  RAR                  ;SHIFT OUT DESIRED BIT
2711  49C9  12               7.0          STAX  D              ;STORE RESULT BACK IN TEMP TABLE
2712  49CA  3A  3F  4A      13.0          LDA   GP4X5          ;GET THE GROUP POSITION BYTE
2713  49CD  17               4.0          RAL                  ;PUT IN THIS DATA BIT
2714  49CE  32  3F  4A      13.0          STA   GP4X5          ;SAVE THE UPDATED GROUP POSITION BYTE
2715  49D1  13               6.0          INX   D              ;UPDATE THE TABLE POINTER
2716  49D2  05               4.0          DCR   B              ;DECREMENT THE BIT COUNT
2717  49D3  C2  C2  49      10.0          JNZ   C4X5           ;DO UNTIL ALL 4 BITS ARE RECEIVED
2718
2719                                ;AT THIS POINT GP4X5 CONTAINS 4 BITS - ONE FROM EACH OF THE FOUR BYTES
2720
2721                                ;PERFORM A DOUBLE PRECISION SHIFT ON THE TRANSLATED DATA
2722  49D6  06  05          7.0          MVI   B,5            ;SET UP LOOP COUNT
2723  49D8  11  35  4A      10.0          LXI   D,TRNOUT       ;GET POINTER TO TRANSLATED DATA TABLE
2724  49DB  13               6.0  E4X5:   INX   D              ;POINT TO PARITY BIT
2725  49DC  1A               7.0          LDAX  D              ;GET PARITY BIT
2726  49DD  A7               4.0          ANA   A              ;CLEAR THE CARRY BIT
2727  49DE  1F               4.0          RAR                  ;SHIFT TO THE CARRY BIT
2728  49DF  12               7.0          STAX  D              ;STORE IT BACK
2729  49E0  1B               6.0          DCX   D              ;DECREMENT TO DATA BITS
2730  49E1  1A               7.0          LDAX  D              ;GET THE DATA BITS
2731  49E2  1F               4.0          RAR                  ;SHIFT IN CARRY BITS
2732  49E3  12               7.0          STAX  D              ;STORE BACK IN TABLE
2733  49E4  13               6.0          INX   D              ;POINT TO NEXT TABLE ENTRY
2734  49E5  13               6.0          INX   D              ;
2735  49E6  05               4.0          DCR   B              ;DECREMENT LOOP COUNT
2736  49E7  C2  DB  49      10.0          JNZ   E4X5           ;DO UNTIL LOOP COUNT=0
2737  49EA  11  17  4A      10.0          LXI   D,TAB4X5       ;LOAD ADDRESS OF TRANSLATION TABLE
2738  49ED  26  00          7.0          MVI   H,0            ;CLEAR REGISTER H
2739  49EF  3A  3F  4A      13.0          LDA   GP4X5          ;GET GROUP POSITION COUNT
2740  49F2  6F               4.0          MOV   L,A
2741  49F3  19              10.0          DAD   D              ;ADD GROUP POSITION COUNT TO TABLE
2742  49F4  7E               7.0          MOV   A,M            ;GET TRANSLATED DATA
2743  49F5  32  40  4A      13.0          STA   TRNTMP
2744                                ;"TRNTMP" = THE TRANSLATED DATA
2745
2746  49F8  11  35  4A      10.0          LXI   D,TRNOUT       ;GET POINTER TO TRANSLATED DATA TABLE
2747  49FB  06  05          7.0          MVI   B,5            ;SET UP LOOP COUNT
2748  49FD  3A  40  4A      13.0  F4X5:   LDA   TRNTMP         ;GET TRANSLATED DATA
2749  4A00  13               6.0          INX   D              ;POINT TO PARITY POSITION
2750  4A01  A7               4.0          ANA   A              ;CLEAR CARRY BIT
2751  4A02  1F               4.0          RAR                  ;SHIFT OUT A BIT
2752  4A03  32  40  4A      13.0          STA   TRNTMP         ;STORE TRANSLATED DATA
2753  4A06  1A               7.0          LDAX  D              ;GET THE PARITY BIT (ALWAYS ZERO)
2754  4A07  17               4.0          RAL                  ;ROLL IN THE CARRY BIT
2755  4A08  12               7.0          STAX  D              ;STORE AS THE PARITY BIT
```

```
2756  4A09  13          6.0   INX   D      ;UPDATE TABLE POINTER
2757  4A0A  05          4.0   DCR   B      ;DECREMENT THE LOOP COUNT
2758  4A0B  C2  FD  49  10.0   JNZ   F4X5   ;DO UNTIL LOOP COUNT=0
2759
2760                           ;NOW THE TRANSLATED DATA TABLE HAS BEEN UPDATED
2761
2762  4A0E  0D          4.0   DCR   C      ;DECREMENT THE TRACK COUNT
2763  4A0F  C2  B9  49  10.0   JNZ   B4X5   ;DO UNTIL ALL TRACKS TRANSLATED
2764  4A12  E1          10.0   POP   H      ;RESTORE REGISTER H + L
2765  4A13  D1          10.0   POP   D      ;RESTORE REGISTER D + E
2766  4A14  C1          10.0   POP   B      ;RESTORE REGISTER B + C
2767  4A15  F1          10.0   POP   PSW    ;RESTORE REGISTER A + PSW
2768  4A16  C9          10.0   RET          ;RETURN TO USER
```

```
2770                                    .SBTTL  TABLE 4 X 5 TRANSLATION
2771
2772                                    ;THIS TABLE TRANSLATED DATA FOR AS FOLLOWS
2773                                    ;  INPUT GROUP POSITIONS              OUTPUT GROUP POSITIONS
2774                                    ;   1 2 3 4 / 5 6 7 8                  10 9 8 7 6 / 5 4 3 2 1
2775                                    ;
2776                                    ;       0000                                10011
2777                                    ;       0001                                11011
2778                                    ;       0010                                01001
2779                                    ;       0011                                11001
2780                                    ;       0100                                10111
2781                                    ;       0101                                10101
2782                                    ;       0110                                01101
2783                                    ;       0111                                11101
2784                                    ;       1000                                01011
2785                                    ;       1001                                10010
2786                                    ;       1010                                01010
2787                                    ;       1011                                11010
2788                                    ;       1100                                01111
2789                                    ;       1101                                10110
2790                                    ;       1110                                01110
2791                                    ;       1111                                11110
2792
2793  4A17  13                          TAB4X5: .BYTE   @23
2794  4A18  1B                                  .BYTE   @33
2795  4A19  09                                  .BYTE   @11
2796  4A1A  19                                  .BYTE   @31
2797  4A1B  17                                  .BYTE   @27
2798  4A1C  15                                  .BYTE   @25
2799  4A1D  0D                                  .BYTE   @15
2800  4A1E  1D                                  .BYTE   @35
2801  4A1F  0B                                  .BYTE   @13
2802  4A20  12                                  .BYTE   @22
2803  4A21  0A                                  .BYTE   @12
2804  4A22  1A                                  .BYTE   @32
2805  4A23  0F                                  .BYTE   @17
2806  4A24  16                                  .BYTE   @26
2807  4A25  0E                                  .BYTE   @16
2808  4A26  1E                                  .BYTE   @36
```

```
2810                                    .SBTIL  SUBROUTINE VARIABLES
2811
2812  4A27   00                         SDDRA:  .BYTE   0        ;STORAGE FOR DDR A DATA
2813  4A28   00                         SDDRB:  .BYTE   0        ;STORAGE FOR DDR B DATA
2814  4A29   00                         SDDRC:  .BYTE   0        ;STORAGE FOR DDR C DATA
2815  4A2A   00                         SCLKCT: .BYTE   0        ;COUNT OF 'SCLK' RECEIVED
2816  4A2B   00                         ESCLK:  .BYTE   0        ;NUMBER OF EXPECTED SCLK PER PASS
2817  4A2C   00                         FMTNUM: .BYTE   0        ;SLEECTED PACKING MODE PER PASS
2818  4A2D   00                         UNITMP: .BYTE   0        ;PORT NUMBER TO USE AS RESPONDED BY HOST
2819  4A2E   00                         SUNIT:  .BYTE   0        ;STORAGE OF THE SELECTED UNIT NUMBER
2820  4A2F   00                         REVERSE: .BYTE  0        ;READ FWD/REV FLAG ... =0 FOR READ FWD
2821  4A30   00                         CCTLWD: .BYTE   0        ;LOAD INTO THE SYSTEM CLOCK CONTROL WORD
2822
2823  4A31 0004                         TRNIN:  .BLKB   4        ;TABLE CONTAINING THE 4 DATA BYTES TO
2824                                                             ;BE TRANSLATED
2825  4A35 000A                         TRNOUT: .BLKB   10       ;TRANSLATED DATA TABLE DATA AFTER 4X5
2826                                                             ;TRANSLATION
2827  4A3F   00                         GP4X5:  .BYTE   0        ;ACCUMULATOR FOR THE 4 BIT SUBGROUP TC
2828                                                             ;BE TRANSLATED
2829  4A40   00                         TRNTMP: .BYTE   0        ;TEMP. STORAGE FOR THE TRANSLATED SUBGROUP.
2830
2831  4A41   00                         MOREFG: .BYTE   0        ;FLAG FOR LOADING AN ADDITIONAL DATA GRP
```

```
                                        .SBTTL  SUBROUTINE CLEAR ECC
2834  4A42                              $
 (1)                                    ; ************************************************************************
2835                                    ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION "ECCCHR" -
2836                                    ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
2837                                    ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
2838  4A42                              $
 (1)                                    ; ************************************************************************
2839
2840  4A42  AF              4.0 CLECC:  XRA     A               ;CLEAR THE ACCUMULATOR
2841  4A43  32  6E  4A     13.0        STA     ECCCHR          ;CLEAR THE ECC CHARACTER
2842  4A46  C9              10.0        RET                     ;RETURN TO USER
```

```
                                .SBTTL  SUBROUTINE CALCUALTE ECC CHARACTER
4A47                            S
 (1)                            ; ************************************************************************
                                ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
                                ;THE CONTENTS OF "ECCCHR" TO UPDATE THE CONTENTS OF "ECCCHR" ACCORDING
                                ;TO THE ANSI STANDARD ECC POLYNOMIAL.
4A47                            S
 (1)                            ; ************************************************************************

4A47  E5              12.0  ECC:    PUSH    H
4A48  C5              12.0          PUSH    B
4A49  21  6E  4A      10.0          LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
4A4C  AE               7.0          XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
4A4D  5F               4.0          MOV     E,A             ;SAVE XOR RESULT IN E
4A4E  E6  10           7.0          ANI     $10             ;IS BIT #4 OF RESULT SET
4A50  7B               4.0          MOV     A,E             ;RESTORE XOR RESULT FROM B
4A51  CA  56  4A      10.0          JZ      ECC1            ;CONTINUE IF BIT #4 RESET
4A54  EE  23           7.0          XRI     $23             ;ELSE-XOR WITH 23
4A56  5F               4.0  ECC1:    MOV     E,A             ;STORE THE ECC RESULT IN E
4A57  AF               4.0          XRA     A               ;CLEAR A
4A58  4F               4.0          MOV     C,A             ;CLEAR THE TRANSLATE RESULT
4A59  21  66  4A      10.0          LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
4A5C  CD  6F  4A      18.0          CALL    TRANS           ;TRANSLATE THE BITS
4A5F  79               4.0          MOV     A,C             ;GET THE TRANSLATED RESULT
4A60  32  6E  4A      13.0          STA     ECCCHR          ;STORE RESULT
4A63  C1              10.0          POP     B
4A64  E1              10.0          POP     H
4A65  C9              10.0          RET

4A66  08                      ECCTBL: $08                     ;BIT 0 = POSITION 3
4A67  20                              $20                     ,BIT 1 = POSITION 5
4A68  02                              $02                     ;BIT 2 = POSITION 1
4A69  40                              $40                     ;BIT 3 = POSITION 6
4A6A  80                              $80                     ;BIT 4 = POSITION 7
4A6B  01                              $01                     ;BIT 5 = POSITION 0
4A6C  10                              $10                     ;BIT 6 = POSITION 4
4A6D  04                              $04                     ;BIT 7 = POSITION 2

4A6E  00                      ECCCHR: .BYTE   0
```

```
                                        .SBTTL  SUBROUTINE POLYNOMIAL BIT TRANSLATION

                                        ;
                                        ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
                                        ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
                                        ;
4A6F  06 01           7.0  TRANS:  MVI   B,1            ;INIT 'B' TO BIT POSITION 0
4A71  7B              4.0  TRANS1: MOV   A,E            ;GET CHAR TO BE TRANSLATED
4A72  A0              4.0          ANA   B              ;SEE IF BIT POSITION IN 'B' IS SET
4A73  CA 79 4A       10.0          JZ    TRANS2         ;DO NEXT BIT POSITION IF NOT SET
4A76  79              4.0          MOV   A,C            ;GET PREVIOUS RESULT OF 'OR'
4A77  B6              7.0          ORA   M              ;'OR' IN NEW POSITION
4A78  4F              4.0          MOV   C,A            ;SAVE RESULT

4A79  78              4.0  TRANS2: MOV   A,B
4A7A  07              4.0          RLC                  ;POSITION MASK TO NEXT BIT
4A7B  47              4.0          MOV   B,A
4A7C  D8             12.0          RC                   ;EXIT WHEN ALL POSITIONS DONE
4A7D  23              6.0          INX   H              ;POINT TO NEXT TABLE ENTRY
4A7E  C3 71 4A       10.0          JMP   TRANS1         ;PROCESS NEXT BIT

4A81                                S
 (1)                                ; ***************************************************************************
                                    ;CLKSYS - ROUTINE TO CLOCK THE SYSTEM USING THE CONTENTS OF 'CCTLWD'
                                    ;       AS A BASE AND TOGGLING THE CLOCK BIT IN THE CLOCK CONTROL WORD
4A81                                S
 (1)                                ; ***************************************************************************

4A81  F5             12.0  CLKSYS: PUSH  PSW
4A82  3A 30 4A       13.0          LDA   CCTLWD         ;GET THE CONTROL WORD BASE
4A85  F6 40           7.0          ORI   SSCLK          ;GET THE CLOCK BIT
4A87  D3 F0          10.0          OUT   CLKCTL         ;SET THE CLOCK ON
4A89  E6 BF           7.0          ANI   @277           ;CLEAR THE CLOCK BIT
4A8B  D3 F0          10.0          OUT   CLKCTL         ;FINISH THE CLOCK CYCLE
4A8D  F1             10.0          POP   PSW
4A8E  C9             10.0          RET
0000                                .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE   4F9B           ATTCD   4F97          AXNUM   4F91
B      =%0000          BADST  = 0090          BIT0   = 0001          BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008          BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080          BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00           BSAVE   4F9C           BYTCNT= 00D4          BYTEH   4F24
BYTEL   4F23           B4X5    49B9           C      =%0001          CASCT   4F21
CASCTL= 00A0           CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088
CBUSST= 00A1           CBYTH  = 008B          CBYTL  = 008A          CCTLWD  4A30
CDG1H  = 0087          CDG1L  = 0086          CDG2H  = 0093          CDG2L  = 0092
CDG3H  = 0095          CDG3L  = 0094          CDVTH  = 008D          CDVTL  = 008C
CHPTIE= 0028           CH0TIE= 0020           CH1TIE= 0021          CH2TIE= 0022
CH3TIE= 0023           CH4TIE= 0024           CH5TIE= 0025          CH6TIE= 0026
CH7TIE= 0027           CKLOP  = 2815          CLEAR   497C          CLECC   4A42
CLKCTL= 00F0           CLKRS   43E6           CLKRS2  462C          CLKSYS  4A81
CLOCK   4F26           CLRLP   4980           CMC0H  = 0099          CMC0L  = 0098
CMC1H  = 009B          CMC1L  = 009A          CMC2H  = 009D          CMC2L  = 009C
CMC3H  = 009F          CMC3L  = 009E          CMINH  = 0097          CMINL  = 0096
CNTCTL= 00D7           CNV1    4849           CONVRT  4843          CRCG    467F
CRCGEC  4686           CRCG2   4683           CRCWRD= 0018          CSAVE   4F9D
CSRLH  = 0091          CSRLL  = 0090          CTCH   = 0085          CTCL   = 0084
CTSTH  = 008F          CTSTL  = 008E          CXCTH  = 0081          CXCTL  = 0080
CXINH  = 0083          CXINL  = 0082          C.     = 0001          C.AVAI= 0080
C.DP   = 0008          C.DSE  = 0010          C.DTU  = 0003          C.DVA  = 0008
C.FAIL= 00FC           C.FMT  = 0070          C.FNCT= 003E          C.GO   = 0001
C.INTC= 00FE           C.MAIN= 0020           C.NSA  = 0080          C.RCT  = 00FC
C.SER  = 0080          C.SHR  = 0040          C.SKPC= 000F          C.TAPE= 0040
C.WCS  = 0002          C1EC1   43F1           C1EC1L  43F9          C1EC2   4408
C1EC2L  4410           C2EC1   4637           C2EC1L  463F          C2EC2   464E
C2EC2L  4656           C2EC3   4665           C2EC3L  466D          C4X5    49C2
D      =%0002          DATACT= 00D0           DBUS    4F28          DBUSCT= 00C0
DBUSST= 00C0           DDRA   = 00D8          DDRAIN= 0010          DDRB   = 00D9
DDRBIN= 0002           DDRC   = 00DA          DDRCIN= 0001          DDRCO  = 0088
DDRCTL= 00DB           DG1     488F           DG1ECC  4896          DG2     4897
DG2ECC  489E           DIAFLG  4F22           DIAGPG= 4300          DIAGRM= 4F90
DIARD  = 000B          DONE1  = 0045          DONINT= 0010          DSAVE   4F9E
DSE    = 0006          DUMM1   447E  G        DUMM2   449E  G       DUMM3   4722  G
DUMM4   4758  G        D.ATH0= 0001           D.ATH1= 0002          D.EOTD= 0010
D.LAGC= 0020           D.NOTW= 0040           D.NTHR= 0004          D.TACH= 0008
D.WR4  = 0080          D4X5    49B1           E      =%0003          ECC     4A47
ECCBAD= 0042           ECCCHR  4A6E           ECCCOR= 0019          ECCOK  = 0041
ECCSTA= 001A           ECCTBL  4A66           ECCTST= 000E          ECC1    4A56
EDATA  = 0095          EDFWD   48AF           EDREV   48BD          EOTCLR= 0003
ERFLG   4F93           ERLP   = 2809          ERLPA  = 280F          ERLPB  = 2812
ERLPE  = 280C          ERNUM   4F90           ERRCNT= 00D6          ESAVE   4F9F
ESCLK   4A2B           ESCLKT  48CB           EXIT    456F          EXIT2   4829
EXPADR  48D2           E.ACRC= 0010           E.AMT  = 0020          E.CDF  = 0080
E.CRC  = 0080          E.PNTR= 0008           E.RPE  = 0040          E.STEC= 0001
E.TTEC= 0002           E.UNC  = 0004          E4X5    49DB          FDIMAG  4952
FD10C   48FE           FD10D   4922           FD10HC  493A          FD10HD  493A
FD11N   48D4           FD15N   48E9           FIFORD= 006A          FLOAD   485F
FMTNUM  4A2C           FORMAT  4F25           FOUND   4327          FPORT   4347
FWDTST= 0061           F4X5    49FD           GCRID  = 0089          GCRSET= 0002
```

```
GOODTM= 0092        GP4X5   4A3F        H      =%0004       HLSAVE  4FA0
IE     = 0008       INTSTA= 00E0        ITERA   4F9A        I.PWR = 0020
I.RMPE= 0040        I5.5  = 0010        I6.5  = 0020        I7.5  = 0040
KCALL = 005F        KCLR  = 007B        KDEP  = 003F        KENAB = 0078
KEXAM = 003E        KEYBRD= 00C8        KEY1  = 0078        KEY10 = 006D
KEY11 = 006E        KEY12 = 006F        KEY13 = 005C        KEY14 = 005D
KEY15 = 005E        KEY16 = 005F        KEY17 = 003C        KEY18 = 003D
KEY19 = 003E        KEY2  = 0079        KEY20 = 003F        KEY3  = 007A
KEY4  = 007B        KEY5  = 0074        KEY6  = 0075        KEY7  = 0076
KEY8  = 0077        KEY9  = 006C        KINTA = 006F        KLDAD = 003D
KN0   = 003C        KN1   = 005C        KN2   = 005D        KN3   = 005E
KN4   = 006C        KN5   = 006D        KN6   = 006E        KN7   = 0074
KN8   = 0075        KN9   = 0076        KU2   = 0079        KU3   = 007A
KU8   = 0077        L      =%0005       LBLANK= 000F        LCE   = 000B
LCH   = 000C        LCL   = 000D        LCP   = 000E        LC0   = 0000
LC1   = 0001        LC2   = 0002        LC3   = 0003        LC4   = 0004
LC5   = 0005        LC6   = 0006        LC7   = 0007        LC8   = 0008
LC9   = 0009        LDFIFO  4864        LDLEDA= 00CA        LDLEDB= 00CB
LDLEDC= 00CC        LDLEDD= 00CD        LDLEDE= 00CE        LDLEDF= 00CF
LKDIAG= 2800        LKKBD = 004C        LKKEY = 0049        LKLWMG= 0058
LKLWMP= 0055        LKLWPG= 0052        LKLWPP= 004F        LKMOD7= 0046
LKOPR = 0046        LOADIT  4858        LPFLG   4F94        LPNUM   4F92
M      =%0006       MBSEL = 00E0        MB.A  = 0008        MB.B  = 0004
MEMTOP= 4FFF        MINUS = 000A        MM    = 8000        MOREFG  4A41
MSE   = 0008        MSGN  = 000B        MTACLR= 0000        MT.ARA= 0020
MT.CPE= 0080        MT.DSE= 0001        MT.FWD= 0040        MT.INH= 0008
MT.LWR= 0004        MT.MOT= 0002        MT.NWT= 0080        MT.PEC= 0040
MT.PSB= 0004        MT.PS0= 0001        MT.PS1= 0002        MT.REV= 0020
MT.WRT= 0010        MT.Z  = 0008        M.ATA = 0080        M.CAPE= 0020
M.CONT= 0080        M.DEM = 0020        M.EBL = 0010        M.EXC = 0008
M.FAIL= 0008        M.ILR = 0010        M.INIT= 0010        M.OCC = 0020
M.ONLI= 0001        M.PE  = 0040        M.PORT= 0080        M.RDEN= 0002
M.RDPE= 0008        M.RUN = 0004        M.SCLK= 0001        M.TRA = 0040
M.UNIT= 0007        M.WCLK= 0040        M.WCLN= 0080        M.WREN= 0080
M5.5  = 0001        M6.5  = 0002        M7.5  = 0004        NOTCAP= 0088
NOUNIT  431E  G     OKAY  = 00FF        OPRRAM= 4300        OPSTRT= 0058
OPVER = 0040        PADCNT= 00D5        PADCRC= 0080        PDIAG = 0048
PEID  = 008A        PENAB = 004C        PESET = 0001        PL    = 00B1
PO4X5   49C8        PRDD  = 004C        PRENF = 009C        PS    = 00B2
PSTAT = 0048        PSW    =%0009       P.AMTP= 0001        P.BCTC= 0040
P.CMDP= 0020        P.INTE= 0080        P.LCS = 0040        P.LWR = 0020
P.RDP = 0002        P.RPEN= 0004        P.RPST= 0002        P.RPOE= 0020
P.RP1E= 0010        P.RP2E= 0020        P.RP3E= 0010        P.SING= 0080
P.STAT= 0002        P.STPE= 0080        P.TACH= 0008        P.TUPR= 0010
P.WCSP= 0004        P.WDS = 0040        P.WFLP= 0001        P.WPEN= 0010
P.WPOE= 0008        P.WP1E= 0004        P.WP2E= 0008        P.WP3E= 0004
P.5VOK= 0002        QUE    = 281B       QUEM  = 281E        RAMT  = 0010
RARA  = 0006        RARAI = 0004        RCHBD0= 0048        RCHBD1= 0047
RCHOK = 0046        RCHTST= 000C        RCLRT = 000D        RCMD  = 000B
RCMLP = 0003        RCONT = 0080        RDATA = 0017        RDCLK = 0010
RDG1    489F        RDG1EC  48A6        RDG2    48A7        RDG2EC  48AE
RDIMAG  4952        RDON  = 0011        RD10C   4910        RD10D   492E
RD10HC  4946        RD10HD  4946        RD11N   48E9        RD15N   48D4
READG = 0007        REND  = 0014        REQST = 2806        RESCHR= 00D1
```

```
RESG    467F          RESG2   4683          REVERS  4A2F          REVTST= 0064
REWIND= 0004          RFIFOL= 0008          RGCLK = 0002          RGCRI = 0003
RIBG  = 0001          RILL  = 0012          RINST = 000C          RLFIFO  487B
RLOAD   4876          RMCTST= 0008          RMK2  = 0013          RNOP  = 0000
RPATH = 0001          RPBAD = 0044          RPCHI = 0001          RPCLK = 0003
RPCTL = 0009          RPEI  = 0002          RPFAIL= 0000          RPF1  = 009D
RPF2  = 009E          RPOK  = 0043          RPOSTN= 0016          RPSTA = 0015
RRCMT = 000A          RSG2    48A3          RSG4    48AB          RSTAT = 0002
RTIEB = 000A          RTIER = 0030          RTM   = 0005          RUNKI = 0009
RUPTST= 005E          RWDUNL= 0005          R.AMT = 0001          R.BOP = 0008
R.DATA= 0040          R.DON = 0002          R.DRDY= 0010          R.END = 0010
R.ILL = 0004          R.JVOK= 0004          R.MK2 = 0008          R.PLOD= 0008
R.PLOO= 0010          R.PLO1= 0020          R.POST= 0020          R.STNM= 0002
R.STOP= 0004          R.STPC= 0001          R.TBJN= 0080          R.TSTD= 0040
R.VOK = 0080          R00H  = 0081          R00L  = 0080          R01H  = 0083
R01L  = 0082          R02H  = 0085          R02L  = 0084          R03H  = 0087
R03L  = 0086          R04H  = 0089          R04L  = 0088          R05H  = 008B
R05L  = 008A          R06H  = 008D          R06L  = 008C          R07H  = 008F
R07L  = 008E          R10H  = 0091          R10L  = 0090          R11H  = 0093
R11L  = 0092          R12H  = 0095          R12L  = 0094          R13H  = 0097
R13L  = 0096          R14H  = 0099          R14L  = 0098          R15H  = 009B
R15L  = 009A          R16H  = 009D          R16L  = 009C          R17H  = 009F
R17L  = 009E          R7.5  = 0010          SCLKCT  4A2A          SDDRA   4A27
SDDRB   4A28          SDDRC   4A29          SELCLR= 0000          SETATA= 00A1
SG2     4893          SG4     489B          SID   = 0080          SOD   = 0080
SOE   = 0040          SP    =%0008          SSCLK = 0040          SSTEP = 0005
STACK = 4FFF          STATRM= 4F20          STPCT   4F20          STRSP = 5000
SUNIT   4A2E          TAB4X5  4A17          TADR00= 0080          TADR01= 0081
TADR02= 0082          TADR03= 0083          TADR04= 0084          TADR05= 0085
TADR06= 0086          TADR07= 0087          TADR10= 0088          TADR11= 0089
TADR12= 008A          TADR13= 008B          TAMT  = 0044          TASEL = 0080
TCMD  = 0040          TC.FWD= 0040          TC.INH= 0008          TC.LWR= 0004
TC.REV= 0020          TC.WRT= 0010          TEMP    4F99          TEST1   4300
TEST2   4586          TMF   = 0099          TMRDY = 0040          TRANS   4A6F
TRANS1  4A71          TRANS2  4A79          TRKENA= 00D2          TRNIN   4A31
TRNOUT  4A35          TRNTMP  4A40          TSET  = 2803          TSTEND= 2818
TSTS  = 0040          TUSEL0= 00D1          TUSEL1= 00D2          TU78  = 0010
T.ATH0= 0001          T.ATH1= 0002          T.BOT = 0004          T.EOT = 0002
T.FPT = 0001          T.NTHR= 0004          T.ONL = 0020          T.PES = 0008
T.PSBJ= 0020          T.PSOJ= 0008          T.PS1J= 0010          T.RDY = 0080
T.RDYO= 0040          T.RWD = 0010          T.SCLK= 0002          T1CKA   44D1
T1CKAC  44FA  G       T1CKB   4500          T1CKBC  4514  G       T1CKC   451A
T1CKCC  452E  G       T1CLK   4469          T1CLKA  4463          T1LOOP  4370
T1PLO   441F          T1PM0   4568          T1PM6   455B          T1RF    4388
T1SCKL  4489          T1SCK0  44A7          T1SCLK  4487          T1SG1   4437
T1SG2   4440          T1SG3   4449          T1SG4   4452          T1ST    43EF
T2CKA   478B          T2CKAC  47B4  G       T2CKB   47BA          T2CKBC  47CE  G
T2CKC   47D4          T2CKCC  47E8  G       T2CLK   47CD          T2CLKA  4707
T2EMK   46C1          T2EMKL  46C6          T2LOOP  45B6          T2LP    458B
T2PLO   4687          T2PM0   4822          T2PM6   4815          T2RR    45CE
T2SCKL  4743          T2SCK0  4761          T2SCLK  472B          T2SGD1  46DE
T2SGD2  46D5          T2SGD3  46F0          T2SGD4  46E7          T2SG1   469D
T2SG2   46A6          T2SG3   46AF          T2SG4   46B8          T2ST    4635
T4X5    49A7          UIBG  = 00A1          UNITMP  4A2D          VALFC   4F98
```

```
VALTB   4F95            VELTST= 005B            WDR.P = 0010            WMCCTL= 00D3
WMCERR= 00DA            WMCSTA= 00D0            WRTCLK= 0000            WRTDAT= 00D3
W.ACRC= 0004            W.CRC = 0008            W.DIAG= 0002            W.DONN= 0040
W.ECC = 0010            W.ENAB= 0080            W.ERR = 0020            W.FMT = 0070
W.GCR = 0010            W.LEFT= 0004            W.ONES= 0020            W.RESI= 0002
W.REV = 0004            W.ROME= 0010            W.RST = 0001            W.SKIP= 000F
W.WRIT= 0008            W.XFER= 0020            X     =%000A            X.DONN= 0080
X.ENAB= 0040            X.PEPE= 0002            X.ROME= 0001            X.WCLK= 0001
Y     =%000B            .     = 4A8F


 ERRORS DETECTED:  0

 *WMC6.A78/PTP,WMC6=NLIST,PARAM,MACRO,LIST,WMC6
 RUN-TIME: 5 8 0 SECONDS
 CORE USED:  10K
```

     3   "MEMORY MAP" EXTENDED ADDRESS BIT
    19   READ MICRO CONTROLLER CONTROL & STATUS WORDS
   139   TRANSPORT/PORT COMMAND + STATUS WORDS
   380   COMMON ADDRESS SPACE REGISTERS
   444   MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
   485   MASSBUS CONTROL WORD DEFINITIONS
   557   WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
   709   KEYBOARD/DISPLAY CONTROL + STATUS WORD
   797   MISCELLANEOUS VARIABLE DEFINITIONS
   801   RAM ADDRESS DEFINITIONS
   819   LINKAGE TABLE ADDRESSES
   842   BIT DEFINITIONS
   856   RIM AND SIM INSTRUCTION BIT DEFINITIONS
   877   INTERRUPTABLE INSTRUCTION MACROS
   924   DIAGNOSTIC VARIABLES
   945   COMMON VARIABLE STORAGE
   959   MACRO CALL DEFINITIONS
  1047   TM78 MICRO-DIAGNOSTIC MACRO CALLS
  1101   8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
  1333   TEST 1 - WMC BYTE ASSEMBLY - READ FORWARD
  2155   SUBROUTINE CLEAR ALL TU PORTS
  2199   SUBROUTINE 4X5 TRANSLATE A SUBGROUP
  2311   TABLE 4 X 5 TRANSLATION
  2351   SUBROUTINE VARIABLES
  2378   SUBROUTINE CLEAR ECC
  2389   SUBROUTINE CALCUALTE ECC CHARACTER
  2427   SUBROUTINE POLYNOMIAL BIT TRANSLATION

```
                ;*************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;                "BYTE/SCLK COUNT NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                MM = DATA FORMAT FROM CAS REGISTER 2
                ;                NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;                BYTE-SCLK COUNT = LLL
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                LLL = AS ABOVE
                ;                MM  = AS ABOVE
                ;                NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;                TRANSITION COUNT = LLL
                ;                WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;                EXPECTED 18 BITS =      E EEEEEE
                ;                ACTUAL 18 BITS =        A AAAAAA
                ;
                ;                WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;                BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;                OF CAS REG 15 LOW BYTE.
                ;
                ;                ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;                AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;                17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;                TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;                AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;                "SUBGROUP NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;*************************************************************************
```

```
1016                    ; ************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ; ************************************************************************
1035                    ; ************************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ; ************************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  WMC7 - BYTE ASSEMBLY/ZERO FILL OF READ PATH DATA
                                .SBTTL  TEST 1 - WMC BYTE ASSEMBLY - READ FORWARD
                                ;ID     WMC7-WRITE MICRO CONTROLLER PART #7
                                .MACRO  XFRP    DATADR,FMT,BCT,SCLK
                                        .ADDR   DATADR  ;ADDRESS OF EXPECTED DATA TABLE
                                        .BYTE   FMT     ;PACKING MODE FMT
                                        .BYTE   BCT     ;BYTE COUNT THIS XFR
                                        .BYTE   SCLK    ;SCLK SCLKS EXPECTED THIS XFR
                                .ENDM   XFRP
4300                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*WMC BYTE ASSEMBLY/ZERO FILL TEST FROM READ FORWARD DATA
4300                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST WILL INJECT DATA THROUGH THE READ PATH FIFO'S AND CLOCK THE
                                ;*SYSTEM UNTIL AN 'SCLK' APPEARS TO SIGNAL 18 BITS OF DATA READY FOR
                                ;*TRANSMISSION ACROSS THE MASSBUS. EACH 'SCLK' TIME WILL HAVE THE 'DDR'
                                ;*REGISTERS SETUP WITH THE ASSEMBLED DATA AS COMMANDED BY THE PACKING
                                ;*MODE SELECTED IN THE TEST. ALL PACKING MODES ARE TESTED (I.E. 11-NORM
                                ;*TO 10-HI DEN COMPAT) IN A READ FORWARD DATA TRANSFER. THE TEST WILL
                                ;*CLOCK THE SYSTEM ONLY UNTIL THE EXPECTED AMOUNT OF 'SCLK' DATA HAS
                                ;*BEEN TESTED TO VERIFY THAT THE SELECTED PACKING MODE IS WORKING AS
                                ;*EXPECTED. THE READ PATH WILL BE FILLED WITH DATA TO SIMULATE SHORT DATA
                                ;*DATA TRANSFERS (ONLY RESIDUAL DATA GROUPS - XFR LENGTH FROM 1 - 6 BYTES).
                                ;*THIS TEST WILL VERIFY THAT THE WMC CAN SEND THE PROPER AMOUNT OF 'PAD'
                                ;*BYTES TO FILL AN 'SCLK' TRANSFER OF 18 BITS DATA TO THE HOST IN THE
                                ;*SELECTED DATA PACKING MODE.
4300                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  REQUEST A IU PORT FROM 'HOST' CPU
                                ;*  IF NO TU PORT SELECTED FROM 'HOST' CPU
                                ;*  :  THEN-REPORT OPERATOR ERROR
                                ;*  :  ELSE-PROCEED
                                ;*  ENDIF
                                ;*  BGNDO
                                ;*  :  CLEAR ALL PORTS
                                ;*  :  INIT THE SYSTEM
                                ;*  :  SET READ PATH TO NORMAL CLOCK SPEED
                                ;*  :  INIT THE PARAMETER TABLE POINTER - PTABLE
                                ;*  :  SET THE BYTE COUNT & DATA PACKING FORMAT FROM 'PTABLE' DATA
                                ;*  :  INIT COUNTERS + CONTROLS
                                ;*  :  LOAD BYTE COUNTER WITH EXPECTED # BYTES FROM READ PATH
                                ;*  :  RESTART WRITE MICROCONTROLLER
                                ;*  :  CLOCK WMC THROUGH RESTART CODE
                                ;*  :  START 'GCR' READ DATA XFR
                                ;*  :  CALCULATE ALL NECESSARY ECC CHARACTERS FOR INJECTION
                                ;*  :  SET READ PATH PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
```

```
                                ;*  :  CLOCK THE FIFO'S
                                ;*  :  ISSUE 'CLEAR ALL' TO READ PATH
                                ;*  :  WAIT
                                ;*  :  STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
                                ;*  :  CONVERT SUBGROUPS TO 4X5 & LOAD INTO FIFO
                                ;*      (END-MARK, RESIDUAL GRP  CRC GRP, MARK-2)
                                ;*  :  SETUP A WATCHDOG TIMER
                                ;*  :  CLOCK SYSTEM TO XFR DATA FROM FIFO TO WMC BYTE ASSY LOGIC
                                ;*  :  IF WATCHDOG TIMER=0
                                ;*  :  :  THEN-REPORT NO 'SCLK'
                                ;*  :  :  ELSE-CONT
                                ;*  :  ENDIF
                                ;*  :  CLOCK SYSTEM TILL SCLK=0...DATA READY TO BE TESTED
                                ;*  :  IF SCLK STILL UP AFTER 5000. CLOCKS
                                ;*  :  :  THEN - CHECK BYTE COUNTER TO SEE IF =0
                                ;*  :  :  :  IF BYTE COUNTER =0
                                ;*  :  :  :  :  THEN - DROP WMC ENABLE
                                ;*  :  :  :  :  ELSE - CONTINUE
                                ;*  :  :  :  ENDIF
                                ;*  :  :  ELSE - REPORT ERROR ... SCLK WON'T CLEAR AFTER SETTING
                                ;*  :  ENDIF
                                ;*  :  GET 'SCLK' 18 BIT DATA FROM BYTE ASSY INTO 'DDR' REGISTER
                                ;*  :  IF 18 BIT DATA SAME AS EXPECTED
                                ;*  :  :  THEN-PROCEED
                                ;*  :  :  ELSE-REPORT BYTE ASSY ERROR + WHAT WAS ASSEMBLED
                                ;*  :  ENDIF
                                ;*  :  POINT TO NEXT ENTRY IN 'PTABLE' STORAGE
                                ;*  :  IF ALL DONE WITH DATA TRANSFERS IN 'PTABLE'
                                ;*  :  :  THEN - DECREMENT THE ITERATION COUNTER
                                ;*  :  :  :    IF ALL ITERATIONS FINISHED
                                ;*  :  :  :    :  THEN-EXIT
                                ;*  :  :  :    :  ELSE-CONTINUE
                                ;*  :  :  :    ENDIF
                                ;*  :  :  :  ELSE-CONTINUE
                                ;*  :  :  ENDIF
                                ;*  :  :  DO UNTIL ALL ITERATIONS EXECUTED
                                ;*  :  ENDDO
                                ;*ENDTS*
1417  4300                      SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1418                            ;*WMC7 MICRO TEST 01
1419                            ;*WMC7 MICRO ERROR 01
1420                            ;*WMC BYTE ASSY FOR READ FORWARD XFR
1421                            ;*M8959 (WMC) & ALL OTHER BOARDS
1422                            ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
1423                            ;*FATAL ERROR - TEST ABORTED
1424                            ;*
1425                            ;*WMC7 MICRO TEST 01
1426                            ;*WMC7 MICRO ERROR 02
1427                            ;*WMC BYTE ASSY FOR READ FORWARD XFR
1428                            ;*M8959 (WMC) & ALL OTHER BOARDS
1429                            ;*TIMEOUT WHILE WAITING FOR 'SCLK' TO SET ... DATA NOT READY FROM BYTE ASSY
```

```
                                        ;*DATA HAS BEEN SENT TO WMC FROM CRC BOARD - 'XFER' SEEN IN WMCSTA WORD
                                        ;*
                                        ;*WMC7 MICRO TEST 01
                                        ;*WMC7 MICRO ERROR 03
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*TIMEOUT WHILE WAITING FOR 'SCLK' TO SET IN DBUSSTA WORD ... DATA NOT READY
                                        ;*AFTER ISSUEING A READ FORWARD COMMAND TO THE TM78 - NO 'XFER' SEEN IN
                                        ;*WMCSTA WORD. DATA NOT SEEN FROM CRC LOGIC.
                                        ;*
                                        ;*WMC7 MICRO TEST 01
                                        ;*WMC7 MICRO ERROR 04
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*TIMEOUT WHILE WAITING FOR 'SCLK' TO CLEAR AFTER SETTING
                                        ;*DURING A READ FORWARD COMMAND
                                        ;*
                                        ;*WMC7 MICRO TEST 01
                                        ;*WMC7 MICRO ERROR 05
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 7-0
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*WMC7 MICRO TEST 01
                                        ;*WMC7 MICRO ERROR 06
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 15-8
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*WMC7 MICRO TEST 01
                                        ;*WMC7 MICRO ERROR 07
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 17-16
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
4300                                    $
 (1)                                    ; ************************************************************************************
                                        
4300                                    TEST1:  TESTX   @1
 (1)  4300  3E  01            7.0                MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                CALL    TSET            ;SETUP THE TEST
                                        ;%WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;&M8959 (WMC) & ALL OTHER BOARDS
                                        ;REQUEST THE 'HOST' CPU TO SELECT A PORT FOR TESTING
                                        
4305                                            REQ     7
 (1)  4305  CD  06  28       18.0                CALL    REQST
 (1)  4308  00                                   .BYTE                   ;DATA PATTERN NUMBER
 (1)  4309  00  00                               .WORD                   ;SYSTEM '*' COUNT
```

```
  (1)  430B  00  00                                        .WORD                    ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (1)  430D  00                                            .BYTE                    ;DATA COMPARE FLAG IF =1
  (1)  430E  07                                            .BYTE   7                ;REQUEST CODE
1478   430F                                  RIN     R12L                ;GET THE SELECTED PORT FROM 'HOST'
  (1)  430F  DB  94              10.0                    IN      R12L                ;READ R12' INTO AC
  (1)  4311  7F                   4.0                    MOV     A,A                 ;RETRY LINK
1479   4312  32  03  48           13.0           STA     UNITMP              ;SAVE IT
1480   4315  A7                    4.0           ANA     A                   ;SET THE CONDITION BITS
1481   4316  C2  27  43           10.0           JNZ     FOUND               ;SEE IF ANY WERE SELECTED IN 'HOST'
1482   4319                                      ERR     EXIT,NOUNIT
  (1)                                           ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
  (1)  4319  CD  09  28           18.0                    CALL    ERLP                ;PROCESS ERROR - DO 2.3
  (1)        0001                                         MSGN    =       MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  431C  01                                           .BYTE   MSGN                ;MESSAGE NUMBER ID
  (1)  431D  00                                           .BYTE
  (1)  431E  CD  15  28           18.0                    NOUNIT::        CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.3
  (1)  4321  DA  F8  45           10.0                    JC      EXIT                ;LOOP ADDRESS IF LOOP SPECIFIED
1483                                        ;>OPERATOR ERROR - NO TU78 UNIT SELECTED
1484                                        ;>FATAL ERROR - TEST ABORTED
1485   4324  C3  F8  45           10.0           JMP     EXIT
1486                                        ;HERE IF THE HOST CPU INDICATED A PORT TO USE
1487
1488   4327  CD  4F  47           18.0  FOUND:   CALL    CLEAR               ;INIT ALL THE PORTS IN THE PORT BOARD
1489   432A  3A  03  48           13.0           LDA     UNITMP              ;GET THE RESPONSE FROM 'HOST' CPU
1490   432D  06  00                7.0           MVI     B,0                 ;INIT 'B'
1491   432F  E6  01                7.0           ANI     1                   ;IS UNIT 0 SELECTED?
1492   4331  C2  47  43           10.0           JNZ     FPORT               ;YUP - FOUND THE PORT TO USE
1493   4334  04                    4.0           INR     B                   ;NO - COUNT THE PORT
1494   4335  3A  03  48           13.0           LDA     UNITMP              ;GET RESPONSE AGAIN
1495   4338  E6  02                7.0           ANI     2                   ;PORT 1 SELECTED?
1496   433A  C2  47  43           10.0           JNZ     FPORT               ;YUP
1497   433D  04                    4.0           INR     B                   ;COUNT THE PORT
1498   433E  3A  03  48           13.0           LDA     UNITMP              ;GET RESPONSE
1499   4341  E6  04                7.0           ANI     4                   ;PORT 2 SELECTED?
1500   4343  C2  47  43           10.0           JNZ     FPORT               ;YUP
1501   4346  04                    4.0           INR     B                   ;NO - ASSUME PORT 3
1502
1503   4347  78                    4.0  FPORT:   MOV     A,B                 ;PUT UNIT # IN AC
1504   4348  32  04  48           13.0           STA     SUNIT               ;SAVE THE UNIT UNDER TEST
1505   434B  DB  E0                10.0           IN      INTSTA              ;GET THE PORT SELECT STATUS
1506   434D  E6  80                7.0           ANI     BIT7
1507   434F  B0                    4.0           ORA     B                   ;ADD IN THE SELECTED PORT #
1508   4350  D3  E0                10.0           OUT     MBSEL               ;SELECT THE DESIRED PORT TO USE
1509   4352  3E  60                7.0           MVI     A,P.LWR!P.LCS       ;GET THE LOOP-COMMAND-STATUS BIT
1510   4354  D3  48                10.0           OUT     PDIAG               ;SELECT IT
1511   4356  AF                    4.0           XRA     A
1512   4357  D3  44                10.0           OUT     TAMT                ;CLEAR THE AMTIE LINES
1513   4359  D3  D2                10.0           OUT     TRKENA              ;CLEAR THE TRACK ENABLE BITS
1514   435B  D3  D2                10.0           OUT     TRKENA
1515   435D  3E  04                7.0           MVI     A,P.RPEN            ;SET READ PATH ENABLE
1516   435F  D3  4C                10.0           OUT     PENAB               ;NOW
1517   4361  3E  10                7.0           MVI     A,RDCLK             ;SETUP THE READ AND WRITE CLOCK TO 'NORMAL'
1518   4363  D3  F0                10.0           OUT     CLKCTL
1519   4365  D3  C0                10.0           OUT     DBUSCTL             ;CLEAR THE DATA BUS CONTROL WORD
```

```
1520  4367  21  4E  46  10.0          LXI   H,XFRPTB        ;POINT TO THE 1ST DATA XFR PARAMETER TABLE
1521  436A  22  08  48  16.0          SHLD  PTABLE          ;SAVE THE TABLE ADDRESS
1522                              ;INIT THE COUNTER CONTROL AND COUNTERS
1523
1524  436D                        T1LOOP: INICNT
 (1)  436D  3E  30        7.0                MVI   A,@060          ;INIT THE
 (1)  436F  D3  D7       10.0                OUT   CNTCTL          ;BYTE COUNTER
 (1)  4371  3E  70        7.0                MVI   A,@160          ;INIT THE
 (1)  4373  D3  D7       10.0                OUT   CNTCTL          ;PAD COUNTER
 (1)  4375  3E  B0        7.0                MVI   A,@260          ;INIT THE
 (1)  4377  D3  D7       10.0                OUT   CNTCTL          ;ERROR COUNTER
1525  4379                              CLREC1
 (1)  4379  AF            4.0                XRA   A               ;CLEAR THE ACCUMULATOR
 (1)  437A  D3  D6       10.0                OUT   ERRCNT          ;CLEAR BITS 7-0
 (1)  437C  D3  D6       10.0                OUT   ERRCNT          ;CLEAR BITS 15-8
1526
1527  437E  2A  08  48  16.0          LHLD  PTABLE          ;GET THE PARAMETER POINTER
1528  4381  7E           7.0          MOV   A,M             ;GET THE LOW BYTE OF EXPECTED ADDRESS
1529  4382  32  01  48  13.0          STA   EXPADR          ;SAVE IT
1530  4385  23           6.0          INX   H               ;POINT TO THE HIGH BYTE OF ADDRESS
1531  4386  7E           7.0          MOV   A,M             ;GET THE HIGH BYTE
1532  4387  32  02  48  13.0          STA   EXPADR+1        ;SAVE IT
1533  438A  23           6.0          INX   H               ;POINT TO THE PACKING MODE NUMBER
1534  438B  7E           7.0          MOV   A,M             ;GET IT
1535  438C  32  FF  47  13.0          STA   FMTNUM          ;SAVE IT
1536  438F  23           6.0          INX   H               ;POINT TO THE BYTE COUNT FOR THIS XFR
1537  4390  7E           7.0          MOV   A,M             ;GET IT
1538  4391  32  00  48  13.0          STA   BCNT            ;SAVE IT
1539  4394  23           6.0          INX   H               ;POINT TO THE # SCLK'S TO CHECK IN XFR
1540  4395  7E           7.0          MOV   A,M             ;GET THE COUNT
1541  4396  32  FE  47  13.0          STA   ESCLK           ;SAVE THE NUMBER
1542  4399  23           6.0          INX   H               ;POINT TO THE 1ST WORD IN NEXT ENTRY
1543  439A  22  0A  48  16.0          SHLD  NPTABLE         ;SAVE THE ADDRESS OF NEW TABLE
1544
1545  439D  3A  FF  47  13.0          LDA   FMTNUM          ;GET THE PACKING MODE # AGAIN
1546  43A0  07           4.0          RLC
1547  43A1  07           4.0          RLC
1548  43A2  07           4.0          RLC
1549  43A3  07           4.0          RLC
1550  43A4  E6  70       7.0          ANI   $70            ;POSITION THE FMT #
1551  43A6  D3  D0      10.0          OUT   DATACTL        ;LOAD THE DATA CONTROL WORD
1552  43A8                            ROUT  R02H            ;SAVE FOR ERROR PRINTING
 (1)  43A8  D3  85      10.0                OUT     R02H            ;WRITE AC INTO R02H
 (1)  43AA  7F           4.0                MOV     A,A             ;RETRY LINK
1553  43AB  3E  9B       7.0          MVI   A,@233          ;SET THE DDR CONTROL TO 'IN'
1554  43AD  D3  DB      10.0          OUT   DDRCTL          ;TO READ THE 'DDR'
1555  43AF  06  06       7.0          MVI   B,6             ;SETUP TO CLEAR THE RESIDUAL GROUP
1556  43B1  AF           4.0          XRA   A               ;DATA STORAGE FOR THIS XFR
1557  43B2  21  3E  46  10.0          LXI   H,RGRP          ;POINT TO THE RESID GRP AREA
1558  43B5  77           7.0  T1RCLR: MOV   M,A             ;CLEAR A LOCATION IN THE RESID GRP AREA
1559  43B6  23           6.0          INX   H               ;POINT TO THE NEXT LOCATION
1560  43B7  05           4.0          DCR   B               ;DECREMENT THE LOOP COUNT
1561  43B8  C2  B5  43  10.0          JNZ   T1RCLR          ;LOOP TILL ALL BYTES CLEARED
1562  43BB  21  3E  46  10.0          LXI   H,RGRP          ;RESET THE POINTER
```

```
1563  43BE  3E  01        7.0          MVI   A,1          ;GET THE 1ST DATA BYTE IN THE RESID GRP
1564  43C0  77            7.0  T1RFIL: MOV   M,A          ;INSERT THE DATA BYTE
1565  43C1  23            6.0          INX   H            ;POINT TO THE NEXT LOCATION
1566  43C2  E5           12.0          PUSH  H            ;SAVE H & L
1567  43C3  21  00  48    10.0          LXI   H,BCNT       ;POINT TO THE BYTE COUNT STORAGE
1568  43C6  BE            7.0          CMP   M            ;SEE IF DONE FILLING
1569  43C7  E1           10.0          POP   H
1570  43C8  CA  CF  43    10.0          JZ    T1FOK        ;JUMP IF RESID GRP BUILT
1571  43CB  3C            4.0          INR   A            ;UPDATE DATA BYTE FOR RESID GRP STORAGE
1572  43CC  C3  C0  43    10.0          JMP   T1RFIL       ;LOOP TILL DONE
1573
1574  43CF  3A  00  48    13.0  T1FOK:  LDA   BCNT         ;GET THE DESIRED BYTE COUNT FOR XFR
1575  43D2  32  23  4F    13.0          STA   BYTEL
1576  43D5  AF            4.0          XRA   A
1577  43D6  32  24  4F    13.0          STA   BYTEH
1578  43D9  CD  46  00    18.0          CALL  LKMOD7       ;CALC THE MOD7 REMAINDER
1579  43DC  3A  00  48    13.0          LDA   BCNT
1580  43DF  3D            4.0          DCR   A
1581  43E0  E6  1F        7.0          ANI   @37          ;THIS IS THE MOD32 NUMBER
1582  43E2  57            4.0          MOV   D,A          ;SAVE IT
1583  43E3  78            4.0          MOV   A,B          ;GET THE MOD7 NUMBER
1584  43E4  0F            4.0          RRC                ;POSITION IT TO LOOK LIKE THE MOD7
1585  43E5  0F            4.0          RRC                ;FOR THE CRC DATA GROUP
1586  43E6  0F            4.0          RRC
1587  43E7  B2            4.0          ORA   D            ;ADD IN THE MOD32 NUMBER
1588  43E8  32  4C  46    13.0          STA   RCHRWD       ;SAVE IT IN THE CRC DATA GROUP AREA
1589  43EB  3A  00  48    13.0          LDA   BCNT         ;GET THE BYTE COUNT AGAIN
1590  43EE  D3  D4        10.0          OUT   BYTCNT       ;LOAD BYTE COUNTER WITH EXPECTED # BYTES
1591  43F0  AF            4.0          XRA   A            ;FROM THE READ PATH LOGIC
1592  43F1  D3  D4        10.0          OUT   BYTCNT
1593  43F3  D3  D5        10.0          OUT   PADCNT       ;FAKE OUT THE PAD COUNTER
1594  43F5  D3  D5        10.0          OUT   PADCNT
1595  43F7                              ROUT  R02L         ,CLEAR OUT THE REST OF REG 2 IN CAS
 (1)  43F7  D3  84        10.0                OUT   R02L           ;WRITE AC INTO R02L
 (1)  43F9  7F            4.0                 MOV   A,A            ;RETRY LINK
1596  43FA                              ROUT  R05L         ;CLEAR THE BYTE/SCLK COUNTER FOR REPORTS
 (1)  43FA  D3  8A        10.0                OUT   R05L           ;WRITE AC INTO R05L
 (1)  43FC  7F            4.0                 MOV   A,A            ;RETRY LINK
1597  43FD  32  FD  47    13.0          STA   SCLKCT       ;INIT THE SCLK DETECTED COUNTER
1598  4400  32  07  48    13.0          STA   ONCE         ;CLEAR THE BYTE COUNTER POSITIVE FLAG
1599                          ;RESTART THE WMC AND SET WMC TO SINGLE STEP MODE
1600
1601  4403  3E  01        7.0          MVI   A,W.RST      ;GET THE WMC RESTART BIT
1602  4405  D3  D3        10.0          OUT   WMCCTL       ;SET RESTART
1603  4407  3E  15        7.0          MVI   A,SSTEP!RDCLK ;GET SINGLE STEP FOR WMC
1604  4409  D3  F0        10.0          OUT   CLKCTL       ;SET SINGLE STEP MODE
1605  440B  32  05  48    13.0          STA   CCTLWD       ;SAVE THE WORD
1606  440E  CD  5B  48    18.0          CALL  CLKSYS       ;CLOCK THE SYSTEM TO CAUSE RESTART TO
1607  4411  CD  5B  48    18.0          CALL  CLKSYS       ;TO TAKE HOLD
1608  4414  AF            4.0          XRA   A
1609  4415  D3  D3        10.0          OUT   WMCCTL       ;FINISH THE RESTART CYCLE
1610
1611                          ;CLOCK THE WMC THRU ITS RESTART AREA
1612
```

```
1613  4417  06  B4        7.0          MVI   B,180              ;GET A LOOP COUNT
1614  4419  CD  5B  48    18.0  CLKRS:  CALL  CLKSYS            ;CLOCK THE SYSTEM
1615  441C  05              4.0          DCR   B                ;DECREMENT THE LOOP COUNTER
1616  441D  C2  19  44      10.0          JNZ   CLKRS            ;CLOCK TILL COUNT RUN OUT
1617  4420  3E  90          7.0          MVI   A,W.GCR!W.ENAB   ;SETUP FOR A GCR DATA XFR
1618  4422  D3  D3         10.0  T1ST:   OUT   WMCCTL           ;LOAD THE WORD IN WMC
1619
1620                               ;CALCULATE THE RESIDUAL DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
1621
1622  4424  21  3E  46     10.0  C1EC1:  LXI   H,RGRP          ;POINT THE THE 1ST DATA GROUP DATA TABLE
1623  4427  06  07          7.0          MVI   B,7              ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
1624  4429  CD  1C  48     18.0          CALL  CLECC            ;INIT THE ECC CHARACTER
1625
1626  442C  7E              7.0  C1EC1L: MOV   A,M              ;GET A DATA BYTE FROM THE TABLE
1627  442D  CD  21  48     18.0          CALL  ECC              ;CALC ECC WITH THIS CHARACTER
1628  4430  23              6.0          INX   H                ;POINT TO THE NEXT DATA BYTE IN TABLE
1629  4431  05              4.0          DCR   B                ;DECREMENT THE COUNTER OF DATA BYTES
1630  4432  C2  2C  44      10.0          JNZ   C1EC1L           ;STAY HERE TILL ALL DATA BYTES USED
1631  4435  3A  48  48     13.0          LDA   ECCCHR           ;GET THE CALCULATED ECC CHARACTER
1632  4438  32  45  46     13.0          STA   RGECC            ;SAVE THE CALCULATED ECC CHARACTER
1633
1634                               ;CALCULATE THE 2ND DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
1635
1636  443B  21  46  46     10.0  C1EC2:  LXI   H,CRCGRP        ;POINT THE THE 2ND DATA GROUP DATA TABLE
1637  443E  06  07          7.0          MVI   B,7              ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
1638  4440  CD  1C  48     18.0          CALL  CLECC            ;INIT THE ECC CHARACTER
1639
1640  4443  7E              7.0  C1EC2L: MOV   A,M              ;GET A DATA BYTE FROM THE TABLE
1641  4444  CD  21  48     18.0          CALL  ECC              ;CALC ECC WITH THIS CHARACTER
1642  4447  23              6.0          INX   H                ;POINT TO THE NEXT DATA BYTE IN TABLE
1643  4448  05              4.0          DCR   B                ;DECREMENT THE COUNTER OF DATA BYTES
1644  4449  C2  43  44      10.0          JNZ   C1EC2L           ;STAY HERE TILL ALL DATA BYTES USED
1645  444C  3A  48  48     13.0          LDA   ECCCHR           ;GET THE CALCULATED ECC CHARACTER
1646  444F  32  4D  46     13.0          STA   CRCECC           ;SAVE THE CALCULATED ECC CHARACTER
1647  4452  3E  A8          7.0  T1PLO:  MVI   A,R.PLO1.R.TBJN!R.PLOD  ;SET PLO BYPASS MODE 2 ENABLE READ PATH CLOCK
1648                                                                    ;SET TIE BUS JAM AND PLO DISABLE
1649  4454  D3  09         10.0          OUT   RPCTL
1650  4456  AF              4.0          XRA   A
1651  4457  D3  0A         10.0          OUT   RTIEB            ;CLEAR THE TIE BUS
1652  4459  D3  08         10.0          OUT   RFIFOL           ;CLOCK THE FIFO'S
1653  445B  3E  0D          7.0          MVI   A,RCLRT          ;ISSUE CLEAR ALL COMMAND
1654  445D  D3  0B         10.0          OUT   RCMD
1655
1656  445F  00              4.0          NOP                    ;WAIT
1657  4460  00              4.0          NOP                    ;WAIT
1658  4461  00              4.0          NOP
1659  4462  00              4.0          NOP
1660  4463  00              4.0          NOP
1661
1662  4464  3E  A9          7.0          MVI   A,R.PLO1!R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
1663                                                                    ;SET TIE BUS JAM AND PLO DISABLE
1664  4466  D3  09         10.0          OUT   RPCTL
1665                               ;LOAD THE 'END MARK' INTO THE FIFO
1666
```

```
1667  4468  21  10  48    10.0  T1EMK:   LXI   H,TRNOUT        ;POINT TO THE 4X5 STORAGE AREA
1668  446B  06  05          7.0          MVI   B,5             ;SET A LOOP COUNT
1669  446D  3E  FF          7.0  T1EMKL: MVI   A,$FF           ;GET 8 TRKS OF DATA - ALL 1'S
1670  446F  77              7.0          MOV   M,A             ;SAVE IN THE TABLE
1671  4470  23              6.0          INX   H               ;POINT TO THE PARITY TRK AREA
1672  4471  3E  01          7.0          MVI   A,1
1673  4473  77              7.0          MOV   M,A             ;SET IT TO 1 ALSO
1674  4474  23              6.0          INX   H               ;POINT TO NEXT STORAGE AREA
1675  4475  05              4.0          DCR   B               ;DECREMENT THE LOOP COUNT
1676  4476  C2  6D  44      10.0          JNZ   T1EMKL          ;LOOP TILL END MARK IN STORAGE
1677  4479  CD  27  46      18.0          CALL  LOADIT          ;INSERT IT INTO THE FIFO
1678
1679  447C  21  3E  46      10.0  T1SG1:  LXI   H,RGRP          ;POINT TO THE 1ST DATA GROUP FOR TRANSLATION
1680  447F  CD  12  46      18.0          CALL  CONVRT          ;CONVERT SUBGROUP - 4X5
1681  4482  CD  27  46      18.0          CALL  LOADIT          ;INJECT THE DATA INTO THE FIFO
1682  4485  21  42  46      10.0  T1SG2:  LXI   H,RGRP2         ;POINT TO THE 2ND SUBGROUP OF DATA
1683  4488  CD  12  46      18.0          CALL  CONVRT          ;MAKE IT 4X5
1684  448B  CD  27  46      18.0          CALL  LOADIT          ;INJECT IT INTO FIFO
1685
1686  448E  21  46  46      10.0  T1SG3:  LXI   H,CRCGRP        ;POINT TO THE 2ND DATA GROUP - SUBGROUP 3
1687  4491  CD  12  46      18.0          CALL  CONVRT          ;MAKE IT 4X5
1688  4494  CD  27  46      18.0          CALL  LOADIT          ;INJECT THE DATA INTO FIFO
1689  4497  21  4A  46      10.0  T1SG4:  LXI   H,CRCG2         ;POINT TO THE 4TH SUBGROUP OF DATA
1690  449A  CD  12  46      18.0          CALL  CONVRT          ;MAKE IT 4X5
1691  449D  CD  27  46      18.0          CALL  LOADIT          ;INJECT THE DATA INTO FIFO
1692                                ;LOAD THE MARK 2 SIGNAL INTO THE FIFO
1693  44A0  06  03          7.0  T1MK2:  MVI   B,3
1694  44A2  21  10  48      10.0          LXI   H,TRNOUT        ;POINT TO THE LOAD BUFFER
1695  44A5  3E  FF          7.0  T1MK2L: MVI   A,$FF
1696  44A7  77              7.0          MOV   M,A             ;SAVE 8 TRKS
1697  44A8  23              6.0          INX   H
1698  44A9  3E  01          7.0          MVI   A,1
1699  44AB  77              7.0          MOV   M,A             ;SAVE THE PARITY TRK
1700  44AC  23              6.0          INX   H
1701  44AD  05              4.0          DCR   B               ;DECREMENT THE LOOP COUNTER
1702  44AE  C2  A5  44      10.0          JNZ   T1MK2L          ;LOOP TILL DONE
1703  44B1  06  04          7.0          MVI   B,4
1704  44B3  AF              4.0          XRA   A
1705  44B4  77              7.0  T1M2L2: MOV   M,A             ;SAVE 8 TRKS OF 0
1706  44B5  23              6.0          INX   H
1707  44B6  05              4.0          DCR   B               ;DECREMENT LOOP COUNTER
1708  44B7  C2  B4  44      10.0          JNZ   T1M2L2          ;LOOP TILL ALL MARK - 2 IN BUFFER
1709  44BA  CD  27  46      18.0          CALL  LOADIT          ;STUFF MARK - 2 IN THE FIFO
1710
1711                                ;HERE TO START THE DIAGNOSTIC READ COMMAND IN THE READ PATH TO
1712                                ;CAUSE THE DATA IN THE FIFO TO GO THRU READ PATH & ENTER THE
1713                                ;WMC BYTE ASSY LOGIC VIA THE CRC BOARD
1714
1715  44BD  3E  0B          7.0          MVI   A,DIARD         ;LOAD THE DIAGNOSTIC READ
1716  44BF  D3  0B         10.0          OUT   RCMD            ;COMMAND
1717
1718  44C1  11  01  00      10.0  T1CLKA: LXI   D,1            ;SET WATCH DOG INCREMENT
1719  44C4  21  68  C5      10.0          LXI   H,-15000        ;SET WATCH DOG COUNT TO 15000
1720  44C7  AF              4.0          XRA   A
```

```
1721  44C8  32  06  48      13.0          STA     HAVEXFR         ;CLEAR THE HAVE 'XFER' BIT FLAG
1722  44CB  D3  0C          10.0  T1CLK:  OUT     RINST           ;STEP THE READ PATH
1723  44CD  CD  5B  48      18.0          CALL    CLKSYS          ;CLOCK THE WMC
1724  44D0  DB  D1          10.0          IN      TUSEL0          ;SEE IF BYTE COUNTER WENT TO 0
1725  44D2  E6  40           7.0          ANI     BIT6            ;SO THE WMC CAN BE DISABLED TO FINISH XFR
1726  44D4  C2  DB  44      10.0          JNZ     T1CLK1          ;JUMP IF BYTE TERMINAL COUNTER STILL = 0
1727  44D7  3E  10           7.0          MVI     A,W.GCR         ;DROP THE WMC ENABLE BIT
1728  44D9  D3  D3          10.0          OUT     WMCCTL          ;IN THE WMC CONTROL WORD
1729
1730  44DB  DB  D0          10.0  T1CLK1: IN      WMCSTA          ;SEE IF EVER GET 'XFER' BIT
1731  44DD  E6  20           7.0          ANI     BIT5            ;SAVE JUST 'XFER' BIT IN STATUS
1732  44DF  CA  E5  44      10.0          JZ      T1CLK2          ;JUMP IF NOT UP
1733  44E2  32  06  48      13.0          STA     HAVEXFR         ;SET THE 'XFER' FLAG
1734
1735  44E5  DB  C0          10.0  T1CLK2: IN      DBUSSTA         ;GET THE DATABUS STATUS
1736  44E7  E6  02           7.0          ANI     T.SCLK          ;SEE IF SCLK UP - DATA READY
1737  44E9  C2  0D  45      10.0          JNZ     T1SCLK          ;JUMP IF SCLK UP
1738
1739  44EC  19              10.0  T1CLK3: DAD     D
1740  44ED  D2  CB  44      10.0          JNC     T1CLK           ;STAY IN LOOP UNLESS TIMEOUT
1741
1742  44F0  3A  06  48      13.0          LDA     HAVEXFR         ;SEE IF EVER GOT 'XFER' BIT SET
1743  44F3  A7               4.0          ANA     A
1744  44F4  C2  FC  44      10.0          JNZ     DUMM1A          ;JUMP IF NO SCLK EVER BUT HAVE 'XFER'
1745  44F7                                ERR     T1LOOP,DUMM1A,3
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44F7  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44FA  02                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44FB  03                                     .BYTE   3
 (1)  44FC  CD  15  28      18.0          DUMM1A::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44FF  DA  6D  43      10.0                  JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1746                                ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO SET ... DATA NOT READY FROM BYTE ASSY
1747                                ;>DATA HAS BEEN SENT TO WMC FROM CRC BOARD - 'XFER' SEEN IN WMCSTA WORD
1748  4502                          T1ERNS: ERR     T1LOOP,DUMM1,3
 (1)                                      ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4502  CD  09  28      18.0                  CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0003                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4505  03                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4506  03                                     .BYTE   3
 (1)  4507  CD  15  28      18.0          DUMM1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  450A  DA  6D  43      10.0                  JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1749                                ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO SET IN DBUSSTA WORD ... DATA NOT READY
1750                                ;>AFTER ISSUEING A READ FORWARD COMMAND TO THE TM78 - NO 'XFER' SEEN IN
1751                                ;>WMCSTA WORD. DATA NOT SEEN FROM CRC LOGIC.
1752
1753                                ;HERE TO CLOCK THE WMC SO THE 'SCLK' SIGNAL WILL GO -0
1754                                ;THIS IS NECESSARY BEFORE STROBING THE DDR REGISTERS...THEY ARE ONLY
1755                                ;VALID AFTER 'SCLK' HAS TRANSITIONED TO ZERO INDICATING THE DATA IS
1756                                ;ON THE MASSBUS
1757
1758  450D  11  01  00      10.0  T1SCLK: LXI     D,1
1759  4510  21  78  EC      10.0          LXI     H,-5000         ;SETUP THE DROP DEAD TIMER
1760
```

```
1761  4513  D3  0C        10.0  T1SCKL: OUT   RINST          ;CLOCK THE READ PATH ONCE
1762  4515  CD  5B  48     18.0          CALL  CLKSYS         ;CLOCK THE WMC TO CLEAR 'SCLK'
1763  4518  DB  C0        10.0          IN    DBUSSTA        ;SEE IF 'SCLK' IS GONE
1764  451A  E6  02         7.0          ANI   T.SCLK         ;SAVE ONLY THE 'SCLK' BIT IN STATUS WORD
1765  451C  CA  43  45     10.0          JZ    DUMM2          ;JUMP IF 'SCLK' = 0
1766  451F  19             10.0          DAD   D              ;DECREMENT THE TIMER
1767  4520  D2  13  45     10.0          JNC   T1SCKL         ;LOOP AGAIN IF STILL TIME LEFT
1768
1769  4523  DB  D4        10.0          IN    BYTCNT         ;GET LOW BITS OF BYTE COUNTER
1770  4525  47             4.0          MOV   B,A            ;SAVE IT
1771  4526  DB  D4        10.0          IN    BYTCNT         ;THEN HIGH BITS
1772  4528  78             4.0          MOV   A,B            ;GET THE LOW BITS AGAIN
1773  4529  A7             4.0          ANA   A
1774  452A  C2  3E  45     10.0          JNZ   T1SCKE         ;JUMP IF POSITIVE - ERROR CONDITION
1775  452D  3A  07  48     13.0          LDA   ONCE           ;SEE IF HERE BEFORE
1776  4530  A7             4.0          ANA   A              ;FOR THIS SAME DATA XFR
1777  4531  C2  3E  45     10.0          JNZ   T1SCKE         ;WHOOPS...ERROR DETECTED!
1778  4534  3E  10         7.0          MVI   A,W.GCR        ;CLEAR THE WMC ENABLE BIT SO WMC WIL
1779  4536  D3  D3        10.0          OUT   WMCCTL         ;FINISH THE DATA XFR
1780  4538  32  07  48     13.0          STA   ONCE           ;SET THE ONCE FLAG
1781  453B  C3  0D  45     10.0          JMP   T1SCLK         ;BACK TO CLOCK LOOP
1782
1783  453E                       T1SCKE: ERR   T1LOOP,DUMM2,3
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  453E  CD  09  28     18.0                    CALL  ERLP          ;PROCESS ERROR - DO 2.3
 (1)       0004                                    MSGN  =     MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4541  04                                     .BYTE MSGN          ;MESSAGE NUMBER ID
 (1)  4542  03                                     .BYTE 3
 (1)  4543  CD  15  28     18.0            DUMM2:: CALL  CKLOP         ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4546  DA  6D  43     10.0                    JC    T1LOOP        ;LOOP ADDRESS IF LOOP SPECIFIED
1784                            ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO CLEAR AFTER SETTING
1785                            ;>DURING A READ FORWARD COMMAND
1786                            ;HERE WHEN 'SCLK' WAS SET - 18 BITS OF DATA HAS BEEN ASSEMBLED FROM
1787                            ;THE BYTE ASSEMBLY LOGIC. DATA CAME FROM THE READ PATH (FIFO).
1788
1789  4549  DB  D8        10.0  T1SCK0: IN    DDRA           ;GET THE DDR A DATA
1790  454B  32  FA  47     13.0          STA   SDDRA          ;SAVE IT
1791  454E                              ROUT  R16L           ;SAVE FOR PRINTING
 (1)  454E  D3  9C        10.0                OUT   R16L             ;WRITE AC INTO R16L
 (1)  4550  7F             4.0                MOV   A,A              ;RETRY LINK
1792  4551  DB  D9        10.0          IN    DDRB           ;GET DDR B DATA
1793  4553  32  FB  47     13.0          STA   SDDRB          ;SAVE IT
1794  4556                              ROUT  R16H           ;SAVE FOR ERROR REPORTING
 (1)  4556  D3  9D        10.0                OUT   R16H             ;WRITE AC INTO R16H
 (1)  4558  7F             4.0                MOV   A,A              ;RETRY LINK
1795  4559  DB  DA        10.0          IN    DDRC           ;GET DDR C DATA
1796  455B  E6  03         7.0          ANI   3
1797  455D  32  FC  47     13.0          STA   SDDRC          ;SAVE IT
1798  4560  F6  80         7.0          ORI   BIT7           ;ADD IN THE PRINT FLAG
1799  4562                              ROUT  R17L           ;SAVE IT FOR PRINTING
 (1)  4562  D3  9E        10.0                OUT   R17L             ;WRITE AC INTO R17L
 (1)  4564  7F             4.0                MOV   A,A              ;RETRY LINK
1800
1801  4565  3A  FD  47     13.0          LDA   SCLKCT         ;GET THE CURRENT SCLK COUNT
```

```
1802  4568  3C                4.0          INR     A               ;COUNT THIS SCLK
1803  4569  32  FD  47        13.0          STA     SCLKCT          ;SAVE THE NEW COUNT
1804  456C                                  ROUT    R05L            ;ALSO IN CAS FOR ERROR REPORTING
 (1)  456C  D3  8A             10.0                  OUT     R05L            ;WRITE AC INTO R05L
 (1)  456E  7F                  4.0                  MOV     A,A             ;RETRY LINK
1805  456F  AF                  4.0          XRA     A
1806  4570                                  ROUT    R05H
 (1)  4570  D3  8B             10.0                  OUT     R05H            ;WRITE AC INTO R05H
 (1)  4572  7F                  4.0                  MOV     A,A             ;RETRY LINK
1807
1808                                  ;CHECK THE DATA IN DDR A - BITS 7-0 OF THE 18 BITS ASSEMBLED
1809
1810  4573  2A  01  48          16.0  T1CKA:  LHLD    EXPADR          ;POINT TO THE TABLE OF EXPECTED DATA
1811  4576  7E                  7.0          MOV     A,M             ;GET BITS 7-0
1812  4577                                  ROUT    R14L            ;SAVE FOR ERROR REPORTING
 (1)  4577  D3  98             10.0                  OUT     R14L            ;WRITE AC INTO R14L
 (1)  4579  7F                  4.0                  MOV     A,A             ;RETRY LINK
1813  457A  23                  6.0          INX     H               ;POINT TO NEXT BYTE
1814  457B  7E                  7.0          MOV     A,M             ;GET BITS 15-8
1815  457C                                  ROUT    R14H            ;SAVE IT
 (1)  457C  D3  99             10.0                  OUT     R14H            ;WRITE AC INTO R14H
 (1)  457E  7F                  4.0                  MOV     A,A             ;RETRY LINK
1816  457F  23                  6.0          INX     H
1817  4580  7E                  7.0          MOV     A,M             ;GET BITS 17-16
1818  4581  F6  80              7.0          ORI     BIT7            ;ADD IN THE PRINT FLAG
1819  4583                                  ROUT    R15L            ;SAVE FOR ERROR REPORT
 (1)  4583  D3  9A             10.0                  OUT     R15L            ;WRITE AC INTO R15L
 (1)  4585  7F                  4.0                  MOV     A,A             ;RETRY LINK
1820  4586  3A  FA  47          13.0          LDA     SDDRA           ;GET THE DDR A DATA
1821  4589                                  ROUT    ADATA           ;SAVE IN CASE OF ERROR DETECTED
 (1)  4589  D3  94             10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  458B  7F                  4.0                  MOV     A,A             ;RETRY LINK
1822  458C  2A  01  48          16.0          LHLD    EXPADR          ;POINT TO THE EXPECTED DATA TABLE
1823  458F  BE                  7.0          CMP     M               ;ACTUAL DATA = EXPECTED?
1824  4590  CA  9C  45          10.0          JZ      T1CKAC          ;JUMP IF OK
1825  4593  7E                  7.0          MOV     A,M             ;GET THE EXPECTED DATA
1826  4594                                  ROUT    EDATA           ;SAVE FOR ERROR REPORT
 (1)  4594  D3  95             10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4596  7F                  4.0                  MOV     A,A             ;RETRY LINK
1827  4597                                  ERRB    T1LOOP,T1CKAC,@10
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4597  CD  12  28          18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0005                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  459A  05                                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  459B  08                                     .BYTE   @10             ;PRINT ROUTINE NUMBER
 (1)  459C  CD  15  28          18.0          T1CKAC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  459F  DA  5D  43          10.0                  JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1828                          ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 7-0
1829  45A2  23                  6.0  T1CKB:  INX     H               ;POINT TO THE EXPECTED DATA FOR DDR B
1830  45A3  3A  FB  47          13.0          LDA     SDDRB           ;GET THE DDR B DATA
1831  45A6                                  ROUT    ADATA           ;SAVE FOR ERROR REPORT
 (1)  45A6  D3  94             10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  45A8  7F                  4.0                  MOV     A,A             ;RETRY LINK
1832  45A9  BE                  7.0          CMP     M               ;ACTUAL - EXPECTED DATA?
```

```
 1833  45AA  CA  B6  45   10.0          JZ      T1CKBC          ;JUMP IF OK
 1834  45AD  7E               7.0       MOV     A,M             ;GET THE EXPECTED DATA
 1835  45AE                             ROUT    EDATA           ;SAVE FOR ERROR REPORT
  (1)  45AE  D3  95           10.0              OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  45B0  7F               4.0               MOV     A,A             ;RETRY LINK
 1836  45B1                             ERRB    T1LOOP,T1CKBC,@10
  (1)                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  45B1  CD  12  28       18.0              CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0006                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  45B4  06                               .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  45B5  08                               .BYTE   @10             ;PRINT ROUTINE NUMBER
  (1)  45B6  CD  15  28       18.0      T1CKBC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  45B9  DA  6D  43       10.0              JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
 1837                          ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 15-8
 1838
 1839  45BC  23                6.0  T1CKC:  INX     H               ;POINT TO EXPECTED DDR C DATA
 1840  45BD  3A  FC  47       13.0          LDA     SDDRC           ;GET DDR C DATA
 1841  45C0                            ROUT    ADATA           ;SAVE FOR ERROR REPORT
  (1)  45C0  D3  94           10.0              OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  45C2  7F               4.0               MOV     A,A             ;RETRY LINK
 1842  45C3  BE               7.0       CMP     M               ;ACTUAL DATA = EXPECTED DDRC DATA?
 1843  45C4  CA  D0  45       10.0      JZ      T1CKCC          ;JUMP IF OK
 1844  45C7  7E               7.0       MOV     A,M             ;GET THE EXPECTED DATA
 1845  45C8                            ROUT    EDATA           ;SAVE FOR ERROR REPORT
  (1)  45C8  D3  95           10.0              OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  45CA  7F               4.0               MOV     A,A             ;RETRY LINK
 1846  45CB                            ERRB    T1LOOP,T1CKCC,@10
  (1)                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  45CB  CD  12  28       18.0              CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0007                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  45CE  07                               .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  45CF  08                               .BYTE   @10             ;PRINT ROUTINE NUMBER
  (1)  45D0  CD  15  28       18.0      T1CKCC::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  45D3  DA  6D  43       10.0              JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
 1847                          ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 17-16
 1848
 1849  45D6  2A  01  48       16.0          LHLD    EXPADR          ;GET THE CURRENT EXPECTED DATA ADDR.
 1850  45D9  23                6.0          INX     H
 1851  45DA  23                6.0          INX     H               ;UPDATE THE EXPECTED ADDRESS POINTER
 1852  45DB  23                6.0          INX     H               ;TO THE DATA
 1853  45DC  22  01  48       16.0          SHLD    EXPADR          ;SAVE THE 'NEW' EXPECTED DATA TABLE ADDR.
 1854  45DF  3A  FD  47       13.0          LDA     SCLKCT          ;GET THE SCLK COUNT
 1855  45E2  21  FE  47       10.0          LXI     H,ESCLK         ;POINT TO THE EXPECTED SCLK COUNT
 1856  45E5  BE                7.0          CMP     M               ;DONE WITH THIS TEST PASS?
 1857  45E6  C2  C1  44       10.0          JNZ     T1CLKA          ;JUMP IF STILL EXPECTING MORE SCLKS
 1858
 1859                          ;HERE TO SEE IF ALL XFRS IN THE 'PTABLE' HAVE BEEN DONE
 1860
 1861  45E9  2A  0A  48       16.0          LHLD    NPTABLE         ;POINT TO THE TABLE
 1862  45EC  7E                7.0          MOV     A,M             ;GET THE 1ST WORD OF AN ENTRY
 1863  45ED  FE  FF            7.0          CPI     @377            ;END OF THE TABLE?
 1864  45EF  CA  F8  45       10.0          JZ      EXIT            ;EXIT IF END OF TABLE
 1865  45F2  22  08  48       16.0          SHLD    PTABLE          ;SAVE ADDRESS OF NEXT XFR TABLE
 1866  45F5  C3  6D  43       10.0          JMP     T1LOOP          ;EXECUTE THE NEXT XFR
```

```
 1867
 1868  45F8                                EXIT:   ENDTST  FOUND
  (1)                                      ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  45F8                                        REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
  (2)  45F8  CD  06  28        18.0                          CALL    REQST
  (2)  45FB  00                                              .BYTE                   ;DATA PATTERN NUMBER
  (2)  45FC  00  00                                          .WORD                   ;SYSTEM '*' COUNT
  (2)  45FE  00  00                                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  4600  00                                              .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  4601  07                                              .BYTE   7               ;REQUEST CODE
  (1)  4602  3A  9A  4F        13.0                LDA     ITERA           ;GET ITERATION COUNT
  (1)  4605  3D                 4.0                DCR     A               ;DOWNCOUNT
  (1)  4606  32  9A  4F        13.0                STA     ITERA           ;SAVE COUNT
  (1)  4609  F2  27  43        10.0                JP      FOUND           ;DO TEST UNTIL TILL = 0
 1869  460C  CD  4F  47        18.0                CALL    CLEAR           ;CLEAR ALL TU PORTS
 1870  460F  C3  18  28        10.0                JMP     TSTEND
 1871                                     ;HERE IS THE SUBROUTINES NECESSARY FOR CONVERTING SUBGROUPS AND LOADING
 1872                                     ;THEM INTO THE FIFO
 1873
 1874  4612                               $
  (1)                                     ; ***************************************************************************
 1875                                     ;*CONVRT -- THIS SUBROUTINE WILL GET A SUBGROUP OF DATA AS POINTED TO BY
 1876                                     ;*      THE H & L REGISTERS AND CAUSE THEM TO BE TRANSLATED (4X5). THE
 1877                                     ;*      TRANSLATED DATA WILL BE IN A TABLE CALLED 'TRNOUT'.
 1878  4612                               $
  (1)                                     ; ***************************************************************************
 1879
 1880  4612  D5                 12.0  CONVRT: PUSH    D                 ;SAVE D & E
 1881  4613  11  0C  48         10.0          LXI     D,TRNIN           ;POINT TO THE TEMP STORAGE FOR DATA BYTES
 1882  4616  06  04              7.0          MVI     B,4               ;SETUP A LOOP COUNTER FOR 4 BYTES
 1883
 1884  4618  7E                  7.0  CNV1:   MOV     A,M               ;GET A BYTE OF DATA
 1885  4619  EB                  4.0          XCHG                      ;POINT TO THE STORAGE
 1886  461A  77                  7.0          MOV     M,A               ;SAVE DATA BYTE IN TEMP STORAGE
 1887  461B  EB                  4.0          XCHG                      ;RESET POINTER
 1888  461C  23                  6.0          INX     H                 ;POINT TO NEXT DATA BYTE
 1889  461D  13                  6.0          INX     D                 ;POINT TO NEXT TEMP STORAGE ADDRESS
 1890  461E  05                  4.0          DCR     B                 ;UPDATE THE LOOP COUNT
 1891  461F  C2  18  46         10.0          JNZ     CNV1              ;LOOP TILL 4 BYTES IN TEMP STORAGE
 1892  4622  CD  7A  47         18.0          CALL    T4X5              ;CONVERT THE 'SUBGROUP' OF DATA
 1893  4625  D1                 10.0          POP     D
 1894  4626  C9                 10.0          RET
 1895  4627                               $
  (1)                                     ; ***************************************************************************
 1896                                     ;*LOADIT -- THIS SUBROUTINE WILL TAKE THE TRANSLATED SUBGROUP (4X5)
 1897                                     ;*      DATA FROM TABLE 'TRNOUT' AND LOAD THE DATA INTO THE READ
 1898                                     ;*      PATH FIFO.
 1899  4627                               $
  (1)                                     ; ***************************************************************************
 1900
 1901  4627  21  10  48         10.0  LOADIT: LXI     H,TRNOUT          ;POINT TO THE TRANSLATED DATA TABLE
 1902  462A  06  05              7.0          MVI     B,5               ;GET THE LOOP COUNT FOR SUBGROUP
 1903
 1904  462C  7E                  7.0  LDFIFO: MOV     A,M               ;GET A BYTE OF DATA
```

```
1905  462D  D3  40          10.0        OUT     TCMD            ;STORE IT IN THE CMD REG FOR LOADING
1906  462F  23              6.0         INX     H               ;POINT TO THE PARITY BIT
1907  4630  7E              7.0         MOV     A,M             ;GET THE PARITY BIT
1908  4631  07              4.0         RLC
1909  4632  F6  60          7.0         ORI     P.LWR.P.LCS     ;ADD IN THE OTHER NECESSARY BITS
1910  4634  D3  48          10.0        OUT     PDIAG           ;LOAD INTO CONTROL REG FOR LOADING
1911  4636  D3  08          10.0        OUT     RFIFOL          ;STROBE THE DATA INTO FIFO
1912  4638  23              6.0         INX     H               ;POINT TO THE NEXT SET OF DATA
1913  4639  05              4.0         DCR     B               ;DECREMENT THE LOOP COUNTER
1914  463A  C2  2C  46       10.0        JNZ     LDFIFO          ;LOOP TILL 5 SETS OF DATA LOADED INTO FIFO
1915  463D  C9              10.0        RET
1916                                ;READ PATH BYTE ASSEMBLY DATA BYTES TO BE INJECTED INTO THE READ PATH
1917                                ;FIFO - 4 SUBGROUPS WORTH OF DATA - ALL DATA TRANSFERS ARE 'READ FORWARD'
1918
1919  463E  00                      RGRP:   .BYTE   0       ;RESIDUAL DATA GROUP DATA BYTE #1
1920  463F  00                              .BYTE   0
1921  4640  00                              .BYTE   0
1922  4641  00                              .BYTE   0
1923
1924  4642  00                      RGRP2:  .BYTE   0       ;2ND SUBGROUP IN RESIDUAL DATA GROUP
1925  4643  00                              .BYTE   0       ;LAST DATA BYTE IN RESIDUAL DATA GROUP
1926  4644  00                              .BYTE   0       ;THIS IS AN 'ACRC' DATA BYTE
1927  4645  00                      RGECC:  .BYTE   0       ;THIS IS THE ECC STORAGE FOR THE RESIDUAL DATA
1928
1929  4646  00                      CRCGRP: .BYTE   0       ;1ST BYTE IN CRC DATA GROUP
1930  4647  00                              .BYTE   0
1931  4648  00                              .BYTE   0
1932  4649  00                              .BYTE   0
1933
1934  464A  00                      CRCG2:  .BYTE   0       ;2ND SUBGROUP IN CRC DATA GROUP
1935  464B  00                              .BYTE   0
1936  464C  00                      RCHRWD: .BYTE   0       ;THIS IS THE RESIDUAL CHARACTER WORD
1937  464D  00                      CRCECC: .BYTE   0       ;ECC CHARACTER IN CRC SUBGROUP
1938                                ;HERE IS THE PARAMETER TABLE FOR ALL DATA XFRS TO TEST THE SKIP COUNTS
1939                                ;IN ALL DATA PACKING MODES
1940
1941  464E                          XFRPTB: XFRP    F11N,0,3,2
 (1)  464E  B9  46                              .ADDR   F11N    ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4650  00                              .BYTE   0       ;PACKING MODE 0
 (1)  4651  03                              .BYTE   3       ;BYTE COUNT THIS XFR
 (1)  4652  02                              .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1942  4653                                  XFRP    F15N,1,3,2
 (1)  4653  BF  46                              .ADDR   F15N    ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4655  01                              .BYTE   1       ;PACKING MODE 1
 (1)  4656  03                              .BYTE   3       ;BYTE COUNT THIS XFR
 (1)  4657  02                              .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1943  4658                                  XFRP    F10C1,2,1,2
 (1)  4658  C5  46                              .ADDR   F10C1   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  465A  02                              .BYTE   2       ;PACKING MODE 2
 (1)  465B  01                              .BYTE   1       ;BYTE COUNT THIS XFR
 (1)  465C  02                              .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1944  465D                                  XFRP    F10C2,2,2,2
 (1)  465D  CB  46                              .ADDR   F10C2   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  465F  02                              .BYTE   2       ;PACKING MODE 2
```

```
   (1)  4660   02                                    .BYTE   2       ;BYTE COUNT THIS XFR
   (1)  4661   02                                    .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1945    4662                                XFRP    F10C3,2,3,2
   (1)  4662   D1   46                               .ADDR   F10C3   ;ADDRESS OF EXPECTED DATA TABLE
   (1)  4664   02                                    .BYTE   2       ;PACKING MODE 2
   (1)  4665   03                                    .BYTE   3       ;BYTE COUNT THIS XFR
   (1)  4666   02                                    .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1946    4667                                XFRP    F10D1,3,1,2
   (1)  4667   D7   46                               .ADDR   F10D1   ;ADDRESS OF EXPECTED DATA TABLE
   (1)  4669   03                                    .BYTE   3       ;PACKING MODE 3
   (1)  466A   01                                    .BYTE   1       ;BYTE COUNT THIS XFR
   (1)  466B   02                                    .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1947    466C                                XFRP    F10D2,3,2,2
   (1)  466C   DD   46                               .ADDR   F10D2   ;ADDRESS OF EXPECTED DATA TABLE
   (1)  466E   03                                    .BYTE   3       ;PACKING MODE 3
   (1)  466F   02                                    .BYTE   2       ;BYTE COUNT THIS XFR
   (1)  4670   02                                    .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1948    4671                                XFRP    F10D3,3,3,2
   (1)  4671   E3   46                               .ADDR   F10D3   ;ADDRESS OF EXPECTED DATA TABLE
   (1)  4673   03                                    .BYTE   3       ;PACKING MODE 3
   (1)  4674   03                                    .BYTE   3       ;BYTE COUNT THIS XFR
   (1)  4675   02                                    .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1949    4676                                XFRP    F10D4,3,4,2
   (1)  4676   E9   46                               .ADDR   F10D4   ;ADDRESS OF EXPECTED DATA TABLE
   (1)  4678   03                                    .BYTE   3       ;PACKING MODE 3
   (1)  4679   04                                    .BYTE   4       ;BYTE COUNT THIS XFR
   (1)  467A   02                                    .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
```

```
1951  467B                                  XFRP    F10HC1,4,1,2
 (1)  467B  EF   46                                 .ADDR   F10HC1  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  467D  04                                      .BYTE   4       ;PACKING MODE 4
 (1)  467E  01                                      .BYTE   1       ;BYTE COUNT THIS XFR
 (1)  467F  02                                      .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1952  4680                                  XFRP    F10HC2,4,2,2
 (1)  4680  FB   46                                 .ADDR   F10HC2  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4682  04                                      .BYTE   4       ;PACKING MODE 4
 (1)  4683  02                                      .BYTE   2       ;BYTE COUNT THIS XFR
 (1)  4684  02                                      .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1953  4685                                  XFRP    F10HC3,4,3,2
 (1)  4685  07   47                                 .ADDR   F10HC3  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4687  04                                      .BYTE   4       ;PACKING MODE 4
 (1)  4688  03                                      .BYTE   3       ;BYTE COUNT THIS XFR
 (1)  4689  02                                      .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1954  468A                                  XFRP    F10HC4,4,4,2
 (1)  468A  13   47                                 .ADDR   F10HC4  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  468C  04                                      .BYTE   4       ;PACKING MODE 4
 (1)  468D  04                                      .BYTE   4       ;BYTE COUNT THIS XFR
 (1)  468E  02                                      .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1955  468F                                  XFRP    F10HC5,4,5,4
 (1)  468F  1F   47                                 .ADDR   F10HC5  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4691  04                                      .BYTE   4       ;PACKING MODE 4
 (1)  4692  05                                      .BYTE   5       ;BYTE COUNT THIS XFR
 (1)  4693  04                                      .BYTE   4       ;4 SCLKS EXPECTED THIS XFR
1956  4694                                  XFRP    F10HC6,4,6,4
 (1)  4694  2B   47                                 .ADDR   F10HC6  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4696  04                                      .BYTE   4       ;PACKING MODE 4
 (1)  4697  06                                      .BYTE   6       ;BYTE COUNT THIS XFR
 (1)  469E  04                                      .BYTE   4       ;4 SCLKS EXPECTED THIS XFR
```

```
1958  4699                                  XFRP    F10HD1,6,1,2
 (1)  4699  EF   46                                   .ADDR   F10HD1  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  469B  06                                        .BYTE   6       ;PACKING MODE 6
 (1)  469C  01                                        .BYTE   1       ;BYTE COUNT THIS XFR
 (1)  469D  02                                        .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1959  469E                                  XFRP    F10HD2,6,2,2
 (1)  469E  FB   46                                   .ADDR   F10HD2  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  46A0  06                                        .BYTE   6       ;PACKING MODE 6
 (1)  46A1  02                                        .BYTE   2       ;BYTE COUNT THIS XFR
 (1)  46A2  02                                        .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1960  46A3                                  XFRP    F10HD3,6,3,2
 (1)  46A3  07   47                                   .ADDR   F10HD3  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  46A5  06                                        .BYTE   6       ;PACKING MODE 6
 (1)  46A6  03                                        .BYTE   3       ;BYTE COUNT THIS XFR
 (1)  46A7  02                                        .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1961  46A8                                  XFRP    F10HD4,6,4,2
 (1)  46A8  13   47                                   .ADDR   F10HD4  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  46AA  06                                        .BYTE   6       ;PACKING MODE 6
 (1)  46AB  04                                        .BYTE   4       ;BYTE COUNT THIS XFR
 (1)  46AC  02                                        .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1962  46AD                                  XFRP    F10HD5,6,5,2
 (1)  46AD  1F   47                                   .ADDR   F10HD5  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  46AF  06                                        .BYTE   6       ;PACKING MODE 6
 (1)  46B0  05                                        .BYTE   5       ;BYTE COUNT THIS XFR
 (1)  46B1  02                                        .BYTE   2       ;2 SCLKS EXPECTED THIS XFR
1963  46B2                                  XFRP    F10HD6,6,6,4
 (1)  46B2  2B   47                                   .ADDR   F10HD6  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  46B4  06                                        .BYTE   6       ;PACKING MODE 6
 (1)  46B5  06                                        .BYTE   6       ;BYTE COUNT THIS XFR
 (1)  46B6  04                                        .BYTE   4       ;4 SCLKS EXPECTED THIS XFR
1964  46B7  FF   FF                             .ADDR   @177777 ;END OF THE TABLE
1965                              ;HERE IS THE EXPECTED DATA TABLES FOR ALL DATA TRANSFERS
1966
1967  46B9                        F11N:   ;11-NORMAL EXPECTED DATA
1968  46B9  04                              .BYTE   @4      ;DDRA   3 BYTE XFR OF
1969  46BA  08                              .BYTE   @10     ;DDRB   1,2,3,ZERO
1970  46BB  00                              .BYTE   0
1971  46BC  0C                              .BYTE   @14     ;2ND SCLK
1972  46BD  00                              .BYTE   0
1973  46BE  00                              .BYTE   0
1974
1975  46BF                        F15N:   ;15-NORMAL EXPECTED DATA
1976  46BF  08                              .BYTE   @10     ;DDRA   3 BYTE XFR OF
1977  46C0  04                              .BYTE   @4      ;DDRB   1,2,3,ZERO
1978  46C1  00                              .BYTE   0
1979  46C2  00                              .BYTE   0       ;2ND SCLK
1980  46C3  0C                              .BYTE   @14
1981  46C4  00                              .BYTE   0
1982
1983  46C5                        F10C1:  ;10-COMPAT BYTE COUNT =1
1984  46C5  00                              .BYTE   0       ;DATA IS
1985  46C6  04                              .BYTE   @4      ;       1,Z,Z,Z
1986  46C7  00                              .BYTE   0
1987  46C8  00                              .BYTE   0       ;2ND SCLK
```

```
1988  46C9  00                        .BYTE   0
1989  46CA  00                        .BYTE   0
1990
1991  46CB                    F10C2:  ;10-COMPAT BYTE COUNT =2
1992  46CB  08                        .BYTE   @10     ;DATA IS
1993  46CC  04                        .BYTE   @4      ;       1,2,Z,Z
1994  46CD  00                        .BYTE   0
1995  46CE  00                        .BYTE   0       ;2ND SCLK
1996  46CF  00                        .BYTE   0
1997  46D0  00                        .BYTE   0
1998
1999  46D1                    F10C3:  ;10-COMPAT BYTE COUNT =3
2000  46D1  08                        .BYTE   @10     ;DATA IS
2001  46D2  04                        .BYTE   @4      ;       1,2,3,Z
2002  46D3  00                        .BYTE   0
2003  46D4  00                        .BYTE   0       ;2ND SCLK
2004  46D5  C0                        .BYTE   @300
2005  46D6  00                        .BYTE   0
2006  46D7                    F10D1:  ;10-DUMP BYTE COUNT =1
2007  46D7  00                        .BYTE   0       ;DATA IS
2008  46D8  04                        .BYTE   @4      ;       1,Z,Z,Z,Z
2009  46D9  00                        .BYTE   0
2010  46DA  00                        .BYTE   0       ;2ND SCLK
2011  46DB  00                        .BYTE   0
2012  46DC  00                        .BYTE   0
2013
2014  46DD                    F10D2:  ;10-DUMP BYTE COUNT =2
2015  46DD  08                        .BYTE   @10     ;DATA IS
2016  46DE  04                        .BYTE   @4      ;       1,2,Z,Z,Z
2017  46DF  00                        .BYTE   0
2018  46E0  00                        .BYTE   0       ;2ND SCLK
2019  46E1  00                        .BYTE   0
2020  46E2  00                        .BYTE   0
2021
2022  46E3                    F10D3:  ;10-DUMP BYTE COUNT =3
2023  46E3  08                        .BYTE   @10     ;DATA IS
2024  46E4  04                        .BYTE   @4      ;       1,2,3,Z,Z
2025  46E5  00                        .BYTE   0
2026  46E6  00                        .BYTE   0       ;2ND SCLK
2027  46E7  C0                        .BYTE   @300
2028  46E8  00                        .BYTE   0
2029
2030  46E9                    F10D4:  ;10-DUMP BYTE COUNT =4
2031  46E9  08                        .BYTE   @10     ;DATA IS
2032  46EA  04                        .BYTE   @4      ;       1,2,3,4,Z
2033  46EB  00                        .BYTE   0
2034  46EC  00                        .BYTE   0       ;2ND SCLK
2035  46ED  C1                        .BYTE   @3C1
2036  46EE  00                        .BYTE   0
2037
2038  46EF                    F10HD1:
2039  46EF                    F10HC1: ;10-HI-DEN-COMPAT BYTE COUNT =1
2040  46EF  00                        .BYTE   0       ;DATA IS
2041  46F0  04                        .BYTE   @4      ;       1,Z,Z,Z,Z,Z,Z,Z,Z
```

```
2042  46F1  00                              .BYTE   0
2043  46F2  00                              .BYTE   0       ;2ND SCLK
2044  46F3  00                              .BYTE   0
2045  46F4  00                              .BYTE   0
2046  46F5  00                              .BYTE   0       ;3RD SCLK
2047  46F6  00                              .BYTE   0
2048  46F7  00                              .BYTE   0
2049  46F8  00                              .BYTE   0       ;4TH SCLK
2050  46F9  00                              .BYTE   0
2051  46FA  00                              .BYTE   0
2052  46FB                          F10HD2:
2053  46FB                          F10HC2: ;10-HI-DEN-COMPAT BYTE COUNT =2
2054  46FB  08                              .BYTE   @10     ;DATA IS
2055  46FC  04                              .BYTE   @4      ;       1,2,Z,Z,Z,Z,Z,Z,Z,Z
2056  46FD  00                              .BYTE   0
2057  46FE  00                              .BYTE   0       ;2ND SCLK
2058  46FF  00                              .BYTE   0
2059  4700  00                              .BYTE   0
2060  4701  00                              .BYTE   0       ;3RD SCLK
2061  4702  00                              .BYTE   0
2062  4703  00                              .BYTE   0
2063  4704  00                              .BYTE   0       ;4TH SCLK
2064  4705  00                              .BYTE   0
2065  4706  00                              .BYTE   0
2066
2067  4707                          F10HD3:
2068  4707                          F10HC3: ;10-HI-DEN-COMPAT BYTE COUNT =3
2069  4707  08                              .BYTE   @10     ;DATA IS
2070  4708  04                              .BYTE   @4      ;       1,2,3,Z,Z,Z,Z,Z,Z,Z
2071  4709  00                              .BYTE   0
2072  470A  00                              .BYTE   0       ;2ND SCLK
2073  470B  C0                              .BYTE   @300
2074  470C  00                              .BYTE   0
2075  470D  00                              .BYTE   0       ;3RD SCLK
2076  470E  00                              .BYTE   0
2077  470F  00                              .BYTE   0
2078  4710  00                              .BYTE   0       ;4TH SCLK
2079  4.11  00                              .BYTE   0
2080  4712  00                              .BYTE   0
2081
2082  4713                          F10HD4:
2083  4713                          F10HC4: ;10-HI-DEN-COMPAT BYTE COUNT =4
2084  4713  08                              .BYTE   @10     ;DATA IS
2085  4714  04                              .BYTE   @4      ;       1,2,3,4,Z,Z,Z,Z,Z,Z
2086  4715  00                              .BYTE   0
2087  4716  00                              .BYTE   0       ;2ND SCLK
2088  4717  C1                              .BYTE   @301
2089  4718  00                              .BYTE   0
2090  4719  00                              .BYTE   0       ;3RD SCLK
2091  471A  00                              .BYTE   0
2092  471B  00                              .BYTE   0
2093  471C  00                              .BYTE   0       ;4TH SCLK
2094  471D  00                              .BYTE   0
2095  471E  00                              .BYTE   0
```

```
2096  471F                          F10HD5:
2097  471F                          F10HC5: ;10-HI-DEN-COMPAT BYTE COUNT =5
2098  471F  08                             .BYTE   @10     ;DATA IS
2099  4720  04                             .BYTE   @4      ;       1,2,3,4,5,Z,Z,Z,Z
2100  4721  00                             .BYTE   0
2101  4722  00                             .BYTE   0       ;2ND SCLK
2102  4723  C1                             .BYTE   @301
2103  4724  00                             .BYTE   0
2104  4725  00                             .BYTE   0       ;3RD SCLK
2105  4726  40                             .BYTE   @100
2106  4727  01                             .BYTE   1
2107  4728  00                             .BYTE   0       ;4TH SCLK
2108  4729  00                             .BYTE   0
2109  472A  00                             .BYTE   0
2110
2111  472B                          F10HD6:
2112  472B                          F10HC6: ;10-HI-DEN-COMPAT BYTE COUNT =6
2113  472B  08                             .BYTE   @10     ;DATA IS
2114  472C  04                             .BYTE   @4      ;       1,2,3,4,5,6,Z,Z,Z
2115  472D  00                             .BYTE   0
2116  472E  00                             .BYTE   0       ;2ND SCLK
2117  472F  C1                             .BYTE   @301
2118  4730  00                             .BYTE   0
2119  4731  80                             .BYTE   @200    ;3RD SCLK
2120  4732  41                             .BYTE   @101
2121  4733  01                             .BYTE   @1
2122  4734  00                             .BYTE   0       ;4TH SCLK
2123  4735  00                             .BYTE   0
2124  4736  00                             .BYTE   0
2125  4737                          F10HD7:
2126  4737                          F10HC7: ;10-HI-DEN-COMPAT BYTE COUNT =7
2127  4737  08                             .BYTE   @10     ;DATA IS
2128  4738  04                             .BYTE   @4      ;       1,2,3,4,5,6,7,Z,Z
2129  4739  00                             .BYTE   0
2130  473A  00                             .BYTE   0       ;2ND SCLK
2131  473B  C1                             .BYTE   @301
2132  473C  00                             .BYTE   0
2133  473D  81                             .BYTE   @201    ;3RD SCLK
2134  473E  41                             .BYTE   @101
2135  473F  01                             .BYTE   @1
2136  4740  00                             .BYTE   0       ;4TH SCLK
2137  4741  00                             .BYTE   0
2138  4742  00                             .BYTE   0
2139
2140  4743                          F10HD8:
2141  4743                          F10HC8: ;10-HI-DEN-COMPAT BYTE COUNT =8
2142  4743  08                             .BYTE   @10     ;DATA IS
2143  4744  04                             .BYTE   @4      ;       1,2,3,4,5,6,7,10,Z
2144  4745  00                             .BYTE   0
2145  4746  00                             .BYTE   0       ;2ND SCLK
2146  4747  C1                             .BYTE   @301
2147  4748  00                             .BYTE   0
2148  4749  81                             .BYTE   @201    ;3RD SCLK
2149  474A  41                             .BYTE   @101
```

```
2150  474B  01                    .BYTE   @1
2151  474C  00                    .BYTE   @0      ;4TH SCLK
2152  474D  20                    .BYTE   @40
2153  474E  00                    .BYTE   0
```

```
                                .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
474F                            SSUB
 (1)                            ;*****************************************************************************
 (1)                            ;*SUBROUTINE TITLE
 (1)                            ;*---------- -----
                                ;*CLEAR ALL TU PORTS
474F                            SD
 (1)                            ;*****************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                ;*AND LOOP MODES.
474F                            SP
 (1)                            ;*****************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNSUB
                                ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                ;*     CLEAR PORT SELECT FOR TRANSPORT
                                ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                ;*     CLEAR PORT AMTIE WORD
                                ;*ENDSUB
474F                            S
 (1)                            ;*****************************************************************************
474F F5             12.0 CLEAR: PUSH    PSW             ;SAVE THE SELECTED PORT #
4750 C5             12.0        PUSH    B
4751 06 00           7.0        MVI     B,0             ;START TO CLEAR AT PORT #0
4753 DB E0          10.0 CLRLP: IN      INTSTA          ;GET MB SELECT INFO
4755 E6 80           7.0        ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
4757 B0              4.0        ORA     B               ;ADD IN THE SELECTED PORT #
4758 D3 E0          10.0        OUT     MBSEL           ;RESET TO THIS PORT
475A 3E 80           7.0        MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
475C D3 40          10.0        OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
475E AF              4.0        XRA     A               ;CLEAR TU COMMAND A
475F D3 40          10.0        OUT     TCMD            ;
4761 3E 81           7.0        MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
4763 D3 40          10.0        OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
4765 3E 00           7.0        MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
4767 D3 40          10.0        OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
4769 AF              4.0        XRA     A
476A D3 44          10.0        OUT     TAMT            ;CLEAR AMTIE WORD
476C D3 48          10.0        OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
476E D3 4C          10.0        OUT     PENAB           ;CLEAR PORT ENABLE WORD
4770 04              4.0        INR     B               ;POINT TO THE NEXT PORT TO CLEAR
4771 78              4.0        MOV     A,B
4772 FE 04           7.0        CPI     4               ;DONE?
4774 C2 53 47       10.0        JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
4777 C1             10.0        POP     B               ;RESET B & C
4778 F1             10.0        POP     PSW             ;ALL DONE
4779 C9             10.0        RET                     ;EXIT
```

```
                                        .SBTTL  SUBROUTINE 4X5 TRANSLATE A SUBGROUP
477A                                    SSUB
                                        ; **********************************************************************
                                        ;*SUBROUTINE TITLE
                                        ;*---------- -----
                                        ;*4X5 TRANSLATE A SUBGROUP
477A                                    SD
                                        ; **********************************************************************
                                        ;*DESCRIPTION
                                        ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO 4X5 TRANSLATE A 4 BYTE DATA SUBGROUP INTO
                                        ;*A 5 BYTE SUBGROUP AS RECORDED ON TAPE OR READ FROM TAPE.
477A                                    SP
                                        ; **********************************************************************
                                        ;*PROCEDURE
                                        ;*---------
                                        ;*BGNSUB
                                        ;*    GET THE 4 (INPUT) DATA BYTES TO BE TRANSLATED
                                        ;*    CLEAR THE TRANSLATED (OUTPUT) BUFFER (OUTPUT BUFFER IS 9X5)
                                        ;*    SET LOOP COUNT TO 9
                                        ;*    BGNDO
                                        ;*    :    CALCULATE ODD PARITY FOR THE FOUR (INPUT) DATA BYTES (INPUT BUFFER IS NOW 9X4
                                        ;*    :    ROTATE EACH (5) ENTRY OF THE TRANSLATED (OUTPUT) BUFFER RIGHT
                                        ;*    :    GET THE LEAST SIGNIFICANT BIT OF EACH (INPUT) DATA BYTE AND
                                        ;*    :        FORM A TABLE INDEX (MS BIT OF INDEX = LS BIT OF THE FIRST
                                        ;*    :        BYTE OF THE INPUT DATA, ETC.)
                                        ;*    :    REMOVE THE LS BIT OF EACH INPUT DATA BYTE (THE ONE JUST USED
                                        ;*    :        TO BUILD THE INDEX)
                                        ;*    :    USE THE 4 BIT INDEX TO GET THE TRANSLATION VALUE
                                        ;*    :    PUT ONE BIT OF THE TRANSLATION VALUE, AS THE MOST SIGNIFICANT
                                        ;*    :        BIT EACH OF THE 5 TRANSLATED DATA BYTES (LS BIT OF THE
                                        ;*    :        TRANSLATION VALUE REPLACES MS BIT OF THE FIRST TRANSLATED
                                        ;*    :        DATA BYTE.
                                        ;*    :    DECREMENT THE LOOP COUNT
                                        ;*    :    DO UNTIL THE LOOP COUNT = 0
                                        ;*    ENDDO
                                        ;*ENDSUB
477A                                    S
                                        ; **********************************************************************

477A  F5                    12.0  T4X5:   PUSH    PSW             ;SAVE REGISTER A + PSW
477B  C5                    12.0          PUSH    B               ;SAVE REGISTER B + C
477C  D5                    12.0          PUSH    D               ;SAVE REGISTER D + E
477D  E5                    12.0          PUSH    H               ;SAVE REGISTER H + L
                                        ;FOUR DATA BYTES ARE NOW IS THE BUFFER TABLE
                                        ;CLEAR THE TRANSLATED DATA TABLE.
477E  06  0A                 7.0          MVI     B,10            ;SET UP LOOP COUNT
4780  11  10  48              10.0          LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
4783  AF                      4.0          XRA     A               ;CLEAR A
4784  12                      7.0  D4X5:   STAX    D               ;STORE IN THE TABLE
4785  13                      6.0          INX     D               ;UPDATE TABLE POINTER
4786  05                      4.0          DCR     B               ;DECREMENT LOOP COUNT
4787  C2  84  47              10.0          JNZ     D4X5            ;DO UNTIL LOOP COUNT=0
478A  0E  09                  7.0          MVI     C,9             ;SET UP TRACK COUNT
```

```
2243  478C  06  04        7.0  B4X5:  MVI   B,4          ;SET UP BIT COUNT
2244  478E  11  0C  48    10.0         LXI   D,TRNIN
2245  4791  AF             4.0         XRA   A            ;CLEAR THE GROUP POSITION COUNT
2246  4792  32  1A  48    13.0         STA   GP4X5
2247  4795  1A             7.0  C4X5:  LDAX  D            ;GET A DATA BYTE
2248  4796  A7             4.0         ANA   A            ;SET CONDITION BITS
2249  4797  E2  9B  47    10.0         JPO   PO4X5        ;ODD PARITY LEAVE CARRY CLEAR
2250  479A  37             4.0         STC                ;EVEN PARITY SET CARRY
2251  479B  1F             4.0  PO4X5: RAR                ;SHIFT OUT DESIRED BIT
2252  479C  12             7.0         STAX  D            ;STORE RESULT BACK IN TEMP TABLE
2253  479D  3A  1A  48    13.0         LDA   GP4X5        ;GET THE GROUP POSITION BYTE
2254  47A0  17             4.0         RAL                ;PUT IN THIS DATA BIT
2255  47A1  32  1A  48    13.0         STA   GP4X5        ;SAVE THE UPDATED GROUP POSITION BYTE
2256  47A4  13             6.0         INX   D            ;UPDATE THE TABLE POINTER
2257  47A5  05             4.0         DCR   B            ;DECREMENT THE BIT COUNT
2258  47A6  C2  95  47    10.0         JNZ   C4X5         ;DO UNTIL ALL 4 BITS ARE RECEIVED
2259
2260                                ;AT THIS POINT GP4X5 CONTAINS 4 BITS - ONE FROM EACH OF THE FOUR BYTES
2261
2262                                ;PERFORM A DOUBLE PRECISION SHIFT ON THE TRANSLATED DATA
2263  47A9  06  05        7.0         MVI   B,5          ;SET UP LOOP COUNT
2264  47AB  11  10  48    10.0         LXI   D,TRNOUT     ;GET POINTER TO TRANSLATED DATA TABLE
2265  47AE  13             6.0  E4X5:  INX   D            ;POINT TO PARITY BIT
2266  47AF  1A             7.0         LDAX  D            ;GET PARITY BIT
2267  47B0  A7             4.0         ANA   A            ;CLEAR THE CARRY BIT
2268  47B1  1F             4.0         RAR                ;SHIFT TO THE CARRY BIT
2269  47B2  12             7.0         STAX  D            ;STORE IT BACK
2270  47B3  1B             6.0         DCX   D            ;DECREMENT TO DATA BITS
2271  47B4  1A             7.0         LDAX  D            ;GET THE DATA BITS
2272  47B5  1F             4.0         RAR                ;SHIFT IN CARRY BITS
2273  47B6  12             7.0         STAX  D            ;STORE BACK IN TABLE
2274  47B7  13             6.0         INX   D            ;POINT TO NEXT TABLE ENTRY
2275  47B8  13             6.0         INX   D            ;
2276  47B9  05             4.0         DCR   B            ;DECREMENT LOOP COUNT
2277  47BA  C2  AE  47    10.0         JNZ   E4X5         ;DO UNTIL LOOP COUNT=0
2278  47BD  11  EA  47    10.0         LXI   D,TAB4X5     ;LOAD ADDRESS OF TRANSLATION TABLE
2279  47C0  26  00        7.0         MVI   H,0          ;CLEAR REGISTER H
2280  47C2  3A  1A  48    13.0         LDA   GP4X5        ;GET GROUP POSITION COUNT
2281  47C5  6F             4.0         MOV   L,A
2282  47C6  19            10.0         DAD   D            ;ADD GROUP POSITION COUNT TO TABLE
2283  47C7  7E             7.0         MOV   A,M          ;GET TRANSLATED DATA
2284  47C8  32  1B  48    13.0         STA   TRNTMP
2285                                ;"TRNTMP" = THE TRANSLATED DATA
2286
2287  47CB  11  10  48    10.0         LXI   D,TRNOUT     ;GET POINTER TO TRANSLATED DATA TABLE
2288  47CE  06  05        7.0         MVI   B,5          ;SET UP LOOP COUNT
2289  47D0  3A  1B  48    13.0  F4X5:  LDA   TRNTMP       ;GET TRANSLATED DATA
2290  47D3  13             6.0         INX   D            ;POINT TO PARITY POSITION
2291  47D4  A7             4.0         ANA   A            ;CLEAR CARRY BIT
2292  47D5  1F             4.0         RAR                ;SHIFT OUT A BIT
2293  47D6  32  1B  48    13.0         STA   TRNTMP       ;STORE TRANSLATED DATA
2294  47D9  1A             7.0         LDAX  D            ;GET THE PARITY BIT (ALWAYS ZERO)
2295  47DA  17             4.0         RAL                ;ROLL IN THE CARRY BIT
2296  47DB  12             7.0         STAX  D            ;STORE AS THE PARITY BIT
```

```
2297  47DC  13          6.0   INX   D      ;UPDATE TABLE POINTER
2298  47DD  05          4.0   DCR   B      ;DECREMENT THE LOOP COUNT
2299  47DE  C2  D0  47  10.0   JNZ   F4X5   ;DO UNTIL LOOP COUNT=0
2300
2301                           ;NOW THE TRANSLATED DATA TABLE HAS BEEN UPDATED
2302
2303  47E1  0D          4.0   DCR   C      ;DECREMENT THE TRACK COUNT
2304  47E2  C2  8C  47  10.0   JNZ   B4X5   ;DO UNTIL ALL TRACKS TRANSLATED
2305  47E5  E1          10.0   POP   H      ;RESTORE REGISTER H + L
2306  47E6  D1          10.0   POP   D      ;RESTORE REGISTER D + E
2307  47E7  C1          10.0   POP   B      ;RESTORE REGISTER B + C
2308  47E8  F1          10.0   POP   PSW    ;RESTORE REGISTER A + PSW
2309  47E9  C9          10.0   RET          ;RETURN TO USER
```

```
                                .SBTTL  TABLE 4 X 5 TRANSLATION

                                ;THIS TABLE TRANSLATED DATA FOR AS FOLLOWS
                                ;  INPUT GROUP POSITIONS              OUTPUT GROUP POSITIONS
                                ;   1 2 3 4 / 5 6 7 8                 10 9 8 7 6 / 5 4 3 2 1
                                ;
                                ;       0000                              10011
                                ;       0001                              11011
                                ;       0010                              01001
                                ;       0011                              11001
                                ;       0100                              10111
                                ;       0101                              10101
                                ;       0110                              01101
                                ;       0111                              11101
                                ;       1000                              01011
                                ;       1001                              10010
                                ;       1010                              01010
                                ;       1011                              11010
                                ;       1100                              01111
                                ;       1101                              10110
                                ;       1110                              01110
                                ;       1111                              11110

47EA  13                        TAB4X5: .BYTE   @23
47EB  1B                                .BYTE   @33
47EC  09                                .BYTE   @11
47ED  19                                .BYTE   @31
47EE  17                                .BYTE   @27
47EF  15                                .BYTE   @25
47F0  0D                                .BYTE   @15
47F1  1D                                .BYTE   @35
47F2  0B                                .BYTE   @13
47F3  12                                .BYTE   @22
47F4  0A                                .BYTE   @12
47F5  1A                                .BYTE   @32
47F6  0F                                .BYTE   @17
47F7  16                                .BYTE   @26
47F8  0E                                .BYTE   @16
47F9  1E                                .BYTE   @36
```

```
2351                                    .SBTTL  SUBROUTINE VARIABLES
2352
2353  47FA  00                          SDDRA:  .BYTE   0               ;STORAGE FOR DDR A DATA
2354  47FB  00                          SDDRB:  .BYTE   0               ;STORAGE FOR DDR B DATA
2355  47FC  00                          SDDRC:  .BYTE   0               ;STORAGE FOR DDR C DATA
2356  47FD  00                          SCLKCT: .BYTE   0               ;COUNT OF 'SCLK' RECEIVED
2357  47FE  00                          ESCLK:  .BYTE   0               ;NUMBER OF EXPECTED SCLK PER PASS
2358  47FF  00                          FMTNUM: .BYTE   0               ;SELECTED PACKING MODE PER PASS
2359  4800  00                          BCNT:   .BYTE   0               ;NUMBER OF BYTES IN CURRENT XFR
2360  4801  00    00                    EXPADR: .ADDR   0               ;ADDRESS OF THE EXPECTED DATA TABLE
2361  4803  00                          UNITMP: .BYTE   0               ;PORT NUMBER TO USE AS RESPONDED BY HOST
2362  4804  00                          SUNIT:  .BYTE   0               ;STORAGE OF THE SELECTED UNIT NUMBER
2363  4805  00                          CCTLWD: .BYTE   0               ;LOAD INTO THE SYSTEM CLOCK CONTROL WORD
2364  4806  00                          HAVEXFR: .BYTE  0               ;FLAG TO INDICATE IF 'XFER' EVER SEEN
2365  4807  00                          ONCE:   .BYTE   0               ;DATA XFR ONCE FLAG ... FOR GRONKING THE WMC
2366
2367  4808  00    00                    PTABLE: .ADDR   0               ;ADDRESS OF CURRENT XFR PARAMETER TABLE
2368  480A  00    00                    NPTABLE: .ADDR  0               ;ADDRESS OF NEXT PTABLE
2369
2370  480C 0004                         TRNIN:  .BLKB   4               ;TABLE CONTAINING THE 4 DATA BYTES TO
2371                                                                    ;BE TRANSLATED
2372  4810 000A                         TRNOUT: .BLKB   10              ;TRANSLATED DATA TABLE DATA AFTER 4X5
2373                                                                    ;TRANSLATION
2374  481A  00                          GP4X5:  .BYTE   0               ;ACCUMULATOR FOR THE 4 BIT SUBGROUP TO
2375                                                                    ;BE TRANSLATED
2376  481B  00                          TRNTMP: .BYTE   0               ;TEMP. STORAGE FOR THE TRANSLATED SUBGROUP.
```

```
2378                                    .SBTTL  SUBROUTINE CLEAR ECC
2379  481C                              $
 (1)                                    ; ************************************************************************
2380                                    ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION "ECCCHR" -
2381                                    ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
2382                                    ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
2383  481C                              $
 (1)                                    ; ************************************************************************
2384
2385  481C  AF                   4.0  CLECC:  XRA     A              ;CLEAR THE ACCUMULATOR
2386  481D  32   48   48        13.0         STA     ECCCHR         ;CLEAR THE ECC CHARACTER
2387  4820  C9                  10.0         RET                    ;RETURN TO USER
```

```
                                        .SBTTL  SUBROUTINE CALCUALTE ECC CHARACTER
                                        $
                                        ; ************************************************************************
                                        ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
                                        ;THE CONTENTS OF "ECCCHR" TO UPDATE THE CONTENTS OF "ECCCHR" ACCORDING
                                        ;TO THE ANSI STANDARD ECC POLYNOMIAL.
                                        $
                                        ; ************************************************************************

4821  E5              12.0  ECC:   PUSH   H              ;SAVE THE ADDRESS POINTER IN H & L
4822  C5              12.0         PUSH   B              ;SAVE B & C
4823  21  48  48      10.0         LXI    H,ECCCHR       ;LOAD ADDRESS OF ECC CHAR.
4826  AE               7.0         XRA    M              ;EXCLUSIVE OR CHAR. AND ECC
4827  5F               4.0         MOV    E,A            ;SAVE XOR RESULT IN E
4828  E6  10           7.0         ANI    $10            ;IS BIT #4 OF RESULT SET
482A  7B               4.0         MOV    A,E            ;RESTORE XOR RESULT FROM B
482B  CA  30  48       10.0         JZ     ECC1           ;CONTINUE IF BIT #4 RESET
482E  EE  23           7.0         XRI    $23            ;ELSE-XOR WITH 23
4830  5F               4.0  ECC1:  MOV    E,A            ;STORE THE ECC RESULT IN E
4831  AF               4.0         XRA    A              ;CLEAR A
4832  4F               4.0         MOV    C,A            ;CLEAR THE TRANSLATE RESULT
4833  21  40  48       10.0         LXI    H,ECCTBL       ;POINT TO ECC TABLE TO RE-POSITION
4836  CD  49  48       18.0         CALL   TRANS          ;TRANSLATE THE BITS
4839  79               4.0         MOV    A.C            ;GET THE TRANSLATED RESULT
483A  32  48  48       13.0         STA    ECCCHR         ;STORE RESULT
483D  C1               10.0         POP    B
483E  E1               10.0         POP    H              ;RESET H & L
483F  C9               10.0         RET

4840  08                      ECCTBL: $08                   ;BIT 0 = POSITION 3
4841  20                              $20                   ;BIT 1 = POSITION 5
4842  02                              $02                   ;BIT 2 = POSITION 1
4843  40                              $40                   ;BIT 3 = POSITION 6
4844  80                              $80                   ;BIT 4 = POSITION 7
4845  01                              $01                   ;BIT 5 = POSITION 0
4846  10                              $10                   ;BIT 6 = POSITION 4
4847  04                              $04                   ;BIT 7 = POSITION 2

4848  00                      ECCCHR: .BYTE  0
```

```
                                        .SBTTL  SUBROUTINE POLYNOMIAL BIT TRANSLATION

                                        ;
                                        ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
                                        ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
                                        ;

4849  06  01          7.0   TRANS:  MVI   B,1             ;INIT 'B' TO BIT POSITION 0
484B  7B                4.0   TRANS1: MOV   A,E             ;GET CHAR TO BE TRANSLATED
484C  A0                4.0           ANA   B               ;SEE IF BIT POSITION IN 'B' IS SET
484D  CA  53  48        10.0          JZ    TRANS2          ;DO NEXT BIT POSITION IF NOT SET
4850  79                4.0           MOV   A,C             ;GET PREVIOUS RESULT OF 'OR'
4851  B6                7.0           ORA   M               ;'OR' IN NEW POSITION
4852  4F                4.0           MOV   C,A             ;SAVE RESULT

4853  78                4.0   TRANS2: MOV   A,B
4854  07                4.0           RLC                   ;POSITION MASK TO NEXT BIT
4855  47                4.0           MOV   B,A
4856  D8               12.0           RC                    ;EXIT WHEN ALL POSITIONS DONE
4857  23                6.0           INX   H               ;POINT TO NEXT TABLE ENTRY
4858  C3  4B  48       10.0           JMP   TRANS1          ;PROCESS NEXT BIT

485B                                  S
 (1)                                  ; *********************************************************************
                                      ;CLKSYS - ROUTINE TO CLOCK THE SYSTEM USING THE CONTENTS OF 'CCTLWD'
                                      ;       AS A BASE AND TOGGLING THE CLOCK BIT IN THE CLOCK CONTROL WORD
485B                                  S
 (1)                                  ; *********************************************************************

485B  F5               12.0   CLKSYS: PUSH  PSW
485C  3A  05  48       13.0           LDA   CCTLWD          ;GET THE CONTROL WORD BASE
485F  F6  40            7.0           ORI   SSCLK           ;GET THE CLOCK BIT
4861  D3  F0           10.0           OUT   CLKCTL          ;SET THE CLOCK ON
4863  E6  BF            7.0           ANI   @277            ;CLEAR THE CLOCK BIT
4865  D3  F0           10.0           OUT   CLKCTL          ;FINISH THE CLOCK CYCLE
4867  F1               10.0           POP   PSW
4868  C9               10.0           RET

      0000                      .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE    4F9B          ATTCD    4F97         AXNUM    4F91
B      =%0000          BADST  = 0090          BCNT     4800         BIT0   = 0001
BIT1   = 0002          BIT15  = 8000          BIT2   = 0004         BIT3   = 0008
BIT4   = 0010          BIT5   = 0020          BIT6   = 0040         BIT7   = 0080
BIT8   = 0100          BIT9   = 0200          BRKPBC= 4F0A          BRKRAM= 4F10
BRKSTR= 4E60           BRKXCT= 4F00          BSAVE    4F9C         BYTCNT= 00D4
BYTEH    4F24          BYTEL    4F23          B4X5     478C         C      =%0001
CASCT    4F21          CASCTL= 00A0          CASSTA= 00A0          CATTH  = 0089
CATTL  = 0088          CBUSST= 00A1          CBYTH  = 008B         CBYTL  = 008A
CCTLWD   4805          CDG1H  = 0087         CDG1L  = 0086         CDG2H  = 0093
CDG2L  = 0092          CDG3H  = 0095         CDG3L  = 0094         CDVTH  = 008D
CDVTL  = 008C          CHPTIE= 0028          CH0TIE= 0020          CH1TIE= 0021
CH2TIE= 0022           CH3TIE= 0023          CH4TIE= 0024          CH5TIE= 0025
CH6TIE= 0026           CH7TIE= 0027          CKLOP  = 2815         CLEAR    474F
CLECC    481C          CLKCTL= 00F0          CLKRS    4419         CLKSYS   485B
CLOCK    4F26          CLRLP    4753         CMC0H  = 0099         CMC0L  = 0098
CMC1H  = 009B          CMC1L  = 009A         CMC2H  = 009D         CMC2L  = 009C
CMC3H  = 009F          CMC3L  = 009E         CMINH  = 0097         CMINL  = 0096
CNTCTL= 00D7           CNV1     4618         CONVRT   4612         CRCECC   464D
CRCGRP   4646          CRCG2    464A         CRCWRD= 0018          CSAVE    4F9D
CSRLH  = 0091          CSRLL  = 0090         CTCH   = 0085         CTCL   = 0084
CTSTH  = 008F          CTSTL  = 008E         CXCTH  = 0081         CXCTL  = 0080
CXINH  = 0083          CXINL  = 0082         C.     = 0001         C.AVAI= 0080
C.DP   = 0008          C.DSE  = 0010         C.DTU  = 0003         C.DVA  = 0008
C.FAIL= 00FC           C.FMT  = 0070         C.FNCT= 003E          C.GO   = 0001
C.INTC= 00FE           C.MAIN= 0020          C.NSA  = 0080         C.RCT  = 00FC
C.SER  = 0080          C.SHR  = 0040         C.SKPC= 000F          C.TAPE= 0040
C.WCS  = 0002          C1EC1    4424         C1EC1L   442C         C1EC2    443B
C1EC2L   4443          C4X5     4795         D      =%0002         DATACT= 00D0
DBUS     4F28          DBUSCT= 00C0          DBUSST= 00C0          DDRA   = 00D8
DDRAIN= 0010           DDRB   = 00D9         DDRBIN= 0002          DDRC   = 00DA
DDRC1N= 0001           DDRCO  = 0088         DDRCTL= 00DB          DIAFLG   4F22
DIAGPG= 4300           DIAGRM= 4F90          DIARD  = 000B         DONE1  = 0045
DONINT= 0010           DSAVE    4F9E         DSE    = 0006         DUMM1    4507  G
DUMM1A   44FC  G       DUMM2    4543  G      D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010           D.LAGC= 0020          D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008           D.WR4  = 0080         D4X5     4784         E      =%0003
ECC      4821          ECCBAD= 0042          ECCCHR   4848         ECCCOR= 0019
ECCOK  = 0041          ECCSTA= 001A          ECCTBL   4840         ECCTST= 000E
ECC1     4830          EDATA  = 0095         EOTCLR= 0003          ERFLG    4F93
ERLP   = 2809          ERLPA  = 280F         ERLPB  = 2812         ERLPE  = 280C
ERNUM    4F90          ERRCNT= 00D6          ESAVE    4F9F         ESCLK    47FE
EXIT     45F8          EXPADR   4801         E.ACRC= 0010          E.AMT  = 0020
E.CDP  = 0080          E.CRC  = 0080         E.PNTR= 0008          E.RPE  = 0040
E.STEC= 0001           E.TTEC= 0002          E.UNC  = 0004         E4X5     47AE
FIFORD= 006A           FMTNUM   47FF         FORMAT   4F25         FOUND    4327
FPORT    4347          FWDTST= 0061          F10C1    46C5         F10C2    46CB
F10C3    46D1          F10D1    46D7         F10D2    46DD         F10D3    46E3
F10D4    46E9          F10HC1   46EF         F10HC2   46FB         F10HC3   4707
F10HC4   4713          F10HC5   471F         F10HC6   472B         F10HC7   4737
F10HC8   4743          F10HD1   46EF         F10HD2   46FB         F10HD3   4707
F10HD4   4713          F10HD5   471F         F10HD6   472B         F10HD7   4737
F10HD8   4743          F11N     46B9         F15N     46BF         F4X5     47D0
```

```
GCRID = 0089          GCRSET= 0002          GOODTM= 0092          GP4X5   481A
H     =%0004          HAVEXF  4806          HLSAVE  4FA0          IE    = 0008
INTSTA= 00E0          ITERA   4F9A          I.PWR = 0020          I.RMPE= 0040
I5.5  = 0010          I6.5  = 0020          I7.5  = 0040          KCALL = 005F
KCLR  = 007B          KDEP  = 003F          KENAB = 0078          KEXAM = 003E
KEYSRD= 00C8          KEY1  = 0078          KEY10 = 006D          KEY11 = 006E
KEY12 = 006F          KEY13 = 005C          KEY14 = 005D          KEY15 = 005E
KEY16 = 005F          KEY17 = 003C          KEY18 = 003D          KEY19 = 003E
KEY2  = 0079          KEY20 = 003F          KEY3  = 007A          KEY4  = 007B
KEY5  = 0074          KEY6  = 0075          KEY7  = 0076          KEY8  = 0077
KEY9  = 006C          KINTA = 006F          KLDAD = 003D          KN0   = 003C
KN1   = 005C          KN2   = 005D          KN3   = 005E          KN4   = 006C
KN5   = 006D          KN6   = 006E          KN7   = 0074          KN8   = 0075
KN9   = 0076          KU2   = 0079          KU3   = 007A          KU8   = 0077
L     =%0005          LBLANK= 000F          LCE   = 000B          LCH   = 000C
LCL   = 000D          LCP   = 000E          LC0   = 0000          LC1   = 0001
LC2   = 0002          LC3   = 0003          LC4   = 0004          LC5   = 0005
LC6   = 0006          LC7   = 0007          LC8   = 0008          LC9   = 0009
LDFIFO  462C          LDLEDA= 00CA          LDLEDB= 00CB          LDLEDC= 00CC
LDLEDD= 00CD          LDLEDE= 00CE          LDLEDF= 00CF          LKDIAG= 2800
LKKBD = 004C          LKKEY = 0049          LKLWMG= 0058          LKLWMP= 0055
LKLWPG= 0052          LKLWPP= 004F          LKMOD7= 0046          LKOPR = 0046
LOADIT  4627          LPFLG   4F94          LPNUM   4F92          M     =%0006
MBSEL = 00E0          MB.A  = 0008          MB.B  = 0004          MEMTOP= 4FFF
MINUS = 000A          MM    = 8000          MSE   = 0008          MSGN  = 0007
MTACLR= 0000          MT.ARA= 0020          MT.CPE= 0080          MT.DSE= 0001
MT.FWD= 0040          MT.INH= 0008          MT.LWR= 0004          MT.MOT= 0002
MT.NWT= 0080          MT.PEC= 0040          MT.PSB= 0004          MT.PS0= 0001
MT.PS1= 0002          MT.REV= 0020          MT.WRT= 0010          MT.Z  = 0008
M.ATA = 0080          M.CAPE= 0020          M.CONT= 0080          M.DEM = 0020
M.EBL = 0010          M.EXC = 0008          M.FAIL= 0008          M.ILR = 0010
M.INIT= 0010          M.OCC = 0020          M.ONLI= 0001          M.PE  = 0040
M.PORT= 0080          M.RDEN= 0002          M.RDPE= 0008          M.RUN = 0004
M.SCLK= 0001          M.TRA = 0040          M.UNIT= 0007          M.WCLK= 0040
M.WCLN= 0080          M.WREN= 0080          M5.5  = 0001          M6.5  = 0002
M7.5  = 0004          NOTCAP= 0088          NOUNIT  431E  G       NPTABL  480A
OKAY  = 00FF          ONCE    4807          OPRRAM= 4300          OPSTRT= 0058
OPVER = 0040          PADCNT= 00D5          PADCRC= 0080          PDIAG = 0048
PEID  = 008A          PENAB = 004C          PESET = 0001          PL    = 00B1
PO4X5   479B          PRDD  = 004C          PRENF = 009C          PS    = 00B2
PSTAT = 0048          PSW   =%0009          PTABLE  4808          P.AMTP= 0001
P.BCTC= 0040          P.CMDP= 0020          P.INTE= 0080          P.LCS = 0040
P.LWR = 0020          P.RDP = 0002          P.RPEN= 0004          P.RPST= 0002
P.RPOE= 0020          P.RP1E= 0010          P.RP2E= 0020          P.RP3E= 0010
P.SING= 0080          P.STAT= 0002          P.STPE= 0080          P.TACH= 0008
P.TUPR= 0010          P.WCSP= 0004          P.WDS = 0040          P.WFLP= 0001
P.WPEN= 0010          P.WPOE= 0008          P.WP1E= 0004          P.WP2E= 0008
P.WP3E= 0004          P.5VOK= 0002          QUE   = 281B          QUEM  = 281E
RAMT  = 0010          RARA  = 0006          RARAI = 0004          RCHBD0= 0048
RCHBD1= 0047          RCHOK = 0046          RCHRWD  464C          RCHTST= 000C
RCLRT = 000D          RCMD  = 000B          RCMLP = 0003          RCONT = 0080
RDATA = 0017          RDCLK = 0010          RDON  = 0011          READG = 0007
REND  = 0014          REQST = 2806          RESCHR= 00D1          REVTST= 0064
REWIND= 0004          RFIFOL= 0008          RGCLK = 0002          RGCRI = 0003
```

```
RGECC   4645          RGRP    463E          RGRP2   4642          RIBG  = 0001
RILL  = 0012          RINST = 000C          RMCTST= 0008          RMK2  = 0013
RNOP  = 0000          RPATH = 0001          RPBAD = 0044          RPCHI = 0001
RPCLK = 0003          RPCTL = 0009          RPEI  = 0002          RPFAIL= 0000
RPF1  = 009D          RPF2  = 009E          RPOK  = 0043          RPOSTN= 0016
RPSTA = 0015          RRCMT = 000A          RSTAT = 0002          RTIEB = 000A
RTIER = 0030          RTM   = 0005          RUNKI = 0009          RUPTST= 005E
RWDUNL= 0005          R.AMT = 0001          R.BOP = 0008          R.DATA= 0040
R.DON = 0002          R.DRDY= 0010          R.END = 0010          R.ILL = 0004
R.JVOK= 0004          R.MK2 = 0008          R.PLOD= 0008          R.PLOO= 0010
R.PLO1= 0020          R.POST= 0020          R.STNM= 0002          R.STOP= 0004
R.STPC= 0001          R.TBJN= 0080          R.TSTD= 0040          R.VOK = 0080
R00H  = 0081          R00L  = 0080          R01H  = 0083          R01L  = 0082
R02H  = 0085          R02L  = 0084          R03H  = 0087          R03L  = 0086
R04H  = 0089          R04L  = 0088          R05H  = 008B          R05L  = 008A
R06H  = 008D          R06L  = 008C          R07H  = 008F          R07L  = 008E
R10H  = 0091          R10L  = 0090          R11H  = 0093          R11L  = 0092
R12H  = 0095          R12L  = 0094          R13H  = 0097          R13L  = 0096
R14H  = 0099          R14L  = 0098          R15H  = 009B          R15L  = 009A
R16H  = 009D          R16L  = 009C          R17H  = 009F          R17L  = 009E
R7.5  = 0010          SCLKCT  47FD          SDDRA   47FA          SDDRB   47FB
SDDRC   47FC          SELCLR= 0000          SETATA= 00A1          SID   = 0080
SOD   = 0080          SOE   = 0040          SP    =%0008          SSCLK = 0040
SSTEP = 0005          STACK = 4FFF          STATRM= 4F20          STPCT   4F20
STRSP = 5000          SUNIT   4804          TAB4X5  47EA          TADR00= 0080
TADR01= 0081          TADR02= 0082          TADR03= 0083          TADR04= 0084
TADR05= 0085          TADR06= 0086          TADR07= 0087          TADR10= 0088
TADR11= 0089          TADR12= 008A          TADR13= 008B          TAMT  = 0044
TASEL = 0080          TCMD  = 0040          TC.FWD= 0040          TC.INH= 0008
TC.LWR= 0004          TC.REV= 0020          TC.WRT= 0010          TEMP    4F99
TEST1   4300          TMF   = 0099          TMRDY = 0040          TRANS   4849
TRANS1  484B          TRANS2  4853          TRKENA= 00D2          TRNIN   480C
TRNOUT  4810          TRNTMP  481B          TSET  = 2803          TSTEND= 2818
TSTS  = 0040          TUSEL0= 00D1          TUSEL1= 00D2          TU78  = 0010
T.ATH0= 0001          T.ATH1= 0002          T.BOT = 0004          T.EOT = 0002
T.FPT = 0001          T.NTHR= 0004          T.ONL = 0020          T.PES = 0008
T.PSBJ= 0020          T.PSOJ= 0008          T.PS1J= 0010          T.RDY = 0080
T.RDYO= 0040          T.RWD = 0010          T.SCLK= 0002          T1CKA   4573
T1CKAC  459C  G       T1CKB   45A2          T1CKBC  45B6  G       T1CKC   45BC
T1CKCC  45D0  G       T1CLK   44CB          T1CLKA  44C1          T1CLK1  44DB
T1CLK2  44E5          T1CLK3  44EC          T1EMK   4468          T1EMKL  446D
T1ERNS  4502          T1FOK   43CF          T1LOOP  436D          T1MK2   44A0
T1MK2L  44A5          T1M2L2  44B4          T1PLO   4452          T1RCLR  43B5
T1RFIL  43C0          T1SCKE  453E          T1SCKL  4513          T1SCKO  4549
T1SCLK  450D          T1SG1   447C          T1SG2   4485          T1SG3   448E
T1SG4   4497          T1ST    4422          T4X5    477A          UIBG  = 00A1
UNITMP  4803          VALFC   4F98          VALTB   4F95          VELTST= 005B
WDR.P = 0010          WMCCTL= 00D3          WMCERR= 00DA          WMCSTA= 00D0
WRTCLK= 0000          WRTDAT= 00D3          W.ACRC= 0004          W.CRC = 0008
W.DIAG= 0002          W.DONN= 0040          W.ECC = 0010          W.ENAB= 0080
W.ERR = 0020          W.FMT = 0070          W.GCR = 0010          W.LEFT= 0004
W.ONES= 0020          W.RESI= 0002          W.REV = 0004          W.ROME= 0010
W.RST = 0001          W.SKIP= 000F          W.WRIT= 0008          W.XFER= 0020
X     =%000A          XFRPTB  464E          X.DONN= 0080          X.ENAB= 0040
```

```
X.PEPE= 0002            X.ROME= 0001            X.WCLK= 0001            Y      =%000B
.     = 4869


 ERRORS DETECTED:  0

 *WMC7.A78/PTP,WMC7=NLIST,PARAM,MACRO,LIST,WMC7
 RUN-TIME: 4 6 0 SECONDS
 COPE USED:  10K
```

3	'MEMORY MAP'' EXTENDED ADDRESS BIT
19	READ MICRO CONTROLLER CONTROL & STATUS WORDS
139	TRANSPORT/PORT COMMAND + STATUS WORDS
380	COMMON ADDRESS SPACE REGISTERS
444	MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485	MASSBUS CONTROL WORD DEFINITIONS
557	WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709	KEYBOARD/DISPLAY CONTROL + STATUS WORD
797	MISCELLANEOUS VARIABLE DEFINITIONS
801	RAM ADDRESS DEFINITIONS
819	LINKAGE TABLE ADDRESSES
842	BIT DEFINITIONS
856	RIM AND SIM INSTRUCTION BIT DEFINITIONS
877	INTERRUPTABLE INSTRUCTION MACROS
924	DIAGNOSTIC VARIABLES
945	COMMON VARIABLE STORAGE
959	MACRO CALL DEFINITIONS
1047	TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101	8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1333	TEST 1 - WMC BYTE ASSEMBLY - READ FORWARD
2034	SUBROUTINE CLEAR ALL TU PORTS
2078	SUBROUTINE 4X5 TRANSLATE A SUBGROUP
2190	TABLE 4 X 5 TRANSLATION
2230	SUBROUTINE VARIABLES
2252	SUBROUTINE CLEAR ECC
2263	SUBROUTINE CALCUALTE ECC CHARACTER
2301	SUBROUTINE POLYNOMIAL BIT TRANSLATION

```
961                 ;*************************************************************************
962                 ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
963                 ;ERROR MACRO CALLS
964                 ;
965                 ;       1  -    REQUEST HOST CPU TO PRINT:
966                 ;               "BYTE/SCLK COUNT NUMBER = LLL"
967                 ;               WHERE:
968                 ;               LLL = THE VALUE STORED IN CAS REGISTER 5
969                 ;                     (THE BYTE COUNT REGISTER 16 BITS).
970                 ;
971                 ;       2  -    REQUEST HOST CPU TO PRINT:
972                 ;               DATA FORMAT = MM
973                 ;               SKIP COUNT  = NN
974                 ;               WHERE:
975                 ;               MM = DATA FORMAT FROM CAS REGISTER 2
976                 ;               NN = SKIP COUNT FROM CAS REGISTER 2
977                 ;
978                 ;       3  -    REQUEST HOST CPU TO PRINT:
979                 ;               BYTE-SCLK COUNT = LLL
980                 ;               DATA FORMAT = MM
981                 ;               SKIP COUNT  = NN
982                 ;               WHERE:
983                 ;               LLL = AS ABOVE
984                 ;               MM  = AS ABOVE
985                 ;               NN  = AS ABOVE
986                 ;
987                 ;       4  -    REQUEST HOST TO PRINT:
988                 ;               TRANSITION COUNT = LLL
989                 ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
990                 ;
991                 ;       5  -    REQUEST HOST CPU TO PRINT:
992                 ;               EXPECTED 18 BITS =      E EEEEEE
993                 ;               ACTUAL 18 BITS =        A AAAAAA
994                 ;
995                 ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
996                 ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
997                 ;               OF CAS REG 15 LOW BYTE.
998                 ;
999                 ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
1000                ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
1001                ;               17 LOW BYTE SIGN BIT.
1002                ;
1003                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
1004                ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
1005                ;               AND/OR ACTUAL DATA.
1006                ;
1007                ;       7  -    REQUEST HOST CPU TO PRINT:
1008                ;               "SUBGROUP NUMBER = LLL"
1009                ;               WHERE:
1010                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
1011                ;                     (THE BYTE COUNT REGISTER 16 BITS).
1012                ;
1013                ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
1014                ;*************************************************************************
```

```
1016                    ;*****************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ;*****************************************************************
1035                    ;*****************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ;*****************************************************************
1329    4300                    .-      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  WMC8 - BYTE ASSEMBLY/SKIP COUNT OF READ PATH DATA
                                .SBTTL  TEST 1 - WMC BYTE ASSEMBLY - READ FORWARD
                                ;ID     WMC8-WRITE MICRO CONTROLLER PART #8

1336  4300                      ST
 (1)                            ;*****************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1337                            ;*WMC BYTE ASSEMBLY/SKIP COUNT TEST FROM READ FORWARD DATA
1338  4300                      SD
 (1)                            ;*****************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1339                            ;*THIS TEST WILL INJECT DATA THROUGH THE READ PATH FIFO'S AND CLOCK THE
1340                            ;*SYSTEM UNTIL AN 'SCLK' APPEARS TO SIGNAL 18 BITS OF DATA READY FOR
1341                            ;*TRANSMISSION ACROSS THE MASSBUS. EACH 'SCLK' TIME WILL HAVE THE 'DDR'
1342                            ;*REGISTERS SETUP WITH THE ASSEMBLED DATA AS COMMANDED BY THE PACKING
1343                            ;*MODE SELECTED IN THE TEST. ALL PACKING MODES ARE TESTED (I.E. 11-NORM
1344                            ;*TO 10-HI DEN COMPAT) IN A READ FORWARD DATA TRANSFER. THE TEST WILL
1345                            ;*CLOCK THE SYSTEM ONLY UNTIL THE EXPECTED AMOUNT OF 'SCLK' DATA HAS
1346                            ;*BEEN TESTED TO VERIFY THAT THE SELECTED PACKING MODE IS WORKING AS
1347                            ;*EXPECTED. THE READ PATH WILL BE FILLED WITH 8 SUBGROUPS WORTH OF DATA
1348                            ;*REGUARDLESS OF THE AMOUNT OF EXPECTED DATA GROUPS TO BE TESTED (USUALLY
1349                            ;*4 SUBGROUPS OF DATA WILL BE SUFFICIENT TO TEST A PACKING MODE). THE EXTRA
1350                            ;*SUBGROUPS OF DATA INJECTED WILL ELIMINATE THE NECESSITY OF CALCULATING
1351                            ;*THE 'SPECIAL' CHARACTERS FROM THE TAPE IN GCR MODE (I.E. CRC, ACRC, &
1352                            ;*RESID CHARACTERS). EVERY 4 BYTES OF DATA TO BE INJECTED WILL BE TURNED
1353                            ;*INTO A 5 BYTE SUBGROUP TO SIMULATE DATA FROM THE 'TAPE'. THE 'ECC'
1354                            ;*WILL BE CALCULATED FOR EACH 7 BYTE OF DATA INJECTED AND ALSO INJECTED
1355                            ;*AT THE PROPER TIME INTO THE FIFO'S.
1356  4300                      SP
 (1)                            ;*****************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1357                            ;*BGNTST
1358                            ;*  REQUEST A TU PORT FROM 'HOST' CPU
1359                            ;*  IF NO TU PORT SELECTED FROM 'HOST' CPU
1360                            ;*  :  THEN-REPORT OPERATOR ERROR
1361                            ;*  :  ELSE-PROCEED
1362                            ;*  ENDIF
1363                            ;*  BGNDO
1364                            ;*  :  CLEAR ALL PORTS
1365                            ;*  :  INIT THE SYSTEM
1366                            ;*  :  SET READ PATH TO NORMAL CLOCK SPEED
1367                            ;*  :  INIT TRANSFER TABLE PARAMETER POINTER - PTABLE
1368                            ;*  :  LOAD SELECTED SKIP COUNT & PACKING MODE
1369                            ;*  :  INIT COUNTERS + CONTROLS
1370                            ;*  :  LOAD BYTE COUNTER + PAD COUNTER
1371                            ;*  :  RESTART WRITE MICROCONTROLLER
1372                            ;*  :  CLOCK WMC THROUGH RESTART CODE
1373                            ;*  :  START 'GCR' READ DATA XFR
1374                            ;*  :  CALCULATE ALL NECESSARY ECC CHARACTERS FOR INJECTION
1375                            ;*  :  SET READ PATH PLO BYPASS MODE 2, TIE BUS JAM AND PLO DISABLE
1376                            ;*  :  CLOCK THE FIFO'S
```

```
1377                                    ;*  :  ISSUE 'CLEAR ALL' TO READ PATH
1378                                    ;*  :  WAIT
1379                                    ;*  :  STOP THE READ PATH, SET TIE BUS JAM AND PLO DISABLE
1380                                    ;*  :  CONVERT SUBGROUPS TO 4X5 & LOAD INTO FIFO
1381                                    ;*  :  SETUP A WATCHDOG TIMER
1382                                    ;*  :  CLOCK SYSTEM TO XFR DATA FROM FIFO TO WMC BYTE ASSY LOGIC
1383                                    ;*  :  IF WATCHDOG TIMER=0
1384                                    ;*  :  :  THEN-REPORT NO 'SCLK'
1385                                    ;*  :  :  ELSE-CONT
1386                                    ;*  :  ENDIF
1387                                    ;*  :  CLOCK SYSTEM TILL SCLK=0...DATA READY TO BE TESTED
1388                                    ;*  :  GET 'SCLK' 18 BIT DATA FROM BYTE ASSY INTO 'DDR' REGISTER
1389                                    ;*  :  IF 18 BIT DATA SAME AS EXPECTED
1390                                    ;*  :  :  THEN-PROCEED
1391                                    ;*  :  :  ELSE-REPORT BYTE ASSY ERROR + WHAT WAS ASSEMBLED
1392                                    ;*  :  ENDIF
1393                                    ;*  :  POINT TO NEXT PARAMETER TABLE ENTRY SET
1394                                    ;*  :  IF ALL DONE WITH DATA TRANSFERS IN 'PTABLE'
1395                                    ;*  :  :  THEN - DECREMENT THE ITERATION COUNTER
1396                                    ;*  :  :  :  IF ALL ITERATIONS HAVE BEEN EXECUTED
1397                                    ;*  :  :  :  :  THEN-EXIT
1398                                    ;*  :  :  :  :  ELSE-CONTINUE
1399                                    ;*  :  :  :  ENDIF
1400                                    ;*  :  :  ELSE-CONTINUE
1401                                    ;*  :  :  ENDIF
1402                                    ;*  :  :  DO UNTIL ALL ITERATIONS EXECUTED
1403                                    ;*  :  ENDDO
1404                                    ;*ENDTST
1405  4300                              SE
 (1)                                    ; ************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1406                                    ;*WMC8 MICRO TEST 01
1407                                    ;*WMC8 MICRO ERROR 01
1408                                    ;*WMC BYTE ASSY FOR READ FORWARD XFR
1409                                    ;*M8959 (WMC) & ALL OTHER BOARDS
1410                                    ;*OPERATOR ERROR - NO TU78 UNIT SELECTED
1411                                    ;*FATAL ERROR - TEST ABORTED
1412                                    ;*
1413                                    ;*WMC8 MICRO TEST 01
1414                                    ;*WMC8 MICRO ERROR 02
1415                                    ;*WMC BYTE ASSY FOR READ FORWARD XFR
1416                                    ;*M8959 (WMC) & ALL OTHER BOARDS
1417                                    ;*TIMEOUT WHILE WAITING FOR 'SCLK' TO SET IN DBUSSTA
1418                                    ;*AFTER ISSUEING A READ FORWARD COMMAND TO THE TM78
1419                                    ;*FATAL ERROR - TEST ABORTED
1420                                    ;*
1421                                    ;*WMC8 MICRO TEST 01
1422                                    ;*WMC8 MICRO ERROR 03
1423                                    ;*WMC BYTE ASSY FOR READ FORWARD XFR
1424                                    ;*M8959 (WMC) & ALL OTHER BOARDS
1425                                    ;*TIMEOUT WHILE WAITING FOR 'SCLK' TO CLEAR AFTER SETTING
1426                                    ;*DURING A READ FORWARD COMMAND
1427                                    ;*FATAL ERROR - TEST ABORTED
```

```
                                        ;*
                                        ;*WMC8 MICRO TEST 01
                                        ;*WMC8 MICRO ERROR 04
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 7-0
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*WMC8 MICRO TEST 01
                                        ;*WMC8 MICRO ERROR 05
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 15-8
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
                                        ;*
                                        ;*WMC8 MICRO TEST 01
                                        ;*WMC8 MICRO ERROR 06
                                        ;*WMC BYTE ASSY FOR READ FORWARD XFR
                                        ;*M8959 (WMC) & ALL OTHER BOARDS
                                        ;*ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 17-16
                                        ;*ACTUAL = NNNN
                                        ;*EXPECTED = NNNN
1452  4300                              $
 (1)                                    ; ***************************************************************************
1453
1454  4300                              TEST1:  TESTX   @1
 (1)  4300  3E  01            7.0                MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4302  CD  03  28       18.0                CALL    TSET            ;SETUP THE TEST
1455                                    ;%WMC BYTE ASSY FOR READ FORWARD XFR
1456                                    ;&M8959 (WMC) & ALL OTHER BOARDS
1457                                    ;REQUEST THE 'HOST' CPU TO SELECT A PORT FOR TESTING
1458
1459  4305                                      REQ     7
 (1)  4305  CD  06  28       18.0                CALL    REQST
 (1)  4308  00                                   .BYTE                   ;DATA PATTERN NUMBER
 (1)  4309  00  00                               .WORD                   ;SYSTEM '*' COUNT
 (1)  430B  00  00                               .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                                   .BYTE                   ;DATA COMPARE FLAG IF =1
 (1)  430E  07                                   .BYTE   7               ;REQUEST CODE
1460  430F                                RIN     R12L                    ;GET THE SELECTED PORT FROM 'HOST'
 (1)  430F  DB  94            10.0               IN      R12L                    ;READ R12L INTO AC
 (1)  4311  7F                 4.0               MOV     A,A                     ;RETRY LINK
1461  4312  32  5E  47         13.0       STA     UNITMP                  ;SAVE IT
1462  4315  A7                 4.0       ANA     A                       ;SET THE CONDITION BITS
1463  4316  C2  27  43         10.0       JNZ     FOUND                   ;SEE IF ANY WERE SELECTED IN 'HOST'
```

```
1465  4319                                        ERR     EXIT,NOUNIT
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0001                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  431D  00                                            .BYTE
 (1)  431E  CD  15  28      18.0                NOUNIT::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  4F  45      10.0                          JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1466                                        ;>OPERATOR ERROR - NO TU78 UNIT SELECTED
1467                                        ;>FATAL ERROR - TEST ABORTED
1468  4324  C3  4F  45      10.0                  JMP     EXIT
1469                                        ;HERE IF THE HOST CPU INDICATED A PORT TO USE
1470
1471  4327  CD  AA  46      18.0          FOUND:  CALL    CLEAR           ;INIT ALL THE PORTS IN THE PORT BOARD
1472  432A  3A  5E  47      13.0                  LDA     UNITMP          ;GET THE RESPONSE FROM 'HOST' CPU
1473  432D  06  00           7.0                  MVI     B,0             ;INIT 'B'
1474  432F  E6  01           7.0                  ANI     1               ;IS UNIT 0 SELECTED?
1475  4331  C2  47  43       10.0                  JNZ     FPORT           ;YUP - FOUND THE PORT TO USE
1476  4334  04               4.0                  INR     B               ;NO - COUNT THE PORT
1477  4335  3A  5E  47      13.0                  LDA     UNITMP          ;GET RESPONSE AGAIN
1478  4338  E6  02           7.0                  ANI     2               ;PORT 1 SELECTED?
1479  433A  C2  47  43       10.0                  JNZ     FPORT           ;YUP
1480  433D  04               4.0                  INR     B               ;COUNT THE PORT
1481  433E  3A  5E  47      13.0                  LDA     UNITMP          ;GET RESPONSE
1482  4341  E6  04           7.0                  ANI     4               ;PORT 2 SELECTED?
1483  4343  C2  47  43       10.0                  JNZ     FPORT           ;YUP
1484  4346  04               4.0                  INR     B               ;NO - ASSUME PORT 3
1485
1486  4347  78               4.0          FPORT:  MOV     A,B             ;PUT UNIT # IN AC
1487  4348  32  5F  47      13.0                  STA     SUNIT           ;SAVE THE UNIT UNDER TEST
1488  434B  DB  E0           10.0                  IN      INTSTA          ;GET THE PORT SELECT STATUS
1489  434D  E6  80           7.0                  ANI     BIT7
1490  434F  B0               4.0                  ORA     B               ;ADD IN THE SELECTED PORT #
1491  4350  D3  E0           10.0                  OUT     MBSEL           ;SELECT THE DESIRED PORT TO USE
1492  4352  3E  60           7.0                  MVI     A,P.LWR!P.LCS   ;GET THE LOOP-COMMAND-STATUS BIT
1493  4354  D3  48           10.0                  OUT     PDIAG           ;SELECT IT
1494  4356  AF               4.0                  XRA     A
1495  4357  D3  44           10.0                  OUT     TAMT            ;CLEAR THE AMTIE LINES
1496  4359  D3  D2           10.0                  OUT     TRKENA          ;CLEAR THE TRACK ENABLE BITS
1497  435B  D3  D2           10.0                  OUT     TRKENA
1498  435D  3E  04           7.0                  MVI     A,P.RPEN        ;SET READ PATH ENABLE
1499  435F  D3  4C           10.0                  OUT     PENAB           ;NOW
1500  4361  3E  10           7.0                  MVI     A,RDCLK         ;SETUP THE READ AND WRITE CLOCK TO 'NORMAL'
1501  4363  D3  F0           10.0                  OUT     CLKCTL
1502  4365  D3  C0           10.0                  OUT     DBUSCTL         ;CLEAR THE DATA BUS CONTROL WORD
1503  4367  21  A5  45      10.0                  LXI     H,XFRPTB        ;POINT TO THE 1ST DATA XFR PARAMETER TABLE
1504  436A  22  24  46      16.0                  SHLD    PTABLE          ;SAVE THE TABLE ADDRESS
1505                                        ;INIT THE COUNTER CONTROL AND COUNTERS
1506
1507  436D                                T1LOOP: INICNT
 (1)  436D  3E  30           7.0                          MVI     A,@060          ;INIT THE
 (1)  436F  D3  D7           10.0                         OUT     CNTCTL          ;BYTE COUNTER
 (1)  4371  3E  70           7.0                          MVI     A,@160          ;INIT THE
 (1)  4373  D3  D7           10.0                         OUT     CNTCTL          ;PAD COUNTER
```

```
 (1)  4375  3E  B0        7.0                    MVI   A,@260          ;INIT THE
 (1)  4377  D3  D7       10.0                    OUT   CNTCTL          ;ERROR COUNTER
1508  4379                          CLRECT
 (1)  4379  AF            4.0                    XRA   A               ;CLEAR THE ACCUMULATOR
 (1)  437A  D3  D6       10.0                    OUT   ERRCNT          ;CLEAR BITS 7-0
 (1)  437C  D3  D6       10.0                    OUT   ERRCNT          ;CLEAR BITS 15-8
1509
1510  437E  2A  24  46   16.0            LHLD   PTABLE                ;GET THE PARAMETER POINTER
1511  4381  7E            7.0            MOV    A,M                   ;GET THE LOW BYTE OF EXPECTED ADDRESS
1512  4382  32  5C  47   13.0            STA    EXPADR                ;SAVE IT
1513  4385  23            6.0            INX    H                     ;POINT TO THE HIGH BYTE OF ADDRESS
1514  4386  7E            7.0            MOV    A,M                   ;GET THE HIGH BYTE
1515  4387  32  5D  47   13.0            STA    EXPADR+1              ;SAVE IT
1516  438A  23            6.0            INX    H                     ;POINT TO THE PACKING MODE NUMBER
1517  438B  7E            7.0            MOV    A,M                   ;GET IT
1518  438C  32  5A  47   13.0            STA    FMTNUM                ;SAVE IT
1519  438F  23            6.0            INX    H                     ;POINT TO THE SKIP COUNT FOR THIS XFR
1520  4390  7E            7.0            MOV    A,M                   ;GET IT
1521  4391  32  5B  47   13.0            STA    SKIPCT                ;SAVE IT
1522  4394  23            6.0            INX    H                     ;POINT TO THE # SCLK'S TO CHECK IN XFR
1523  4395  7E            7.0            MOV    A,M                   ;GET THE COUNT
1524  4396  32  59  47   13.0            STA    ESCLK                 ;SAVE THE NUMBER
1525  4399  23            6.0            INX    H                     ;POINT TO THE 1ST WORD IN NEXT ENTRY
1526  439A  22  24  46   16.0            SHLD   PTABLE                ;SAVE THE ADDRESS OF NEW TABLE
1527
1528  439D  3A  5A  47   13.0            LDA    FMTNUM                ;GET THE PACKING MODE # AGAIN
1529  43A0  07            4.0            RLC
1530  43A1  07            4.0            RLC
1531  43A2  07            4.0            RLC
1532  43A3  07            4.0            RLC
1533  43A4  E6  70        7.0            ANI    $70                   ;POSITION THE FMT #
1534  43A6  47            4.0            MOV    B,A                   ;TEMP SAVE
1535  43A7  3A  5B  47   13.0            LDA    SKIPCT                ;GET THE SELECTED SKIP COUNT FOR XFR
1536  43AA  B0            4.0            ORA    B                     ;ADD IN THE FORMAT NUMBER
1537  43AB  D3  D0       10.0            OUT    DATACTL               ;LOAD THE DATA CONTROL WORD
1538  43AD                               ROUT   RO2H                  ;SAVE FOR ERROR PRINTING
 (1)  43AD  D3  85       10.0                   OUT    RO2H                  ;WRITE AC INTO RO2H
 (1)  43AF  7F            4.0                   MOV    A,A                   ;RETRY LINK
1539  43B0  3E  9B        7.0            MVI    A,@233                ;SET THE DDR CONTROL TO 'IN'
1540  43B2  D3  DB       10.0            OUT    DDRCTL                ;TO READ THE 'DDR'
1541  43B4  3E  00        7.0            MVI    A,@0                  ;GET THE RESIDUAL CHAR WORD
1542  43B6  D3  D1       10.0            OUT    RESCHR                ;LOAD IT
1543  43B8  3E  32        7.0            MVI    A,50                  ;SETUP THE BYTE COUNTER FOR 8 DATA GROUPS
1544  43BA  D3  D4       10.0            OUT    BYTCNT
1545  43BC  AF            4.0            XRA    A
1546  43BD  D3  D4       10.0            OUT    BYTCNT
1547  43BF  3E  06        7.0            MVI    A,6                   ;NEED 6 PAD CHARACTERS FOR THE XFR
1548  43C1  D3  D5       10.0            OUT    PADCNT                ;LOAD IT
1549  43C3  AF            4.0            XRA    A
1550  43C4  D3  D5       10.0            OUT    PADCNT
1551  43C6                               ROUT   RO2L                  ;CLEAR OUT THE REST OF REG 2 IN CAS
 (1)  43C6  D3  84       10.0                   OUT    RO2L                  ;WRITE AC INTO RO2L
 (1)  43C8  7F            4.0                   MOV    A,A                   ;RETRY LINK
1552  43C9                               ROUT   RO5L                  ;CLEAR THE BYTE/SCLK COUNTER FOR REPORTS
```

```
  (1)  43C?  D3  8A          10.0                  OUT     R05L            ;WRITE AC INTO R05L
  (1)  43CB  7F              4.0                   MOV     A,A             ;RETRY LINK
1553   43CC  32  58  47       13.0          STA     SCLKCT          ;INIT THE SCLK DETECTED COUNTER
1554
1555                                 ;RESTART THE WMC AND SET WMC TO SINGLE STEP MODE
1556
1557   43CF  3E  01            7.0           MVI     A,W.RST         ;GET THE WMC RESTART BIT
1558   43D1  D3  D3           10.0           OUT     WMCCTL          ;SET RESTART
1559   43D3  3E  15            7.0           MVI     A,SSTEP!RDCLK   ;GET SINGLE STEP FOR WMC
1560   43D5  D3  F0           10.0           OUT     CLKCTL          ;SET SINGLE STEP MODE
1561   43D7  32  60  47       13.0           STA     CCTLWD          ;SAVE THE WORD
1562   43DA  CD  B0  47       18.0           CALL    CLKSYS          ;CLOCK THE SYSTEM TO CAUSE RESTART TO
1563   43DD  CD  B0  47       18.0           CALL    CLKSYS          ;TO TAKE HOLD
1564   43E0  AF                4.0           XRA     A
1565   43E1  D3  D3           10.0           OUT     WMCCTL          ;FINISH THE RESTART CYCLE
1566
1567                                 ;CLOCK THE WMC THRU ITS RESTART AREA
1568
1569   43E3  06  B4            7.0           MVI     B,180           ;GET A LOOP COUNT
1570   43E5  CD  B0  47       18.0  CLKRS:  CALL    CLKSYS          ;CLOCK THE SYSTEM
1571   43E8  05                4.0           DCR     B               ;DECREMENT THE LOOP COUNTER
1572   43E9  C2  E5  43       10.0           JNZ     CLKRS           ;CLOCK TILL COUNT RUN OUT
1573
1574   43EC  3E  90            7.0           MVI     A,W.GCR!W.ENAB  ;SETUP FOR A GCR DATA XFR
1575   43EE  D3  D3           10.0  T1ST:    OUT     WMCCTL          ;LOAD THE WORD IN WMC
1576
1577                                 ;CALCULATE THE 1ST DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
1578
1579   43F0  21  95  45       10.0  C1EC1:   LXI     H,DG1           ;POINT THE THE 1ST DATA GROUP DATA TABLE
1580   43F3  06  07            7.0           MVI     B,7             ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
1581   43F5  CD  71  47       18.0           CALL    CLECC           ;INIT THE ECC CHARACTER
1582
1583   43F8  7E                7.0  C1EC1L:  MOV     A,M             ;GET A DATA BYTE FROM THE TABLE
1584   43F9  CD  76  47       18.0           CALL    ECC             ;CALC ECC WITH THIS CHARACTER
1585   43FC  23                6.0           INX     H               ;POINT TO THE NEXT DATA BYTE IN TABLE
1586   43FD  05                4.0           DCR     B               ;DECREMENT THE COUNTER OF DATA BYTES
1587   43FE  C2  F8  43       10.0           JNZ     C1EC1L          ;STAY HERE TILL ALL DATA BYTES USED
1588   4401  3A  9D  47       13.0           LDA     ECCCHR          ;GET THE CALCULATED ECC CHARACTER
1589   4404  32  9C  45       13.0           STA     DG1ECC          ;SAVE THE CALCULATED ECC CHARACTER
1590
1591                                 ;CALCULATE THE 2ND DATA GROUP ECC CHARACTER FROM THE 'SUPPLY' DATA
1592
1593   4407  21  9D  45       10.0  C1EC2:   LXI     H,DG2           ;POINT THE THE 2ND DATA GROUP DATA TABLE
1594   440A  06  07            7.0           MVI     B,7             ;SETUP THE LOOP COUNTER FOR 7 DATA BYTES
1595   440C  CD  71  47       18.0           CALL    CLECC           ;INIT THE ECC CHARACTER
1596
1597   440F  7E                7.0  C1EC2L:  MOV     A,M             ;GET A DATA BYTE FROM THE TABLE
1598   4410  CD  76  47       18.0           CALL    ECC             ;CALC ECC WITH THIS CHARACTER
1599   4413  23                6.0           INX     H               ;POINT TO THE NEXT DATA BYTE IN TABLE
1600   4414  05                4.0           DCR     B               ;DECREMENT THE COUNTER OF DATA BYTES
1601   4415  C2  0F  44       10.0           JNZ     C1EC2L          ;STAY HERE TILL ALL DATA BYTES USED
1602   4418  3A  9D  47       13.0           LDA     ECCCHR          ;GET THE CALCULATED ECC CHARACTER
1603   441B  32  A4  45       13.0           STA     DG2ECC          ;SAVE THE CALCULATED ECC CHARACTER
1604   441E  3E  A8            7.0  T1PLO:   MVI     A,R.PLO1!R.TBJN!R.PLOD  ;SET PLO BYPASS MODE 2 ENABLE REA. PATH CLOCK
```

```
1605                                                                    ;SET TIE BUS JAM AND PLO DISABLE
1606  4420  D3  09          10.0          OUT   RPCTL
1607  4422  AF               4.0          XRA   A
1608  4423  D3  0A          10.0          OUT   RTIEB              ;CLEAR THE TIE BUS
1609  4425  D3  08          10.0          OUT   RFIFOL             ;CLOCK THE FIFO'S
1610  4427  3E  0D           7.0          MVI   A,RCLRT           ;ISSUE CLEAR ALL COMMAND
1611  4429  D3  0B          10.0          OUT   RCMD
1612  442B  00               4.0          NOP                     ;WAIT
1613  442C  00               4.0          NOP                     ;WAIT
1614  442D  00               4.0          NOP
1615  442E  00               4.0          NOP
1616  442F  00               4.0          NOP
1617  4430  3E  A9           7.0          MVI   A,R.PLO1.R.STPC!R.PLOD!R.TBJN   ;STOP THE READ PATH
1618                                                                  ;SET TIE BUS JAM AND PLO DISABLE
1619  4432  D3  09          10.0          OUT   RPCTL
1620
1621  4434  0E  03           7.0          MVI   C,@3              ;INIT THE LOOP COUNT FOR 6 DATA GROUPS
1622
1623  4436  21  95  45      10.0  T1SG1:  LXI   H,DG1             ;POINT TO THE 1ST DATA GROUP FOR TRANSLATION
1624  4439  CD  69  45      18.0          CALL  CONVRT            ;CONVERT SUBGROUP - 4X5
1625  443C  CD  7E  45      18.0          CALL  LOADIT            ;INJECT THE DATA INTO THE FIFO
1626
1627  443F  21  99  45      10.0  T1SG2:  LXI   H,SG2             ;POINT TO THE 2ND SUBGROUP OF DATA
1628  4442  CD  69  45      18.0          CALL  CONVRT            ;MAKE IT 4X5
1629  4445  CD  7E  45      18.0          CALL  LOADIT            ;INJECT IT INTO FIFO
1630
1631  4448  21  9D  45      10.0  T1SG3:  LXI   H,DG2             ;POINT TO THE 2ND DATA GROUP - SUBGROUP 3
1632  444B  CD  69  45      18.0          CALL  CONVRT            ;MAKE IT 4X5
1633  444E  CD  7E  45      18.0          CALL  LOADIT            ;INJECT THE DATA INTO FIFO
1634
1635  4451  21  A1  45      10.0  T1SG4:  LXI   H,SG4             ;POINT TO THE 4TH SUBGROUP OF DATA
1636  4454  CD  69  45      18.0          CALL  CONVRT            ;MAKE IT 4X5
1637  4457  CD  7E  45      18.0          CALL  LOADIT            ;INJECT THE DATA INTO FIFO
1638
1639  445A  0D               4.0          DCR   C                 ;DECREMENT THE LOOP COUNTER
1640  445B  C2  36  44      10.0          JNZ   T1SG1             ;KEEP INJECTING DATA TILL 6 GROUPS IN FIFO
1641
1642  445E  3E  0B           7.0          MVI   A,DIARD           ;LOAD THE DIAGNOSTIC READ
1643  4460  D3  0B          10.0          OUT   RCMD              ;COMMAND
1644  4462  11  01  00      10.0  T1CLKA: LXI   D,1               ;SET WATCH DOG INCREMENT
1645  4465  21  68  C5      10.0          LXI   H,-15000          ;SET WATCH DOG COUNT TO 15000
1646  4468  D3  0C          10.0  T1CLK:  OUT   RINST             ;STEP THE READ PATH
1647  446A  CD  B0  47      18.0          CALL  CLKSYS            ;CLOCK THE WMC
1648  446D  DB  C0          10.0          IN    DBUSSTA           ;GET THE DATABUS STATUS
1649  446F  E6  02           7.0          ANI   T.SCLK            ;SEE IF SCLK UP - DATA READY
1650  4471  C2  86  44      10.0          JNZ   T1SCLK            ;JUMP IF SCLK UP
1651  4474  19              10.0          DAD   D
1652  4475  D2  68  44      10.0          JNC   T1CLK             ;STAY IN LOOP UNLESS TIMEOUT
1653
```

```
1655  4478                                        ERR     T1LOOP,DUMM1,3
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4478  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0002                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  447B  02                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  447C  03                                            .BYTE   3
 (1)  447D  CD  15  28      18.0                DUMM1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4480  DA  6D  43      10.0                        JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1656                                        ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO SET IN DBUSSTA
1657                                        ;>AFTER ISSUEING A READ FORWARD COMMAND TO THE TM78
1658                                        ;<FATAL ERROR - TEST ABORTED
1659  4483  C3  4F  45      10.0                JMP     EXIT
1660
1661                                        ;HERE TO CLOCK THE WMC SO THE 'SCLK' SIGNAL WILL GO =0
1662                                        ;THIS IS NECESSARY BEFORE STROBING THE DDR REGISTERS...THEY ARE ONLY
1663                                        ;VALID AFTER 'SCLK' HAS TRANSITIONED TO ZERO INDICATING THE DATA IS
1664                                        ;ON THE MASSBUS
1665
1666  4486  06  FF           7.0  T1SCLK: MVI     B,$FF           ;SETUP A DROP DEAD COUNT
1667  4488  D3  0C          10.0  T1SCKL: OUT     RINST           ;CLOCK THE READ PATH ONCE
1668  448A  CD  B0  47      18.0          CALL    CLKSYS          ;CLOCK THE WMC TO CLEAR 'SCLK'
1669  448D  DB  C0          10.0          IN      DBUSSTA         ;SEE IF 'SCLK' IS GONE
1670  448F  E6  02           7.0          ANI     T.SCLK          ;SAVE ONLY THE 'SCLK' BIT IN STATUS WORD
1671  4491  CA  A6  44      10.0          JZ      T1SCK0          ;JUMP IF 'SCLK' = 0
1672  4494  05                4.0          DCR     B               ;DECREMENT THE COUNTER
1673  4495  C2  88  44      10.0          JNZ     T1SCKL          ;LOOP BACK TILL COUNTER =0
1674  4498                                ERR     T1LOOP,DUMM2,3
 (1)                                              ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4498  CD  09  28      18.0                          CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)       0003                                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  449B  03                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  449C  03                                            .BYTE   3
 (1)  449D  CD  15  28      18.0                DUMM2:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44A0  DA  6D  43      10.0                        JC      T1LOOP          ;LOOP ADDRESS IF LOOP SPECIFIED
1675                                        ;>TIMEOUT WHILE WAITING FOR 'SCLK' TO CLEAR AFTER SETTING
1676                                        ;>DURING A READ FORWARD COMMAND
1677                                        ;<FATAL ERROR - TEST ABORTED
1678  44A3  C3  4F  45      10.0                JMP     EXIT
1679                                        ;HERE WHEN 'SCLK' WAS SET - 18 BITS OF DATA HAS BEEN ASSEMBLED FROM
1680                                        ;THE BYTE ASSEMBLY LOGIC. DATA CAME FROM THE READ PATH (FIFO).
1681
1682  44A6  DB  D8          10.0  T1SCK0: IN      DDRA            ;GET THE DDR A DATA
1683  44A8  32  55  47      13.0          STA     SDDRA           ;SAVE IT
1684  44AB                                ROUT    R16L            ;SAVE FOR PRINTING
 (1)  44AB  D3  9C          10.0                        OUT     R16L            ;WRITE AC INTO R16L
 (1)  44AD  7F                4.0                        MOV     A,A             ;RETRY LINK
1685  44AE  DB  D9          10.0          IN      DDRB            ;GET DDR B DATA
1686  44B0  32  56  47      13.0          STA     SDDRB           ;SAVE IT
1687  44B3                                ROUT    R16H            ;SAVE FOR ERROR REPORTING
 (1)  44B3  D3  9D          10.0                        OUT     R16H            ;WRITE AC INTO R16H
 (1)  44B5  7F                4.0                        MOV     A,A             ;RETRY LINK
1688  44B6  DB  DA          10.0          IN      DDRC            ;GET DDR C DATA
1689  44B8  E6  03           7.0          ANI     3
1690  44BA  32  57  47      13.0          STA     SDDRC           ;SAVE IT
```

```
1691  44BD  F6  80        7.0          ORI   BIT7          ;ADD IN THE PRINT FLAG
1692  44BF                             ROUT  R17L          ;SAVE IT FOR PRINTING
 (1)  44BF  D3  9E       10.0                OUT   R17L           ;WRITE AC INTO R17L
 (1)  44C1  7F            4.0                MOV   A,A            ;RETRY LINK
1693
1694  44C2  3A  58  47   13.0          LDA   SCLKCT        ;GET THE CURRENT SCLK COUNT
1695  44C5  3C            4.0          INR   A             ;COUNT THIS SCLK
1696  44C6  32  58  47   13.0          STA   SCLKCT        ;SAVE THE NEW COUNT
1697  44C9                             ROUT  R05L          ;ALSO IN CAS FOR ERROR REPORTING
 (1)  44C9  D3  8A       10.0                OUT   R05L           ;WRITE AC INTO R05L
 (1)  44CB  7F            4.0                MOV   A,A            ;RETRY LINK
1698  44CC  AF            4.0          XRA   A
1699  44CD                             ROUT  R05H
 (1)  44CD  D3  8B       10.0                OUT   R05H           ;WRITE AC INTO R05H
 (1)  44CF  7F            4.0                MOV   A,A            ;RETRY LINK
1700
1701                                 ;CHECK THE DATA IN DDR A - BITS 7-0 OF THE 18 BITS ASSEMBLED
1702
1703  44D0  2A  5C  47   16.0  T1CKA: LHLD   EXPADR         ;POINT TO THE TABLE OF EXPECTED DATA
1704  44D3  7E            7.0          MOV   A,M           ;GET BITS 7-0
1705  44D4                             ROUT  R14L          ;SAVE FOR ERROR REPORTING
 (1)  44D4  D3  98       10.0                OUT   R14L           ;WRITE AC INTO R14L
 (1)  44D6  7F            4.0                MOV   A,A            ;RETRY LINK
1706  44D7  23            6.0          INX   H             ;POINT TO NEXT BYTE
1707  44D8  7E            7.0          MOV   A,M           ;GET BITS 15-8
1708  44D9                             ROUT  R14H          ;SAVE IT
 (1)  44D9  D3  99       10.0                OUT   R14H           ;WRITE AC INTO R14H
 (1)  44DB  7F            4.0                MOV   A,A            ;RETRY LINK
1709  44DC  23            6.0          INX   H
1710  44DD  7E            7.0          MOV   A,M           ;GET BITS 17-16
1711  44DE  F6  80        7.0          ORI   BIT7          ;ADD IN THE PRINT FLAG
1712  44E0                             ROUT  R15L          ;SAVE FOR ERROR REPORT
 (1)  44E0  D3  9A       10.0                OUT   R15L           ;WRITE AC INTO R15L
 (1)  44E2  7F            4.0                MOV   A,A            ;RETRY LINK
1713  44E3  3A  55  47   13.0          LDA   SDDRA         ;GET THE DDR A DATA
1714  44E6                             ROUT  ADATA         ;SAVE IN CASE OF ERROR DETECTED
 (1)  44E6  D3  94       10.0                OUT   ADATA          ;WRITE AC INTO ADATA
 (1)  44E8  7F            4.0                MOV   A,A            ;RETRY LINK
1715  44E9  2A  5C  47   16.0          LHLD  EXPADR        ;POINT TO THE EXPECTED DATA TABLE
1716  44EC  BE            7.0          CMP   M             ;ACTUAL DATA = EXPECTED?
1717  44ED  CA  F9  44   10.0          JZ    T1CKAC        ;JUMP IF OK
1718  44F0  7E            7.0          MOV   A,M           ;GET THE EXPECTED DATA
1719  44F1                             ROUT  EDATA         ;SAVE FOR ERROR REPORT
 (1)  44F1  D3  95       10.0                OUT   EDATA          ;WRITE AC INTO EDATA
 (1)  44F3  7F            4.0                MOV   A,A            ;RETRY LINK
1720  44F4                             ERRB  T1LOOP,T1CKAC,@10
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44F4  CD  12  28   18.0                CALL  ERLPB          ;PROCESS ERROR - DO 2.3
 (1)        0004                             MSGN  =       MSGN+1       ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44F7  04                               .BYTE MSGN           ;MESSAGE NUMBER ID
 (1)  44F8  08                               .BYTE @10            ;PRINT ROUTINE NUMBER
 (1)  44F9  CD  15  28   18.0          T1CKAC::      CALL  CKLOP          ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44FC  DA  6D  43   10.0                JC    T1LOOP         ;LOOP ADDRESS IF LOOP SPECIFIED
1721                             ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 7-0
```

```
1722  44FF  23                  6.0  T1CKB:  INX   H                ;POINT TO THE EXPECTED DATA FOR DDR B
1723  4500  3A  56  47           13.0          LDA   SDDRB            ;GET THE DDR B DATA
1724  4503                             ROUT  ADATA                  ;SAVE FOR ERROR REPORT
 (1)  4503  D3  94              10.0          OUT   ADATA                    ;WRITE AC INTO ADATA
 (1)  4505  7F                   4.0          MOV   A,A                      ;RETRY LINK
1725  4506  BE                   7.0          CMP   M                  ;ACTUAL = EXPECTED DATA?
1726  4507  CA  13  45           10.0          JZ    T1CKBC             ;JUMP IF OK
1727  450A  7E                   7.0          MOV   A,M                ;GET THE EXPECTED DATA
1728  450B                             ROUT  EDATA                  ;SAVE FOR ERROR REPORT
 (1)  450B  D3  95              10.0          OUT   EDATA                    ;WRITE AC INTO EDATA
 (1)  450D  7F                   4.0          MOV   A,A                      ;RETRY LINK
1729  450E                             ERRB  T1LOOP,T1CKBC,@10
 (1)                             ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  450E  CD  12  28           18.0                CALL  ERLPB              ;PROCESS ERROR - DO 2.3
 (1)        0005                                     MSGN  =     MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4511  05                                       .BYTE MSGN               ;MESSAGE NUMBER ID
 (1)  4512  08                                       .BYTE @10                ;PRINT ROUTINE NUMBER
 (1)  4513  CD  15  28           18.0          T1CKBC::      CALL  CKLOP              ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4516  DA  6D  43           10.0                JC    T1LOOP             ;LOOP ADDRESS IF LOOP SPECIFIED
1730                        ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 15-8
1731
1732  4519  23                  6.0  T1CKC:  INX   H                ;POINT TO EXPECTED DDR C DATA
1733  451A  3A  57  47           13.0          LDA   SDDRC            ;GET DDR C DATA
1734  451D                             ROUT  ADATA                  ;SAVE FOR ERROR REPORT
 (1)  451D  D3  94              10.0          OUT   ADATA                    ;WRITE AC INTO ADATA
 (1)  451F  7F                   4.0          MOV   A,A                      ;RETRY LINK
1735  4520  BE                   7.0          CMP   M                  ;ACTUAL DATA = EXPECTED DDRC DATA?
1736  4521  CA  2D  45           10.0          JZ    T1CKCC             ;JUMP IF OK
1737  4524  7E                   7.0          MOV   A,M                ;GET THE EXPECTED DATA
1738  4525                             ROUT  EDATA                  ;SAVE FOR ERROR REPORT
 (1)  4525  D3  95              10.0          OUT   EDATA                    ;WRITE AC INTO EDATA
 (1)  4527  7F                   4.0          MOV   A,A                      ;RETRY LINK
1739  4528                             ERRB  T1LOOP,T1CKCC,@10
 (1)                             ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4528  CD  12  28           18.0                CALL  ERLPB              ;PROCESS ERROR - DO 2.3
 (1)        0006                                     MSGN  =     MSGN+1              ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  452B  06                                       .BYTE MSGN               ;MESSAGE NUMBER ID
 (1)  452C  08                                       .BYTE @10                ;PRINT ROUTINE NUMBER
 (1)  452D  CD  15  28           18.0          T1CKCC::      CALL  CKLOP              ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4530  DA  6D  43           10.0                JC    T1LOOP             ;LOOP ADDRESS IF LOOP SPECIFIED
1740                        ;>ASSEMBLED 18 BITS OF DATA FROM READ PATH WAS INCORRECT FOR BITS 17-16
1741
1742  4533  2A  5C  47           16.0          LHLD  EXPADR             ;GET THE CURRENT EXPECTED DATA ADDR.
1743  4536  23                   6.0          INX   H
1744  4537  23                   6.0          INX   H                  ;UPDATE THE EXPECTED ADDRESS POINTER
1745  4538  23                   6.0          INX   H                  ;TO THE DATA
1746  4539  22  5C  47           16.0          SHLD  EXPADR             ;SAVE THE 'NEW' EXPECTED DATA TABLE ADDR.
1747  453C  3A  58  47           13.0          LDA   SCLKCT             ;GET THE SCLK COUNT
1748  453F  21  59  47           10.0          LXI   H,ESCLK            ;POINT TO THE EXPECTED SCLK COUNT
1749  4542  BE                   7.0          CMP   M                  ;DONE WITH THIS TEST PASS?
1750  4543  C2  62  44           10.0          JNZ   T1CLKA             ;JUMP IF STILL EXPECTING MORE SCLKS
1751
1752                        ;HERE TO SEE IF ALL XFRS IN THE 'PTABLE' HAVE BEEN DONE
1753
```

```
1754  4546  2A  24  46   16.0          LHLD    PTABLE          ;POINT TO THE TABLE
1755  4549  7E               7.0          MOV     A,M             ;GET THE 1ST WORD OF AN ENTRY
1756  454A  FE  FF            7.0          CPI     @377            ;END OF THE TABLE?
1757  454C  C2  6D  43   10.0          JNZ     T1LOOP          ;NO - DO THIS ENTRY
1758
1759  454F                         EXIT:   ENDTST  FOUND
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  454F                                REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  454F  CD  06  28   18.0                                 CALL    REQST
 (2)  4552  00                                                .BYTE                   ;DATA PATTERN NUMBER
 (2)  4553  00  00                                            .WORD                   ;SYSTEM '"' COUNT
 (2)  4555  00  00                                            .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4557  00                                                .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4558  07                                                .BYTE   7               ;REQUEST CODE
 (1)  4559  3A  9A  4F   13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  455C  3D               4.0          DCR     A               ;DOWNCOUNT
 (1)  455D  32  9A  4F   13.0          STA     ITERA           ;SAVE COUNT
 (1)  4560  F2  27  43   10.0          JP      FOUND           ;DO TEST UNTIL TILL = 0
1760  4563  CD  AA  46   18.0          CALL    CLEAR           ;CLEAR ALL TU PORTS
1761  4566  C3  18  28   10.0          JMP     TSTEND
1762                               ;HERE IS THE SUBROUTINES NECESSARY FOR CONVERTING SUBGROUPS AND LOADING
1763                               ;THEM INTO THE FIFO
1764
1765  4569                         S
 (1)                                ; ***************************************************************************
1766                                ;*CONVRT -- THIS SUBROUTINE WILL GET A SUBGROUP OF DATA AS POINTED TO BY
1767                                ;*      THE H & L REGISTERS AND CAUSE THEM TO BE TRANSLATED (4X5). THE
1768                                ;*      TRANSLATED DATA WILL BE IN A TABLE CALLED 'TRNOUT'.
1769  4569                         S
 (1)                                ; ***************************************************************************
1770
1771  4569  D5                12.0  CONVRT: PUSH    D               ;SAVE D & E
1772  456A  11  61  47        10.0          LXI     D,TRNIN         ;POINT TO THE TEMP STORAGE FOR DATA BYTES
1773  456D  06  04            7.0          MVI     B,4             ;SETUP A LOOP COUNTER FOR 4 BYTES
1774
1775  456F  7E                 7.0  CNV1:   MOV     A,M             ;GET A BYTE OF DATA
1776  4570  EB                 4.0          XCHG                    ;POINT TO THE STORAGE
1777  4571  77                 7.0          MOV     M,A             ;SAVE DATA BYTE IN TEMP STORAGE
1778  4572  EB                 4.0          XCHG                    ;RESET POINTER
1779  4573  23                 6.0          INX     H               ;POINT TO NEXT DATA BYTE
1780  4574  13                 6.0          INX     D               ;POINT TO NEXT TEMP STORAGE ADDRESS
1781  4575  05                 4.0          DCR     B               ;UPDATE THE LOOP COUNT
1782  4576  C2  6F  45        10.0          JNZ     CNV1            ;LOOP TILL 4 BYTES IN TEMP STORAGE
1783  4579  CD  D5  46        18.0          CALL    T4X5            ;CONVERT THE 'SUBGROUP' OF DATA
1784  457C  D1                10.0          POP     D
1785  457D  C9                10.0          RET
1786  457E                         S
 (1)                                ; ***************************************************************************
1787                                ;*LOADIT -- THIS SUBROUTINE WILL TAKE THE TRANSLATED SUBGROUP (4X5)
1788                                ;*      DATA FROM TABLE 'TRNOUT' AND LOAD THE DATA INTO THE READ
1789                                ;*      PATH FIFO.
1790  457E                         S
 (1)                                ; ***************************************************************************
1791
```

```
1792  457E  21  65  47      10.0  LOADIT: LXI   H,TRNOUT        ;POINT TO THE TRANSLATED DATA TABLE
1793  4581  06  05              7.0          MVI   B,5            ;GET THE LOOP COUNT FOR SUBGROUP
1794
1795  4583  7E                  7.0  LDFIFO: MOV   A,M            ;GET A BYTE OF DATA
1796  4584  D3  40             10.0          OUT   TCMD           ;STORE IT IN THE CMD REG FOR LOADING
1797  4586  23                  6.0          INX   H              ;POINT TO THE PARITY BIT
1798  4587  7E                  7.0          MOV   A,M            ;GET THE PARITY BIT
1799  4588  07                  4.0          RLC
1800  4589  F6  60              7.0          ORI   P.LWR!P.LCS    ;ADD IN THE OTHER NECESSARY BITS
1801  458B  D3  48             10.0          OUT   PDIAG          ;LOAD INTO CONTROL REG FOR LOADING
1802  458D  D3  08             10.0          OUT   RFIFOL         ;STROBE THE DATA INTO FIFO
1803  458F  23                  6.0          INX   H              ;POINT TO THE NEXT SET OF DATA
1804  4590  05                  4.0          DCR   B              ;DECREMENT THE LOOP COUNTER
1805  4591  C2  83  45         10.0          JNZ   LDFIFO         ;LOOP TILL 5 SETS OF DATA LOADED INTO FIFO
1806  4594  C9                 10.0          RET
1807                                  ;READ PATH BYTE ASSEMBLY DATA BYTES TO BE INJECTED INTO THE READ PATH
1808                                  ;FIFO - 4 SUBGROUPS WORTH OF DATA - ALL DATA TRANSFERS ARE 'READ FORWARD'
1809
1810  4595  01                      DG1:    .BYTE  @1       ;1ST BYTE IN THE 1ST SUBGROUP
1811  4596  02                              .BYTE  @2       ;1ST SUBGROUP HAS 4 DATA BYTES IN IT
1812  4597  03                              .BYTE  @3
1813  4598  04                              .BYTE  @4       ;LAST BYTE IN THE 1ST SUBGROUP
1814
1815  4599  05                      SG2:    .BYTE  @5       ;1ST BYTE IN 2ND SUBGROUP
1816  459A  06                              .BYTE  @6       ;2ND SUBGROUP HAS 3 BYTES IN IT TO FINISH THE DATA
1817  459B  07                              .BYTE  @7       ;DATA GROUP OF 7 DATA BYTES
1818  459C  00                      DG1ECC: .BYTE  0        ;STORAGE FOR THE 1ST ECC BYTE (CALCULATED)
1819
1820  459D  08                      DG2:    .BYTE  @10      ;1ST BYTE OF THE 3RD SUBGROUP
1821  459E  09                              .BYTE  @11      ;3RD SUBGROUP HAS 4 DATA BYTES
1822  459F  0A                              .BYTE  @12
1823  45A0  0B                              .BYTE  @13      ;LAST DATA BYTE IN 3RD SUBGROUP
1824
1825  45A1  0C                      SG4:    .BYTE  @14      ;1ST BYTE IN THE 4TH SUBGROUP
1826  45A2  0D                              .BYTE  @15      ;THIS SUBGROUP HAS ONLY 3 DATA BYTES
1827  45A3  0E                              .BYTE  @16      ;LAST BYTE IN 4TH SUBGROUP
1828  45A4  00                      DG2ECC: .BYTE  0        ;2ND DATA GROUP ECC BYTE STORAGE (CALCULATED)
1829                                  ;HERE IS THE PARAMETER TABLE FOR ALL DATA XFRS TO TEST THE SKIP COUNTS
1830                                  ;IN ALL DATA PACKING MODES
1831
1832  45A5                          XFRPTB: XFRP   F11N,0,1,2
 (1)  45A5  26  46                              .ADDR  F11N     ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45A7  00                                  .BYTE  0        ;PACKING MODE 0
 (1)  45A8  01                                  .BYTE  1        ;SKIP COUNT = 1 FOR THIS XFR
 (1)  45A9  02                                  .BYTE  2        ;2 EXPECTED THIS XFR
1833  45AA                                  XFRP   F15N,1,1,2
 (1)  45AA  2C  46                              .ADDR  F15N     ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45AC  01                                  .BYTE  1        ;PACKING MODE 1
 (1)  45AD  01                                  .BYTE  1        ;SKIP COUNT = 1 FOR THIS XFR
 (1)  45AE  02                                  .BYTE  2        ;2 EXPECTED THIS XFR
1834  45AF                                  XFRP   F10C1,2,1,2
 (1)  45AF  32  46                              .ADDR  F10C1    ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45B1  02                                  .BYTE  2        ;PACKING MODE 2
 (1)  45B2  01                                  .BYTE  1        ;SKIP COUNT = 1 FOR THIS XFR
```

```
 (1)  45B3  02                                  .BYTE   2       ;2 EXPECTED THIS XFR
1835  45B4                            XFRP    F10C2,2,2,2
 (1)  45B4  38  46                              .ADDR   F10C2   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45B6  02                                  .BYTE   2       ;PACKING MODE 2
 (1)  45B7  02                                  .BYTE   2       ;SKIP COUNT = 2 FOR THIS XFR
 (1)  45B8  02                                  .BYTE   2       ;2 EXPECTED THIS XFR
1836  45B9                            XFRP    F10C3,2,3,2
 (1)  45B9  3E  46                              .ADDR   F10C3   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45BB  02                                  .BYTE   2       ;PACKING MODE 2
 (1)  45BC  03                                  .BYTE   3       ;SKIP COUNT = 3 FOR THIS XFR
 (1)  45BD  02                                  .BYTE   2       ;2 EXPECTED THIS XFR
1837  45BE                            XFRP    F10D1,3,1,2
 (1)  45BE  44  46                              .ADDR   F10D1   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45C0  03                                  .BYTE   3       ;PACKING MODE 3
 (1)  45C1  01                                  .BYTE   1       ;SKIP COUNT = 1 FOR THIS XFR
 (1)  45C2  02                                  .BYTE   2       ;2 EXPECTED THIS XFR
1838  45C3                            XFRP    F10D2,3,2,2
 (1)  45C3  4A  46                              .ADDR   F10D2   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45C5  03                                  .BYTE   3       ;PACKING MODE 3
 (1)  45C6  02                                  .BYTE   2       ;SKIP COUNT = 2 FOR THIS XFR
 (1)  45C7  02                                  .BYTE   2       ;2 EXPECTED THIS XFR
1839  45C8                            XFRP    F10D3,3,3,2
 (1)  45C8  50  46                              .ADDR   F10D3   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45CA  03                                  .BYTE   3       ;PACKING MODE 3
 (1)  45CB  03                                  .BYTE   3       ;SKIP COUNT = 3 FOR THIS XFR
 (1)  45CC  02                                  .BYTE   2       ;2 EXPECTED THIS XFR
1840  45CD                            XFRP    F10D4,3,4,2
 (1)  45CD  56  46                              .ADDR   F10D4   ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45CF  03                                  .BYTE   3       ;PACKING MODE 3
 (1)  45D0  04                                  .BYTE   4       ;SKIP COUNT = 4 FOR THIS XFR
 (1)  45D1  02                                  .BYTE   2       ;2 EXPECTED THIS XFR
1841  45D2                            XFRP    F10HC1,4,1,4
 (1)  45D2  5C  46                              .ADDR   F10HC1  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45D4  04                                  .BYTE   4       ;PACKING MODE 4
 (1)  45D5  01                                  .BYTE   1       ;SKIP COUNT = 1 FOR THIS XFR
 (1)  45D6  04                                  .BYTE   4       ;4 EXPECTED THIS XFR
1842  45D7                            XFRP    F10HC2,4,2,4
 (1)  45D7  68  46                              .ADDR   F10HC2  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45D9  04                                  .BYTE   4       ;PACKING MODE 4
 (1)  45DA  02                                  .BYTE   2       ;SKIP COUNT = 2 FOR THIS XFR
 (1)  45DB  04                                  .BYTE   4       ;4 EXPECTED THIS XFR
1843  45DC                            XFRP    F10HC3,4,3,4
 (1)  45DC  74  46                              .ADDR   F10HC3  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45DE  04                                  .BYTE   4       ;PACKING MODE 4
 (1)  45DF  03                                  .BYTE   3       ;SKIP COUNT = 3 FOR THIS XFR
 (1)  45E0  04                                  .BYTE   4       ;4 EXPECTED THIS XFR
1844  45E1                            XFRP    F10HC4,4,4,4
 (1)  45E1  80  46                              .ADDR   F10HC4  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45E3  04                                  .BYTE   4       ;PACKING MODE 4
 (1)  45E4  04                                  .BYTE   4       ;SKIP COUNT = 4 FOR THIS XFR
 (1)  45E5  04                                  .BYTE   4       ;4 EXPECTED THIS XFR
1845  45E6                            XFRP    F10HC5,4,5,2
 (1)  45E6  92  46                              .ADDR   F10HC5  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45E8  04                                  .BYTE   4       ;PACKING MODE 4
```

```
 (1)  45E9  05                          .BYTE   5       ;SKIP COUNT = 5 FOR THIS XFR
 (1)  45EA  02                          .BYTE   2       ;2 EXPECTED THIS XFR
1846  45EB                    XFRP      F10HC6,4,6,2
 (1)  45EB  98  46                      .ADDR   F10HC6  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45ED  04                          .BYTE   4       ;PACKING MODE 4
 (1)  45EE  06                          .BYTE   6       ;SKIP COUNT = 6 FOR THIS XFR
 (1)  45EF  02                          .BYTE   2       ;2 EXPECTED THIS XFR
1847  45F0                    XFRP      F10HC7,4,7,2
 (1)  45F0  9E  46                      .ADDR   F10HC7  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45F2  04                          .BYTE   4       ;PACKING MODE 4
 (1)  45F3  07                          .BYTE   7       ;SKIP COUNT = 7 FOR THIS XFR
 (1)  45F4  02                          .BYTE   2       ;2 EXPECTED THIS XFR
1848  45F5                    XFRP      F10HC8,4,8,2
 (1)  45F5  A4  46                      .ADDR   F10HC8  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45F7  04                          .BYTE   4       ;PACKING MODE 4
 (1)  45F8  08                          .BYTE   8       ;SKIP COUNT = 8 FOR THIS XFR
 (1)  45F9  02                          .BYTE   2       ;2 EXPECTED THIS XFR
1849  45FA                    XFRP      F10HD1,6,1,4
 (1)  45FA  5C  46                      .ADDR   F10HD1  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  45FC  06                          .BYTE   6       ;PACKING MODE 6
 (1)  45FD  01                          .BYTE   1       ;SKIP COUNT = 1 FOR THIS XFR
 (1)  45FE  04                          .BYTE   4       ;4 EXPECTED THIS XFR
1850  45FF                    XFRP      F10HD2,6,2,4
 (1)  45FF  68  46                      .ADDR   F10HD2  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4601  06                          .BYTE   6       ;PACKING MODE 6
 (1)  4602  02                          .BYTE   2       ;SKIP COUNT = 2 FOR THIS XFR
 (1)  4603  04                          .BYTE   4       ;4 EXPECTED THIS XFR
1851  4604                    XFRP      F10HD3,6,3,4
 (1)  4604  74  46                      .ADDR   F10HD3  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4606  06                          .BYTE   6       ;PACKING MODE 6
 (1)  4607  03                          .BYTE   3       ;SKIP COUNT = 3 FOR THIS XFR
 (1)  4608  04                          .BYTE   4       ;4 EXPECTED THIS XFR
1852  4609                    XFRP      F10HD4,6,4,2
 (1)  4609  8C  46                      .ADDR   F10HD4  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  460B  06                          .BYTE   6       ;PACKING MODE 6
 (1)  460C  04                          .BYTE   4       ;SKIP COUNT = 4 FOR THIS XFR
 (1)  460D  02                          .BYTE   2       ;2 EXPECTED THIS XFR
1853  460E                    XFRP      F10HD5,6,5,2
 (1)  460E  92  46                      .ADDR   F10HD5  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4610  06                          .BYTE   6       ;PACKING MODE 6
 (1)  4611  05                          .BYTE   5       ;SKIP COUNT = 5 FOR THIS XFR
 (1)  4612  02                          .BYTE   2       ;2 EXPECTED THIS XFR
1854  4613                    XFRP      F10HD6,6,6,2
 (1)  4613  98  46                      .ADDR   F10HD6  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  4615  06                          .BYTE   6       ;PACKING MODE 6
 (1)  4616  06                          .BYTE   6       ;SKIP COUNT = 6 FOR THIS XFR
 (1)  4617  02                          .BYTE   2       ;2 EXPECTED THIS XFR
1855  4618                    XFRP      F10HD7,6,7,2
 (1)  4618  9E  46                      .ADDR   F10HD7  ;ADDRESS OF EXPECTED DATA TABLE
 (1)  461A  06                          .BYTE   6       ;PACKING MODE 6
 (1)  461B  07                          .BYTE   7       ;SKIP COUNT = 7 FOR THIS XFR
 (1)  461C  02                          .BYTE   2       ;2 EXPECTED THIS XFR
1856  461D                    XFRP      F10HD8,6,8,2
 (1)  461D  A4  46                      .ADDR   F10HD8  ;ADDRESS OF EXPECTED DATA TABLE
```

```
 (1)  461F  06                                  .BYTE   6       ;PACKING MODE 6
 (1)  4620  08                                  .BYTE   8       ;SKIP COUNT = 8 FOR THIS XFR
 (1)  4621  02                                  .BYTE   2       ;2 EXPECTED THIS XFR
1857  4622  FF  FF                        .ADDR   @177777 ;END OF THE TABLE
1858
1859  4624  00  00                  PTABLE: .ADDR   0
1860                                ;HERE IS THE EXPECTED DATA TABLES FOR ALL DATA TRANSFERS
1861
1862  4626                          F11N:   ;11-NORMAL EXPECTED DATA
1863  4626  00                              .BYTE   0       ;DDRA   4 BYTE XFR OF
1864  4627  04                              .BYTE   @4      ;DDRB   FILL,1,2,3
1865  4628  00                              .BYTE   0
1866  4629  08                              .BYTE   @10     ;2ND SCLK
1867  462A  0C                              .BYTE   @14
1868  462B  00                              .BYTE   0
1869
1870  462C                          F15N:   ;15-NORMAL EXPECTED DATA
1871  462C  04                              .BYTE   @4      ;DDRA   4 BYTE XFR OF
1872  462D  00                              .BYTE   0       ;DDRB   FILL,1,2,3
1873  462E  00                              .BYTE   0
1874  462F  0C                              .BYTE   @14     ;2ND SCLK
1875  4630  08                              .BYTE   @10
1876  4631  00                              .BYTE   0
1877
1878  4632                          F10C1:  ;10-COMPAT SKIP COUNT =1
1879  4632  04                              .BYTE   @4      ;DATA IS
1880  4633  00                              .BYTE   0       ;       FILL,1,2,3
1881  4634  00                              .BYTE   0
1882  4635  C0                              .BYTE   @300    ;2ND SCLK
1883  4636  80                              .BYTE   @200
1884  4637  00                              .BYTE   0
1885
1886  4638                          F10C2:  ;10-COMPAT SKIP COUNT =2
1887  4638  00                              .BYTE   0       ;DATA IS
1888  4639  00                              .BYTE   0       ;       FILL,FILL,1,2
1889  463A  00                              .BYTE   0
1890  463B  80                              .BYTE   @200    ;2ND SCLK
1891  463C  40                              .BYTE   @100
1892  463D  00                              .BYTE   0
1893
1894  463E                          F10C3:  ;10-COMPAT SKIP COUNT =3
1895  463E  00                              .BYTE   0       ;DATA IS
1896  463F  00                              .BYTE   0       ;       FILL,FILL,FILL,1
1897  4640  00                              .BYTE   0
1898  4641  40                              .BYTE   @100    ;2ND SCLK
1899  4642  00                              .BYTE   0
1900  4643  00                              .BYTE   0
1901  4644                          F10D1:  ;10-DUMP SKIP COUNT =1
1902  4644  04                              .BYTE   @4      ;DATA IS
1903  4645  00                              .BYTE   0       ;       FILL,1,2,3,4
1904  4646  00                              .BYTE   0
1905  4647  D0                              .BYTE   @320    ;2ND SCLK
1906  4648  80                              .BYTE   @200
1907  4649  00                              .BYTE   0
```

```
1908
1909  464A                          F10D2:  ;10-DUMP SKIP COUNT =2
1910  464A  00                               .BYTE   0       ;DATA IS
1911  464B  00                               .BYTE   0       ;       FILL,FILL,1,2,3
1912  464C  00                               .BYTE   0
1913  464D  8C                               .BYTE   @214    ;2ND SCLK
1914  464E  40                               .BYTE   @100
1915  464F  00                               .BYTE   0
1916
1917  4650                          F10D3:  ;10-DUMP SKIP COUNT =3
1918  4650  00                               .BYTE   0       ;DATA IS
1919  4651  00                               .BYTE   0       ;       FILL,FILL,FILL,1,2
1920  4652  00                               .BYTE   0
1921  4653  48                               .BYTE   @110    ;2ND SCLK
1922  4654  00                               .BYTE   0
1923  4655  00                               .BYTE   0
1924
1925  4656                          F10D4:  ;10-DUMP SKIP COUNT =4
1926  4656  00                               .BYTE   0       ;DATA IS
1927  4657  00                               .BYTE   0       ;       FILL,FILL,FILL,FILL,1
1928  4658  00                               .BYTE   0
1929  4659  04                               .BYTE   @4      ;2ND SCLK
1930  465A  00                               .BYTE   0
1931  465B  00                               .BYTE   0
1932
1933  465C                          F10HD1:
1934  465C                          F10HC1: ;10-HI-DEN-COMPAT SKIP COUNT =1
1935  465C  04                               .BYTE   @4      ;DATA IS
1936  465D  00                               .BYTE   0       ;       F,1,2,3,4,5,6,7,10
1937  465E  00                               .BYTE   0
1938  465F  C0                               .BYTE   @300    ;2ND SCLK
1939  4660  80                               .BYTE   @200
1940  4661  00                               .BYTE   0
1941  4662  41                               .BYTE   @101    ;3RD SCLK
1942  4663  01                               .BYTE   1
1943  4664  01                               .BYTE   1
1944  4665  22                               .BYTE   @42     ;4TH SCLK
1945  4666  1C                               .BYTE   @34
1946  4667  00                               .BYTE   0
1947  4668                          F10HD2:
1948  4668                          F10HC2: ;10-HI-DEN-COMPAT SKIP COUNT =2
1949  4668  00                               .BYTE   0       ;DATA IS
1950  4669  00                               .BYTE   0       ;       F,F,1,2,3,4,5,6,7
1951  466A  00                               .BYTE   0
1952  466B  80                               .BYTE   @200    ;2ND SCLK
1953  466C  40                               .BYTE   @100
1954  466D  00                               .BYTE   0
1955  466E  01                               .BYTE   @1      ;3RD SCLK
1956  466F  C1                               .BYTE   @301
1957  4670  00                               .BYTE   0
1958  4671  1D                               .BYTE   @35     ;4TH SCLK
1959  4672  18                               .BYTE   @30
1960  4673  00                               .BYTE   0
1961
```

```
1962  4674                          F10HD3:
1963  4674                          F10HC3: ;10-HI-DEN-COMPAT SKIP COUNT =3
1964  4674  00                              .BYTE   0       ;DATA IS
1965  4675  00                              .BYTE   0       ;       F,F,F,1,2,3,4,5,6
1966  4676  00                              .BYTE   0
1967  4677  40                              .BYTE   @100    ;2ND SCLK
1968  4678  00                              .BYTE   0
1969  4679  00                              .BYTE   0
1970  467A  C1                              .BYTE   @301    ;3RD SCLK
1971  467B  80                              .BYTE   @200
1972  467C  00                              .BYTE   0
1973  467D  18                              .BYTE   @30     ;4TH SCLK
1974  467E  14                              .BYTE   @24
1975  467F  00                              .BYTE   0
1976
1977  4680                          F10HC4: ;10-HI-DEN-COMPAT SKIP COUNT =4
1978  4680  00                              .BYTE   0       ;DATA IS
1979  4681  00                              .BYTE   0       ;       F,F,F,F,1,2,3,4,5
1980  4682  00                              .BYTE   0
1981  4683  00                              .BYTE   0       ;2ND SCLK
1982  4684  00                              .BYTE   0
1983  4685  00                              .BYTE   0
1984  4686  80                              .BYTE   @200    ;3RD SCLK
1985  4687  40                              .BYTE   @100
1986  4688  00                              .BYTE   0
1987  4689  17                              .BYTE   @27     ;4TH SCLK
1988  468A  10                              .BYTE   @20
1989  468B  00                              .BYTE   0
1990
1991  468C                          F10HD4: ;10-HI-DEN-DUMP SKIP COUNT =4
1992  468C  80                              .BYTE   @200    ;DATA IS
1993  468D  40                              .BYTE   @100    ;       F,F,F,F,1,2,3,4,5
1994  468E  00                              .BYTE   0       ;ONLY GET LAST 2 SCLKS
1995  468F  17                              .BYTE   @27     ;2ND SCLK
1996  4690  10                              .BYTE   @20
1997  4691  00                              .BYTE   0
1998  4692                          F10HD5:
1999  4692                          F10HC5: ;10-HI-DEN-COMPAT SKIP COUNT =5
2000  4692  40                              .BYTE   @100    ;DATA IS
2001  4693  00                              .BYTE   0       ;       F,F,F,F,F,1,2,3,4
2002  4694  00                              .BYTE   0       ;ONLY GET LAST 2 SCLKS
2003  4695  12                              .BYTE   @22     ;2ND SCLK
2004  4696  0C                              .BYTE   @14
2005  4697  00                              .BYTE   0
2006
2007  4698                          F10HD6:
2008  4698                          F10HC6: ;10-HI-DEN-COMPAT SKIP COUNT =6
2009  4698  00                              .BYTE   0       ;DATA IS
2010  4699  00                              .BYTE   0       ;       F,F,F,F,F,F,1,2,3
2011  469A  00                              .BYTE   0       ;ONLY GET LAST 2 SCLKS
2012  469B  0D                              .BYTE   @15     ;2ND SCLK
2013  469C  08                              .BYTE   @10
2014  469D  00                              .BYTE   0
2015
```

```
2016  469E                          F10HD7:
2017  469E                          F10HC7: ;10-HI-DEN-COMPAT SKIP COUNT =7
2018  469E  00                              .BYTE   0       ;DATA IS
2019  469F  00                              .BYTE   0       ;       F,F,F,F,F,F,F,1,2
2020  46A0  00                              .BYTE   0       ;ONLY GET LAST 2 SCLKS
2021  46A1  08                              .BYTE   @10     ;2ND SCLK
2022  46A2  04                              .BYTE   @4
2023  46A3  00                              .BYTE   0
2024
2025  46A4                          F10HD8:
2026  46A4                          F10HC8: ;10-HI-DEN-COMPAT SKIP COUNT =8
2027  46A4  00                              .BYTE   0       ;DATA IS
2028  46A5  00                              .BYTE   0       ;       F,F,F,F,F,F,F,F,1
2029  46A6  00                              .BYTE   0       ;ONLY GET LAST 2 SCLKS
2030  46A7  04                              .BYTE   @4      ;2ND SCLK
2031  46A8  00                              .BYTE   0
2032  46A9  00                              .BYTE   0
```

```
                                        .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
2035  46AA                              SSUB
 (1)                                    ;*********************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
2036                                    ;*CLEAR ALL TU PORTS
2037  46AA                              SD
 (1)                                    ;*********************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
2038                                    ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
2039                                    ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
2040                                    ;*AND LOOP MODES.
2041  46AA                              SP
 (1)                                    ;*********************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
2042                                    ;*BGNSUB
2043                                    ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
2044                                    ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
2045                                    ;*    CLEAR PORT SELECT FOR TRANSPORT
2046                                    ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
2047                                    ;*    CLEAR PORT DIAGNOSTIC CONTROL
2048                                    ;*    CLEAR PORT AMTIE WORD
2049                                    ;*ENDSUB
2050  46AA                              S
 (1)                                    ;*********************************************************************
2051  46AA  F5              12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
2052  46AB  C5              12.0          PUSH    B
2053  46AC  06  00           7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
2054  46AE  DB  E0          10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
2055  46B0  E6  80           7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
2056  46B2  B0               4.0          ORA     B               ;ADD IN THE SELECTED PORT #
2057  46B3  D3  E0          10.0          OUT     MBSEL           ;RESET TO THIS PORT
2058  46B5  3E  80           7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
2059  46B7  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
2060  46B9  AF               4.0          XRA     A               ;CLEAR TU COMMAND A
2061  46BA  D3  40          10.0          OUT     TCMD            ;
2062  46BC  3E  81           7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
2063  46BE  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
2064  46C0  3E  00           7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
2065  46C2  D3  40          10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
2066  46C4  AF               4.0          XRA     A
2067  46C5  D3  44          10.0          OUT     TAMT            ;CLEAR AMTIE WORD
2068  46C7  D3  48          10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
2069  46C9  D3  4C          10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
2070  46CB  04               4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
2071  46CC  78               4.0          MOV     A,B
2072  46CD  FE  04           7.0          CPI     4               ;DONE?
2073  46CF  C2  AE  46      10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
2074  46D2  C1              10.0          POP     B               ;RESET B & C
2075  46D3  F1              10.0          POP     PSW             ;ALL DONE
2076  46D4  C9              10.0          RET                     ;EXIT
```

```
                                        .SBTTL  SUBROUTINE 4X5 TRANSLATE A SUBGROUP
46D5                                    $SUB
 (1)                                    ; ************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
                                        ;*4X5 TRANSLATE A SUBGROUP
46D5                                    $D
 (1)                                    ; ************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
                                        ;*THIS SUBROUTINE IS USED TO 4X5 TRANSLATE A 4 BYTE DATA SUBGROUP INTO
                                        ;*A 5 BYTE SUBGROUP AS RECORDED ON TAPE OR READ FROM TAPE.
46D5                                    $P
 (1)                                    ; ************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
                                        ;*BGNSUB
                                        ;*    GET THE 4 (INPUT) DATA BYTES TO BE TRANSLATED
                                        ;*    CLEAR THE TRANSLATED (OUTPUT) BUFFER (OUTPUT BUFFER IS 9X5)
                                        ;*    SET LOOP COUNT TO 9
                                        ;*    BGNDO
                                        ;*    :    CALCULATE ODD PARITY FOR THE FOUR (INPUT) DATA BYTES (INPUT BUFFER IS NOW 9X4
                                        ;*    :    ROTATE EACH (5) ENTRY OF THE TRANSLATED (OUTPUT) BUFFER RIGHT
                                        ;*    :    GET THE LEAST SIGNIFICANT BIT OF EACH (INPUT) DATA BYTE AND
                                        ;*    :        FORM A TABLE INDEX (MS BIT OF INDEX = LS BIT OF THE FIRST
                                        ;*    :        BYTE OF THE INPUT DATA, ETC.)
                                        ;*    :    REMOVE THE LS BIT OF EACH INPUT DATA BYTE (THE ONE JUST USED
                                        ;*    :        TO BUILD THE INDEX)
                                        ;*    :    USE THE 4 BIT INDEX TO GET THE TRANSLATION VALUE
                                        ;*    :    PUT ONE BIT OF THE TRANSLATION VALUE, AS THE MOST SIGNIFICANT
                                        ;*    :        BIT EACH OF THE 5 TRANSLATED DATA BYTES (LS BIT OF THE
                                        ;*    :        TRANSLATION VALUE REPLACES MS BIT OF THE FIRST TRANSLATED
                                        ;*    :        DATA BYTE.
                                        ;*    :    DECREMENT THE LOOP COUNT
                                        ;*    :    DO UNTIL THE LOOP COUNT = 0
                                        ;*    ENDDO
                                        ;*ENDSUB
46D5                                    $
 (1)                                    ; ************************************************************************

                                        T4X5:   PUSH    PSW             ;SAVE REGISTER A + PSW
                                                PUSH    B               ;SAVE REGISTER B + C
                                                PUSH    D               ;SAVE REGISTER D + E
                                                PUSH    H               ;SAVE REGISTER H + L
                                        ;FOUR DATA BYTES ARE NOW IS THE BUFFER TABLE
                                        ;CLEAR THE TRANSLATED DATA TABLE.
                                                MVI     B,10            ;SET UP LOOP COUNT
                                                LXI     D,TRNOUT        ;GET POINTER TO TRANSLATED DATA TABLE
                                                XRA     A               ;CLEAR A
                                        D4X5:   STAX    D               ;STORE IN THE TABLE
                                                INX     D               ;UPDATE TABLE POINTER
                                                DCR     B               ;DECREMENT LOOP COUNT
                                                JNZ     D4X5            ;DO UNTIL LOOP COUNT=0
                                                MVI     C,9             ;SET UP TRACK COUNT
```

```
2122  46E7  06 04        7.0  B4X5:  MVI   B,4          ;SET UP BIT COUNT
2123  46E9  11 61 47     10.0       LXI   D,TRNIN
2124  46EC  AF           4.0        XRA   A            ;CLEAR THE GROUP POSITION COUNT
2125  46ED  32 6F 47     13.0       STA   GP4X5
2126  46F0  1A           7.0  C4X5:  LDAX  D            ;GET A DATA BYTE
2127  46F1  A7           4.0        ANA   A            ;SET CONDITION BITS
2128  46F2  E2 F6 46     10.0       JPO   PO4X5        ;ODD PARITY LEAVE CARRY CLEAR
2129  46F5  37           4.0        STC                ;EVEN PARITY SET CARRY
2130  46F6  1F           4.0  PO4X5: RAR                ;SHIFT OUT DESIRED BIT
2131  46F7  12           7.0        STAX  D            ;STORE RESULT BACK IN TEMP TABLE
2132  46F8  3A 6F 47     13.0       LDA   GP4X5        ;GET THE GROUP POSITION BYTE
2133  46FB  17           4.0        RAL                ;PUT IN THIS DATA BIT
2134  46FC  32 6F 47     13.0       STA   GP4X5        ;SAVE THE UPDATED GROUP POSITION BYTE
2135  46FF  13           6.0        INX   D            ;UPDATE THE TABLE POINTER
2136  4700  05           4.0        DCR   B            ;DECREMENT THE BIT COUNT
2137  4701  C2 F0 46     10.0       JNZ   C4X5         ;DO UNTIL ALL 4 BITS ARE RECEIVED
2138
2139                               ;AT THIS POINT GP4X5 CONTAINS 4 BITS - ONE FROM EACH OF THE FOUR BYTES
2140
2141                               ;PERFORM A DOUBLE PRECISION SHIFT ON THE TRANSLATED DATA
2142  4704  06 05        7.0        MVI   B,5          ;SET UP LOOP COUNT
2143  4706  11 65 47     10.0       LXI   D,TRNOUT     ;GET POINTER TO TRANSLATED DATA TABLE
2144  4709  13           6.0  E4X5:  INX   D            ;POINT TO PARITY BIT
2145  470A  1A           7.0        LDAX  D            ;GET PARITY BIT
2146  470B  A7           4.0        ANA   A            ;CLEAR THE CARRY BIT
2147  470C  1F           4.0        RAR                ;SHIFT TO THE CARRY BIT
2148  470D  12           7.0        STAX  D            ;STORE IT BACK
2149  470E  1B           6.0        DCX   D            ;DECREMENT TO DATA BITS
2150  470F  1A           7.0        LDAX  D            ;GET THE DATA BITS
2151  4710  1F           4.0        RAR                ;SHIFT IN CARRY BITS
2152  4711  12           7.0        STAX  D            ;STORE BACK IN TABLE
2153  4712  13           6.0        INX   D            ;POINT TO NEXT TABLE ENTRY
2154  4713  13           6.0        INX   D            ;
2155  4714  05           4.0        DCR   B            ;DECREMENT LOOP COUNT
2156  4715  C2 09 47     10.0       JNZ   E4X5         ;DO UNTIL LOOP COUNT=0
2157  4718  11 45 47     10.0       LXI   D,TAB4X5     ;LOAD ADDRESS OF TRANSLATION TABLE
2158  471B  26 00        7.0        MVI   H,0          ;CLEAR REGISTER H
2159  471D  3A 6F 47     13.0       LDA   GP4X5        ;GET GROUP POSITION COUNT
2160  4720  6F           4.0        MOV   L,A
2161  4721  19           10.0       DAD   D            ;ADD GROUP POSITION COUNT TO TABLE
2162  4722  7E           7.0        MOV   A,M          ;GET TRANSLATED DATA
2163  4723  32 70 47     13.0       STA   TRNTMP
2164                               ;"TRNTMP" = THE TRANSLATED DATA
2165
2166  4726  11 65 47     10.0       LXI   D,TRNOUT     ;GET POINTER TO TRANSLATED DATA TABLE
2167  4729  06 05        7.0        MVI   B,5          ;SET UP LOOP COUNT
2168  472B  3A 70 47     13.0 F4X5: LDA   TRNTMP       ;GET TRANSLATED DATA
2169  472E  13           6.0        INX   D            ;POINT TO PARITY POSITION
2170  472F  A7           4.0        ANA   A            ;CLEAR CARRY BIT
2171  4730  1F           4.0        RAR                ;SHIFT OUT A BIT
2172  4731  32 70 47     13.0       STA   TRNTMP       ;STORE TRANSLATED DATA
2173  4734  1A           7.0        LDAX  D            ;GET THE PARITY BIT (ALWAYS ZERO)
2174  4735  17           4.0        RAL                ;ROLL IN THE CARRY BIT
2175  4736  12           7.0        STAX  D            ;STORE AS THE PARITY BIT
```

```
2176  4737  13              5.0     INX   D          ;UPDATE TABLE POINTER
2177  4738  05              4.0     DCR   B          ;DECREMENT THE LOOP COUNT
2178  4739  C2  2B  47      10.0     JNZ   F4X5       ;DO UNTIL LOOP COUNT=0
2179
2180                    ;NOW THE TRANSLATED DATA TABLE HAS BEEN UPDATED
2181
2182  473C  0D              4.0     DCR   C          ;DECREMENT THE TRACK COUNT
2183  473D  C2  E7  46      10.0     JNZ   B4X5       ;DO UNTIL ALL TRACKS TRANSLATED
2184  4740  E1              10.0     POP   H          ;RESTORE REGISTER H + L
2185  4741  D1              10.0     POP   D          ;RESTORE REGISTER D + E
2186  4742  C1              10.0     POP   B          ;RESTORE REGISTER B + C
2187  4743  F1              10.0     POP   PSW        ;RESTORE REGISTER A + PSW
2188  4744  C9              10.0     RET              ;RETURN TO USER
```

```
                                .SBTTL  TABLE 4 X 5 TRANSLATION

                                ;THIS TABLE TRANSLATED DATA FOR AS FOLLOWS
                                ;  INPUT GROUP POSITIONS              OUTPUT GROUP POSITIONS
                                ;   1 2 3 4 / 5 6 7 8                 10 9 8 7 6 / 5 4 3 2 1
                                ;
                                ;       0000                              10011
                                ;       0001                              11011
                                ;       0010                              01001
                                ;       0011                              11001
                                ;       0100                              10111
                                ;       0101                              10101
                                ;       0110                              01101
                                ;       0111                              11101
                                ;       1000                              01011
                                ;       1001                              10010
                                ;       1010                              01010
                                ;       1011                              11010
                                ;       1100                              01111
                                ;       1101                              10110
                                ;       1110                              01110
                                ;       1111                              11110

4745  13                        TAB4X5: .BYTE   @23
4746  1B                                .BYTE   @33
4747  09                                .BYTE   @11
4748  19                                .BYTE   @31
4749  17                                .BYTE   @27
474A  15                                .BYTE   @25
474B  0D                                .BYTE   @15
474C  1D                                .BYTE   @35
474D  0B                                .BYTE   @13
474E  12                                .BYTE   @22
474F  0A                                .BYTE   @12
4750  1A                                .BYTE   @32
4751  0F                                .BYTE   @17
4752  16                                .BYTE   @26
4753  0E                                .BYTE   @16
4754  1E                                .BYTE   @36
```

```
                                .SBTTL  SUBROUTINE VARIABLES

4755  00                SDDRA:  .BYTE   0               ;STORAGE FOR DDR A DATA
4756  00                SDDRB:  .BYTE   0               ;STORAGE FOR DDR B DATA
4757  00                SDDRC:  .BYTE   0               ;STORAGE FOR DDR C DATA
4758  00                SCLKCT: .BYTE   0               ;COUNT OF 'SCLK' RECEIVED
4759  00                ESCLK:  .BYTE   0               ;NUMBER OF EXPECTED SCLK PER PASS
475A  00                FMTNUM: .BYTE   0               ;SELECTED PACKING MODE PER PASS
475B  00                SKIPCT: .BYTE   0               ;SKIP COUNT IN USE FOR CURRENT XFR
475C  00  00            EXPADR: .ADDR   0               ;ADDRESS OF THE EXPECTED DATA TABLE
475E  00                UNITMP: .BYTE   0               ;PORT NUMBER TO USE AS RESPONDED BY HOST
475F  00                SUNIT:  .BYTE   0               ;STORAGE OF THE SELECTED UNIT NUMBER
4760  00                CCTLWD: .BYTE   0               ;LOAD INTO THE SYSTEM CLOCK CONTROL WORD

4761 0004               TRNIN:  .BLKB   4               ;TABLE CONTAINING THE 4 DATA BYTES TO
                                                        ;BE TRANSLATED
4765 000A               TRNOUT: .BLKB   10              ;TRANSLATED DATA TABLE DATA AFTER 4X5
                                                        ;TRANSLATION
476F  00                GP4X5:  .BYTE   0               ;ACCUMULATOR FOR THE 4 BIT SUBGROUP TO
                                                        ;BE TRANSLATED
4770  00                TRNTMP: .BYTE   0               ;TEMP. STORAGE FOR THE TRANSLATED SUBGROUP.
```

```
                                        .SBTTL  SUBROUTINE CLEAR ECC
4771                                    $
 (1)                                    ; ************************************************************************
                                        ;THIS SUBROUTINE CLEARS THE ECC CHARACTER-MEMORY LOCATION "ECCCHR" -
                                        ;USED BY THE ECC SUBROUTINE.  IT SHOULD BE CALLED PRIOR TO CALCULATING
                                        ;THE ECC CHARACTER FOR A GIVEN DATA GROUP.
4771                                    $
 (1)                                    ; ************************************************************************

4771  AF                      4.0 CLECC:  XRA     A               ;CLEAR THE ACCUMULATOR
4772  32    9D    47          13.0        STA     ECCCHR          ;CLEAR THE ECC CHARACTER
4775  C9                      10.0        RET                     ;RETURN TO USER
```

```
                                        .SBTTL  SUBROUTINE CALCUALTE ECC CHARACTER
4776                                    S
 (1)                                    ; ************************************************************************
                                        ;THIS SUBROUTINE TAKES THE CHARACTER PASSED IN THE ACCUMULATOR AND
                                        ;THE CONTENTS OF "ECCCHR" TO UPDATE THE CONTENTS OF "ECCCHR" ACCORDING
                                        ;TO THE ANSI STANDARD ECC POLYNOMIAL.
4776                                    S
 (1)                                    ; ************************************************************************

4776  E5                  12.0  ECC:    PUSH    H
4777  C5                  12.0          PUSH    B
4778  21  9D  47           10.0          LXI     H,ECCCHR        ;LOAD ADDRESS OF ECC CHAR.
477B  AE                   7.0          XRA     M               ;EXCLUSIVE OR CHAR. AND ECC
477C  5F                   4.0          MOV     E,A             ;SAVE XOR RESULT IN E
477D  E6  10               7.0          ANI     $10             ;IS BIT #4 OF RESULT SET
477F  7B                   4.0          MOV     A,E             ;RESTORE XOR RESULT FROM B
4780  CA  85  47           10.0          JZ      ECC1            ;CONTINUE IF BIT #4 RESET
4783  EE  23               7.0          XRI     $23             ;ELSE-XOR WITH 23
4785  5F                   4.0  ECC1:    MOV     E,A             ;STORE THE ECC RESULT IN E
4786  AF                   4.0          XRA     A               ;CLEAR A
4787  4F                   4.0          MOV     C,A             ;CLEAR THE TRANSLATE RESULT
4788  21  95  47           10.0          LXI     H,ECCTBL        ;POINT TO ECC TABLE TO RE-POSITION
478B  CD  9E  47           18.0          CALL    TRANS           ;TRANSLATE THE BITS
478E  79                   4.0          MOV     A,C             ;GET THE TRANSLATED RESULT
478F  32  9D  47           13.0          STA     ECCCHR          ;STORE RESULT
4792  C1                   10.0          POP     B
4793  E1                   10.0          POP     H
4794  C9                   10.0          RET

4795  08                         ECCTBL:  $08                     ;BIT 0 = POSITION 3
4796  20                                  $20                     ;BIT 1 = POSITION 5
4797  02                                  $02                     ;BIT 2 = POSITION 1
4798  40                                  $40                     ;BIT 3 = POSITION 6
4799  80                                  $80                     ;BIT 4 = POSITION 7
479A  01                                  $01                     ;BIT 5 = POSITION 0
479B  10                                  $10                     ;BIT 6 = POSITION 4
479C  04                                  $04                     ;BIT 7 = POSITION 2

479D  00                         ECCCHR:  .BYTE   0
```

```
                                        .SBTTL  SUBROUTINE POLYNOMIAL BIT TRANSLATION

                                        ;
                                        ;THIS ROUTINE IS USED TO TRANSLATE THE BITS IN THE OUTPUT OF THE ECC
                                        ;GENERATOR INTO THE PROPER POSITIONS WITHIN THE DATA BYTE.
                                        ;

479E  06 01            7.0  TRANS:  MVI   B,1            ;INIT 'B' TO BIT POSITION 0
47A0  7B                4.0  TRANS1: MOV   A,E            ;GET CHAR TO BE TRANSLATED
47A1  A0                4.0          ANA   B              ;SEE IF BIT POSITION IN 'B' IS SET
47A2  CA A8 47         10.0          JZ    TRANS2         ;DO NEXT BIT POSITION IF NOT SET
47A5  79                4.0          MOV   A,C            ;GET PREVIOUS RESULT OF 'OR'
47A6  B6                7.0          ORA   M              ;'OR' IN NEW POSITION
47A7  4F                4.0          MOV   C,A            ;SAVE RESULT

47A8  78                4.0  TRANS2: MOV   A,B
47A9  07                4.0          RLC                  ;POSITION MASK TO NEXT BIT
47AA  47                4.0          MOV   B,A
47AB  D8               12.0          RC                   ;EXIT WHEN ALL POSITIONS DONE
47AC  23                6.0          INX   H              ;POINT TO NEXT TABLE ENTRY
47AD  C3 A0 47         10.0          JMP   TRANS1         ;PROCESS NEXT BIT

47B0                                    S
 (1)                                    ; ***************************************************************************
                                        ;CLKSYS - ROUTINE TO CLOCK THE SYSTEM USING THE CONTENTS OF 'CCTLWD'
                                        ;       AS A BASE AND TOGGLING THE CLOCK BIT IN THE CLOCK CONTROL WORD
47B0                                    S
 (1)                                    ; ***************************************************************************

47B0  F5                12.0  CLKSYS: PUSH  PSW
47B1  3A 60 47          13.0          LDA   CCTLWD         ;GET THE CONTROL WORD BASE
47B4  F6 40              7.0          ORI   SSCLK          ;GET THE CLOCK BIT
47B6  D3 F0             10.0          OUT   CLKCTL         ;SET THE CLOCK ON
47B8  E6 BF              7.0          ANI   @277           ;CLEAR THE CLOCK BIT
47BA  D3 F0             10.0          OUT   CLKCTL         ;FINISH THE CLOCK CYCLE
47BC  F1                10.0          POP   PSW
47BD  C9                10.0          RET

      0000                        .END
```

```
A      =%0007    ADATA  = 0094    AMTIEP= 0001    AMTIE7= 0002
ARAIDF= 0098    ASAVE   4F9B    ATTCD   4F97    AXNUM   4F91
B      =%0000    BADST  = 0090    BIT0   = 0001    BIT1   = 0002
BIT15  = 8000    BIT2   = 0004    BIT3   = 0008    BIT4   = 0010
BIT5   = 0020    BIT6   = 0040    BIT7   = 0080    BIT8   = 0100
BIT9   = 0200    BRKPBC= 4F0A    BRKRAM= 4F10    BRKSTR= 4E60
BRKXCT= 4F00    BSAVE   4F9C    BYTCNT= 00D4    BYTEH   4F24
BYTEL   4F23    B4X5    46E7    C      =%0001    CASCT   4F21
CASCTL= 00A0    CASSTA= 00A0    CATTH  = 0089    CATTL  = 0088
CBUSST= 00A1    CBYTH  = 008B    CBYTL  = 008A    CCTLWD  4760
CDG1H  = 0087    CDG1L  = 0086    CDG2H  = 0093    CDG2L  = 0092
CDG3H  = 0095    CDG3L  = 0094    CDVTH  = 008D    CDVTL  = 008C
CHPTIE= 0028    CH0TIE= 0020    CH1TIE= 0021    CH2TIE= 0022
CH3TIE= 0023    CH4TIE= 0024    CH5TIE= 0025    CH6TIE= 0026
CH7TIE= 0027    CKLOP  = 2815    CLEAR   46AA    CLECC   4771
CLKCTL= 00F0    CLKRS   43E5    CLKSYS  47B0    CLOCK   4F26
CLRLP   46AE    CMC0H  = 0099    CMC0L  = 0098    CMC1H  = 009B
CMC1L  = 009A    CMC2H  = 009D    CMC2L  = 009C    CMC3H  = 009F
CMC3L  = 009E    CMINH  = 0097    CMINL  = 0096    CNTCTL= 00D7
CNV1    456F    CONVRT  4569    CRCWRD= 0018    CSAVE   4F9D
CSRLH  = 0091    CSRLL  = 0090    CTCH   = 0085    CTCL   = 0084
CTSTH  = 008F    CTSTL  = 008E    CXCTH  = 0081    CXCTL  = 0080
CXINH  = 0083    CXINL  = 0082    C.     = 0001    C.AVAI= 0080
C.DP   = 0008    C.DSE  = 0010    C.DTU  = 0003    C.DVA  = 0008
C.FAIL= 00FC    C.FMT  = 0070    C.FNCT= 003E    C.GO   = 0001
C.INTC= 00FE    C.MAIN= 0020    C.NSA  = 0080    C.RCT  = 00FC
C.SER  = 0080    C.SHR  = 0040    C.SKPC= 000F    C.TAPE= 0040
C.WCS  = 0002    C1EC1   43F0    C1EC1L  43F8    C1EC2   4407
C1EC2L  440F    C4X5    46F0    D      =%0002    DATACT= 00D0
DBUS    4F28    DBUSCT= 00C0    DBUSST= 00C0    DDRA   = 00D8
DDRAIN= 0010    DDRB   = 00D9    DDRBIN= 0002    DDRC   = 00DA
DDRCIN= 0001    DDRCO  = 0088    DDRCTL= 00DB    DG1     4595
DG1ECC  459C    DG2     459D    DG2ECC  45A4    DIAFLG  4F22
DIAGPG= 4300    DIAGRM= 4F90    DIARD  = 000B    DONE1  = 0045
DONINT= 0010    DSAVE   4F9E    DSE    = 0006    DUMM1   447D  G
DUMM2   449D  G  D.ATH0= 0001    D.ATH1= 0002    D.EOTD= 0010
D.LAGC= 0020    D.NOTW= 0040    D.NTHR= 0004    D.TACH= 0008
D.WR4  = 0080    D4X5    46DF    E      =%0003    ECC     4776
ECCBAD= 0042    ECCCHR  479D    ECCCOR= 0019    ECCOK  = 0041
ECCSTA= 001A    ECCTBL  4795    ECCTST= 000E    ECC1    4785
EDATA  = 0095    EOTCLR= 0003    ERFLG   4F93    ERLP   = 2809
ERLPA  = 280F    ERLPB  = 2812    ERLPE  = 280C    ERNUM   4F90
ERRCNT= 00D6    ESAVE   4F9F    ESCLK   4759    EXIT    454F
EXPADR  475C    E.ACRC= 0010    E.AMT  = 0020    E.CDP  = 0080
E.CRC  = 0080    E.PNTR= 0008    E.RPE  = 0040    E.STEC= 0001
E.TTEC= 0002    E.UNC  = 0004    E4X5    4709    FIFORD= 006A
FMTNUM  475A    FORMAT  4F25    FOUND   4327    FPORT   4347
FWDTST= 0061    F10C1   4632    F10C2   4638    F10C3   463E
F10D1   4644    F10D2   464A    F10D3   4650    F10D4   4656
F10HC1  465C    F10HC2  4668    F10HC3  4674    F10HC4  4680
F10HC5  4692    F10HC6  4698    F10HC7  469E    F10HC8  46A4
F10HD1  465C    F10HD2  4668    F10HD3  4674    F10HD4  468C
F10HD5  4692    F10HD6  4698    F10HD7  469E    F10HD8  46A4
F11N    4626    F15N    462C    F4X5    472B    GCRID   0089
```

```
GCRSET= 0002    GOODTM= 0092    GP4X5   476F    H      =%0004
HLSAVE  4FA0    IE     = 0008   INTSTA= 00E0    ITERA   4F9A
I.PWR = 0020    I.RMPE= 0040    I5.5  = 0010    I6.5   = 0020
I7.5  = 0040    KCALL = 005F    KCLR  = 007B    KDEP   = 003F
KENAB = 0078    KEXAM = 003E    KEYBRD= 00C8    KEY1   = 0078
KEY10 = 006D    KEY11 = 006E    KEY12 = 006F    KEY13  = 005C
KEY14 = 005D    KEY15 = 005E    KEY16 = 005F    KEY17  = 003C
KEY18 = 003D    KEY19 = 003E    KEY2  = 0079    KEY20  = 003F
KEY3  = 007A    KEY4  = 007B    KEY5  = 0074    KEY6   = 0075
KEY7  = 0076    KEY8  = 0077    KEY9  = 006C    KINTA  = 006F
KLDAD = 003D    KN0   = 003C    KN1   = 005C    KN2    = 005D
KN3   = 005E    KN4   = 006C    KN5   = 006D    KN6    = 006E
KN7   = 0074    KN8   = 0075    KN9   = 0076    KU2    = 0079
KU3   = 007A    KU8   = 0077    L      =%0005   LBLANK= 000F
LCE   = 000B    LCH   = 000C    LCL   = 000D    LCP    = 000E
LC0   = 0000    LC1   = 0001    LC2   = 0002    LC3    = 0003
LC4   = 0004    LC5   = 0005    LC6   = 0006    LC7    = 0007
LC8   = 0008    LC9   = 0009    LDFIFO  4583    LDLEDA= 00CA
LDLEDB= 00CB    LDLEDC= 00CC    LDLEDD= 00CD    LDLEDE= 00CE
LDLEDF= 00CF    LKDIAG= 2800    LKKBD = 004C    LKKEY  = 0049
LKLWMG= 0058    LKLWMP= 0055    LKLWPG= 0052    LKLWPP= 004F
LKMOD7= 0046    LKOPR = 0046    LOADIT  457E    LPFLG   4F94
LPNUM   4F92    M      =%0006   MBSEL = 00E0    MB.A   = 0008
MB.B  = 0004    MEMTOP= 4FFF    MINUS = 000A    MM     = 8000
MSE   = 0008    MSGN  = 0006    MTACLR= 0000    MT.ARA= 0020
MT.CPE= 0080    MT.DSE= 0001    MT.FWD= 0040    MT.INH= 0008
MT.LWR= 0004    MT.MOT= 0002    MT.NWT= 0080    MT.PEC= 0040
MT.PSB= 0004    MT.PS0= 0001    MT.PS1= 0002    MT.REV= 0020
MT.WRT= 0010    MT.Z  = 0008    M.ATA = 0080    M.CAPE= 0020
M.CONT= 0080    M.DEM = 0020    M.EBL = 0010    M.EXC  = 0008
M.FAIL= 0008    M.ILR = 0010    M.INIT= 0010    M.OCC  = 0020
M.ONLI= 0001    M.PE  = 0040    M.PORT= 0080    M.RDEN= 0002
M.RDPE= 0008    M.RUN = 0004    M.SCLK= 0001    M.TRA  = 0040
M.UNIT= 0007    M.WCLK= 0040    M.WCLN= 0080    M.WREN= 0080
M5.5  = 0001    M6.5  = 0002    M7.5  = 0004    NOTCAP= 0088
NOUNIT  431E  G OKAY  = 00FF    OPRRAM= 4300    OPSTRT= 0058
OPVER = 0040    PADCNT= 00D5    PADCRC= 0080    PDIAG  = 0048
PEID  = 008A    PENAB = 004C    PESET = 0001    PL     = 00B1
PO4X5   46F6    PRDD  = 004C    PRENF = 009C    PS     = 00B2
PSTAT = 0048    PSW    =%0009   PTABLE  4624    P.AMTP= 0001
P.BCTC= 0040    P.CMDP= 0020    P.INTE= 0080    P.LCS  = 0040
P.LWR = 0020    P.RDP = 0002    P.RPEN= 0004    P.RPST= 0002
P.RPOE= 0020    P.RP1E= 0010    P.RP2E= 0020    P.RP3E= 0010
P.SING= 0080    P.STAT= 0002    P.STPE= 0080    P.TACH= 0008
P.TUPR= 0010    P.WCSP= 0004    P.WDS = 0040    P.WFLP= 0001
P.WPEN= 0010    P.WPOE= 0008    P.WP1E= 0004    P.WP2E= 0008
P.WP3E= 0004    P.5VOK= 0002    QUE   = 281B    QUEM   = 281E
RAMT  = 0010    RARA  = 0006    RARAI = 0004    RCHBD0= 0048
RCHBD1= 0047    RCHOK = 0046    RCHTST= 000C    RCLRT  = 000D
RCMD  = 000B    RCMLP = 0003    RCONT = 0080    RDATA  = 0017
RDCLK = 0010    RDON  = 0011    READG = 0007    REND   = 0014
REQST = 2806    RESCHR= 00D1    REVTST= 0064    REWIND= 0004
RFIFOL= 0008    RGCLK = 0002    RGCRI = 0003    RIBG   = 0001
RILL  = 0012    RINST = 000C    RMCTST= 0008    RMK2   = 0013
```

```
RNOP   = 0000    RPATH  = 0001    RPBAD  = 0044    RPCHI  = 0001
RPCLK  = 0003    RPCTL  = 0009    RPEI   = 0002    RPFAIL= 0000
RPF1   = 009D    RPF2   = 009E    RPOK   = 0043    RPOSTN= 0016
RPSTA  = 0015    RRCMT  = 00DA    RSTAT  = 0002    RTIEB  = 000A
RTIER  = 0030    RTM    = 0005    RUNKI  = 0009    RUPTST= 005E
RWDUNL= 0005     R.AMT  = 0001    R.BOP  = 0008    R.DATA= 0040
R.DON  = 0002    R.DRDY= 0010     R.END  = 0010    R.ILL  = 0004
R.JVOK= 0004     R.MK2  = 0008    R.PLOD= 0008     R.PLOO= 0010
R.PLO1= 0020     R.POST= 0020     R.STNM= 0002     R.STOP= 0004
R.STPC= 0001     R.TBJN= 0080     R.TSTD= 0040     R.VOK  = 0080
R00H   = 0081    R00L   = 0080    R01H   = 0083    R01L   = 0082
R02H   = 0085    R02L   = 0084    R03H   = 0087    R03L   = 0086
R04H   = 0089    R04L   = 0088    R05H   = 008B    R05L   = 008A
R06H   = 008D    R06L   = 008C    R07H   = 008F    R07L   = 008E
R10H   = 0091    R10L   = 0090    R11H   = 0093    R11L   = 0092
R12H   = 0095    R12L   = 0094    R13H   = 0097    R13L   = 0096
R14H   = 0099    R14L   = 0098    R15H   = 009B    R15L   = 009A
R16H   = 009D    R16L   = 009C    R17H   = 009F    R17L   = 009E
R7.5   = 0010    SCLKCT  4758     SDDRA   4755     SDDRB   4756
SDDRC   4757     SELCLR= 0000     SETATA= 00A1     SG2     4599
SG4     45A1     SID    = 0080    SKIPCT  475B     SOD    = 0080
SOE    = 0040    SP     =%0008    SSCLK  = 0040    SSTEP  = 0005
STACK  = 4FFF    STATRM= 4F20     STPCT   4F20     STRSP  = 5000
SUNIT   475F     TAB4X5  4745     TADR00= 0080     TADR01= 0081
TADR02= 0082     TADR03= 0083     TADR04= 0084     TADR05= 0085
TADR06= 0086     TADR07= 0087     TADR10= 0088     TADR11= 0089
TADR12= 008A     TADR13= 008B     TAMT   = 0044    TASEL  = 0080
TCMD   = 0040    TC.FWD= 0040     TC.INH= 0008     TC.LWR= 0004
TC.REV= 0020     TC.WRT= 0010     TEMP    4F99     TEST1   4300
TMF    = 0099    TMRDY  = 0040    TRANS   479E     TRANS1  47A0
TRANS2  47A8     TRKENA= 00D2     TRNIN   4761     TRNOUT  4765
TRNTMP  4770     TSET   = 2803    TSTEND= 2818     TSTS   = 0040
TUSEL0= 00D1     TUSEL1= 00D2     TU78   = 0010    T.ATH0= 0001
T.ATH1= 0002     T.BOT  = 0004    T.EOT  = 0002    T.FPT  = 0001
T.NTHR= 0004     T.ONL  = 0020    T.PES  = 0008    T.PSBJ= 0020
T.PSOJ= 0008     T.PS1J= 0010     T.RDY  = 0080    T.RDYO= 0040
T.RWD  = 0010    T.SCLK= 0002     T1CKA   44D0     T1CKAC  44F9   G
T1CKB   44FF     T1CKBC  4513   G T1CKC   4519     T1CKCC  452D   G
T1CLK   4468     T1CLKA  4462     T1LOOP  436D     T1PLO   441E
T1SCKL  4488     T1SCK0  44A6     T1SCLK  4486     T1SG1   4436
T1SG2   443F     T1SG3   4448     T1SG4   4451     T1ST    43EE
T4X5    46D5     UIBG   = 00A1    UNITMP  475E     VALFC   4F98
VALTB   4F95     VELTST= 005B     WDR.P  = 0010    WMCCTL= 00D3
WMCERR= 00DA     WMCSTA= 00D0     WRTCLK= 0000     WRTDAT= 00D3
W.ACRC= 0004     W.CRC  = 0008    W.DIAG= 0002     W.DONN= 0040
W.ECC  = 0010    W.ENAB= 0080     W.ERR  = 0020    W.FMT  = 0070
W.GCR  = 0010    W.LEFT= 0004     W.ONES= 0020     W.RESI= 0002
W.REV  = 0004    W.ROME= 0010     W.RST  = 0001    W.SKIP= 000F
W.WRIT= 0008     W.XFER= 0020     X      =%000A    XFRPTB  45A5
X.DONN= 0080     X.ENAB= 0040     X.PEPE= 0002     X.RGME= 0001
X.WCLK= 0001     Y      =%000B    .      = 47BE
```

ERRORS DETECTED: 0

*WMC8.A78/PIP,WMC8-NLIST,PARAM,MACRO,LIST,WMC8
RUN-TIME: 4 6 0 SECONDS
CORE USED: 10K

3	"MEMORY MAP" EXTENDED ADDRESS BIT
19	READ MICRO CONTROLLER CONTROL & STATUS WORDS
139	TRANSPORT/PORT COMMAND + STATUS WORDS
380	COMMON ADDRESS SPACE REGISTERS
444	MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485	MASSBUS CONTROL WORD DEFINITIONS
557	WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709	KEYBOARD/DISPLAY CONTROL + STATUS WORD
797	MISCELLANEOUS VARIABLE DEFINITIONS
801	RAM ADDRESS DEFINITIONS
819	LINKAGE TABLE ADDRESSES
842	BIT DEFINITIONS
856	RIM AND SIM INSTRUCTION BIT DEFINITIONS
877	INTERRUPTABLE INSTRUCTION MACROS
924	DIAGNOSTIC VARIABLES
945	COMMON VARIABLE STORAGE
959	MACRO CALL DEFINITIONS
1047	TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101	8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1359	TEST 1 - PHASE LOCKED LOOP TEST PART 1
1550	TEST 2 - PHASE LOCKED LOOP TEST PART 2
1678	TEST 3 - WRITE FAIL LINES CLEAR TEST
1777	TEST 4 - WRITE FAIL LINES SET TEST
1989	SUBROUTINE GCRSUB
2090	SUBROUTINE CLEAR ALL TU PORTS
2135	LWR - GCR LOOP WRITE TO READ AT PORT BOARD
2343	PROGRAM VARIABLES

```
                ;*************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;                "BYTE/SCLK COUNT NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                MM = DATA FORMAT FROM CAS REGISTER 2
                ;                NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;                BYTE-SCLK COUNT = LLL
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                LLL = AS ABOVE
                ;                MM  = AS ABOVE
                ;                NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;                TRANSITION COUNT = LLL
                ;                WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;                EXPECTED 18 BITS =      E EEEEEE
                ;                ACTUAL 18 BITS =        A AAAAAA
                ;
                ;                WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;                BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;                OF CAS REG 15 LOW BYTE.
                ;
                ;                ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;                AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;                17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;                TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;                AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;                "SUBGROUP NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;*************************************************************************
```

```
1016                    ;*************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ;*************************************************************************
1035                    ;*************************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ;*************************************************************************
1329    4300                    .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
1332                                .RADIX 16
1333                                .TITLE  RPM6 - READ PATH MICRO DIAGNOSTIC PART #6
1334                                ;ID     RPM6-READ PATH MICRO DIAGNOSTIC PART #6
1335
1336                                ;DEFINITIONS
1337
1338    0001                                I01=@001                        ;INTERUPT CODE VALUES
1339    0015                                I25=@25
1340    0018                                I30=@30
1341
1342    0C00                                F03=@3*400                      ;FAILURE CODE EQUIVALENTS
1343    1000                                F04=@4*400
1344    1400                                F05=@5*400
1345    1800                                F06=@6*400
1346    1C00                                F07=@7*400
1347    2000                                F10=@10*400
1348    2400                                F11=@11*400
1349    2800                                F12=@12*400
1350    4400                                F21=@21*400
1351    4800                                F22=@22*400
1352    4C00                                F23=@23*400
1353
1354                                ;SUBROUTINE LINK TABLE ADDRESSES
1355
1356    080C                                WRTSET  -       080C
1357
```

```
                                .SBTTL  TEST 1 - PHASE LOCKED LOOP TEST PART 1
4300                            ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*PHASE LOCKED LOOP TEST PART 1
4300                            SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A GCR LOOP WRITE TO READ
                                ;*WITH PLO BYPASS 01 SET
                                ;*IF NO FAILURE OCCURS THEN TEST 2 WILL BE
                                ;*PERFORMED TO RUN A LWR WITH THE PLL'S ENABLED
4300                            SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  SET CLOCKS TO NORMAL
                                ;*  SET THE BYTE COUNT TO 55.
                                ;*  CLEAR ALL AMTIE TEMP LOCATIONS
                                ;*  CLEAR ALL RAM TEMP LOCATIONS
                                ;*  SET XFRCTL SO PLL'S ARE DISABLED
                                ;*  ISSUE A LOOP WRITE TO READ
                                ;*  IF ANY ERRORS OCCURED
                                ;*  :  THEN
                                ;*  :    REPORT THE ERROR
                                ;*  :  ELSE
                                ;*  :    CONTINUE
                                ;*  ENDIF
                                ;*ENDTST
4300                            SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*RPM6 MICRO TEST 01
                                ;*RPM6 MICRO ERROR 01
                                ;*RPM6-PHASE LOCKED LOOPS TEST PART 1
                                ;*M8950'S , M8953
                                ;*OPERATOR ERROR NO TU78 UNITS SPECIFIED
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*RPM6 MICRO TEST 01
                                ;*RPM6 MICRO ERROR 02
                                ;*RPM6-PHASE LOCKED LOOPS TEST PART 1
                                ;*M8950'S , M8953
                                ;*LOOP WRITE/READ ERROR WITH PLL'S DISABLED
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL = FAILURE CODE
```

```
                                ;*EXPECTED = INTERUPT CODE
                                ;*
                                ;*RPM6 MICRO TEST 02
                                ;*RPM6 MICRO ERROR 03-11
                                ;*RPM6-PHASE LOCKED LOOPS TEST PART 1
                                ;*M8950'S , M8953
                                ;*LOOP WRITE / READ ERROR WITH PLL DISABLED
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*CHANNEL (N) FAILED
                                ;*ACTUAL DATA IS CHANNEL (N) TIE BUS REGISTER
1415  4300                      S
 (1)                            ; ********************************************************************************
```

```
1417  4300                              TEST1:  TESTX   @1                      ;INITIALIZE THE TEST
 (1)  4300  3E  01          7.0                  MVI     A,@1                    ;DEFINE THE TEST NUMBER
 (1)  4302  CD  [illegible] 28        18.0       CALL    TSET                    ;SETUP THE TEST
1418                                ;%RPM6-PHASE LOCKED LOOPS TEST PART 1
1419                                ;&M8950'S , M8953
1420  4305                                  REQ     @7,0,0,0,0
 (1)  4305  CD  06  28        18.0           CALL    REQST
 (1)  4308  00                               .BYTE   0                       ;DATA PATTERN NUMBER
 (1)  4309  00  00                           .WORD   0                       ;SYSTEM "0" COUNT
 (1)  430B  00  00                           .WORD   0                       ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  430D  00                               .BYTE   0                       ;DATA COMPARE FLAG IF =1
 (1)  430E  07                               .BYTE   @7                      ;REQUEST CODE
1421  430F                                  RIN     R12L
 (1)  430F  DB  94          10.0             IN      R12L                    ;READ R12L INTO AC
 (1)  4311  7F               4.0             MOV     A,A                     ;RETRY LINK
1422  4312  32  00  49      13.0       STA     UNITMP
1423  4315  A7               4.0       ANA     A
1424  4316  C2  24  43      10.0       JNZ     TST01
1425  4319                             ERR     EXIT,DUMMY
 (1)                                     ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4319  CD  09  28        18.0           CALL    ERLP                    ;PROCESS ERROR - DO 2.3
 (1)       0001                            MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  431C  01                               .BYTE   MSGN                    ;MESSAGE NUMBER ID
 (1)  431D  00                               .BYTE
 (1)  431E  CD  15  28        18.0       DUMMY:: CALL    CKLOP                   ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4321  DA  31  47        10.0           JC      EXIT                    ;LOOP ADDRESS IF LOOP SPECIFIED
1426                                ;>OPERATOR ERROR NO TU78 UNITS SPECIFIED
1427                                ;>FATAL ERROR - TEST ABORTED
1428  4324  21  37  00      10.0  TST01:  LXI     H,55.                   ;SET UP THE BYTE COUNT
1429  4327  22  01  49      16.0       SHLD    SIZE                    ;
1430  432A  3E  01               7.0  MVI     A,W.RST                 ;SET UP WMC RESTART TO STOP
1431  432C  D3  D3              10.0  OUT     WMCCTL                  ;THE WMC WHILE THE CLOCK IS CHANGING
1432  432E  3E  10               7.0  MVI     A,@020                  ;SET UP NORMAL CLOCKS
1433  4330  D3  F0              10.0  OUT     CLKCTL                  ;
1434  4332  AF                   4.0  XRA     A                       ;CLEAR A
1435  4333  D3  D3              10.0  OUT     WMCCTL                  ;CLEAR RESTART BIT
1436  4335  32  0C  49      13.0       STA     SRPFAL                  ;CLEAR TEMP RAM
1437  4338  32  0D  49      13.0       STA     SRPATH
1438  433B  32  FE  48      13.0       STA     TSTAMT
1439  433E  32  FF  48      13.0       STA     TSTAMP
1440  4341  3E  FF               7.0  MVI     A,OFF                   ;
1441  4343  32  08  49      13.0       STA     PLLFLG                  ;SET PLL FLAG
1442  4346  32  09  49      13.0       STA     WFFLG                   ;SET FLAG SO WRITE FAIL ERRORS
1443                                                                   ;WILL NOT SHOW UP AS LWR ERROR
1444  4349  3E  18               7.0  MVI     A,R.PLOD+R.PLOO         ;GET PLO DISABLE AND PLO BYPASS
1445  434B  32  07  49      13.0       STA     XFRCTL                  ;STORE IN RAM FOR SUBROUTINE
1446  434E  CD  34  47      18.0       CALL    GCRSUB                  ;SETUP FOR LWR
1447  4351  CD  EA  47      18.0       CALL    LWR                     ;DO LWR
1448  4354  D2  70  43      10.0       JNC     TST01A                  ;IF NO ERROR GO TO NEXT TEST
1449  4357  78               4.0       MOV     A,B                     ;GET THE FAILURE CODE
1450  4358  E6  FC               7.0  ANI     $FC                     ;CLEAR OUT JUNK
1451  435A  0F                   4.0  RRC                             ;POSITION FOR OUTPUT
1452  435B  0F                   4.0  RRC
1453  435C                             ROUT    ADATA                   ;STORE IN CAS
```

```
  (1)  435C  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
  (1)  435E  7F               4.0                MOV     A,A             ;RETRY LINK
1454   435F  79               4.0        MOV     A,C                     ;GET THE INTERUPT CODE
1455   4360  E6  3F           7.0        ANI     $3F                     ;CLEAR OUT JUNK
1456   4362                              ROUT    EDATA                   ;WRITE TO CAS
  (1)  4362  D3  95          10.0                OUT     EDATA           ;WRITE AC INTO EDATA
  (1)  4364  7F               4.0                MOV     A,A             ;RETRY LINK
1457   4365                              ERRB    TEST1,T1CON1,3
  (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
  (1)  4365  CD  12  28      18.0                        CALL    ERLPB           ;PROCESS ERROR - DO 2.3
  (1)       0002                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
  (1)  4368  02                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
  (1)  4369  03                                          .BYTE   3               ;PRINT ROUTINE NUMBER
  (1)  436A  CD  15  28      18.0                T1CON1::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
  (1)  436D  DA  00  43      10.0                        JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1458                                ;>LOOP WRITE/READ ERROR WITH PLL'S DISABLED
1459                                ;<ACTUAL = FAILURE CODE
1460                                ;<EXPECTED = INTERUPT CODE
```

```
1462
1463                                        ;
1464                                        ;CHECK TO SEE IF ANY TIE BUS ERRORS OCCURED
1465                                        ;
1466  4370  3E 03          7.0  TST01A: MVI     A,R.STNM+R.STPC          ;STOP READ PATH SO WE CAN
1467                                                                     ;READ REGISTERS 20-27 , 40,47
1468  4372  D3 09         10.0         OUT     RPCTL
1469  4374  AF            4.0         XRA     A                        ;
1470  4375                              ROUT    EDATA                    ;WRITE EXPECTED DATA
 (1)  4375  D3 95         10.0                 OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  4377  7F            4.0                 MOV     A,A              ;RETRY LINK
1471  4378  DB 20         10.0         IN      CHOTIE                   ;READ CH 0 TIE BUS REG
1472  437A  A7            4.0         ANA     A                        ;ANY ERROR BITS
1473  437B  CA 86 43       10.0         JZ      CHOCON                   ;NO , CONTINUE
1474  437E                              ROUT    ADATA                    ;WRITE ACTUAL DATA TO CAS
 (1)  437E  D3 94         10.0                 OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  4380  7F            4.0                 MOV     A,A              ;RETRY LINK
1475  4381                              ERRB    TEST1,CHOCON,3
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4381  CD 12 28       18.0                        CALL    ERLPB       ;PROCESS ERROR - DO 2.3
 (1)        0003                                      MSGN    =       MSGN+1      ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4384  03                                        .BYTE   MSGN        ;MESSAGE NUMBER ID
 (1)  4385  03                                        .BYTE   3           ;PRINT ROUTINE NUMBER
 (1)  4386  CD 15 28       18.0                CHOCON::        CALL    CKLOP       ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4389  DA 00 43       10.0                        JC      TEST1       ;LOOP ADDRESS IF LOOP SPECIFIED
1476                                        ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1477                                        ;<CHANNEL 0 FAILED
1478                                        ;<ACTUAL DATA IS CHANNEL 0 TIE BUS REGISTER
1479  438C  DB 21         10.0         IN      CH1TIE                   ;READ CH 1 TIE BUS REG
1480  438E  A7            4.0         ANA     A                        ;ANY ERROR BITS
1481  438F  CA 9A 43       10.0         JZ      CH1CON                   ;NO , CONTINUE
1482  4392                              ROUT    ADATA                    ;WRITE ACTUAL DATA TO CAS
 (1)  4392  D3 94         10.0                 OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  4394  7F            4.0                 MOV     A,A              ;RETRY LINK
1483  4395                              ERRB    TEST1,CH1CON,3
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4395  CD 12 28       18.0                        CALL    ERLPB       ;PROCESS ERROR - DO 2.3
 (1)        0004                                      MSGN    =       MSGN+1      ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4398  04                                        .BYTE   MSGN        ;MESSAGE NUMBER ID
 (1)  4399  03                                        .BYTE   3           ;PRINT ROUTINE NUMBER
 (1)  439A  CD 15 28       18.0                CH1CON::        CALL    CKLOP       ;CHECK LOOP FUNCTION - DO 2.2
 (1)  439D  DA 00 43       10.0                        JC      TEST1       ;LOOP ADDRESS IF LOOP SPECIFIED
1484                                        ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1485                                        ;<CHANNEL 1 FAILED
1486                                        ;<ACTUAL DATA IS CHANNEL 1 TIE BUS REGISTER
```

```
1488  43A0  DB  22        10.0        IN      CH2TIE                  ;READ CH 2 TIE BUS REG
1489  43A2  A7             4.0        ANA     A                       ;ANY ERROR BITS
1490  43A3  CA  AE  43     10.0        JZ      CH2CON                  ;NO , CONTINUE
1491  43A6                             ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  43A6  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43A8  7F              4.0                 MOV     A,A             ;RETRY LINK
1492  43A9                             ERRB    TEST1,CH2CON,3
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  43A9  CD  12  28     18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0005                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43AC  05                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43AD  03                                .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  43AE  CD  15  28     18.0        CH2CON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43B1  DA  00  43     10.0                JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1493                             ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1494                             ;<CHANNEL 2 FAILED
1495                             ;<ACTUAL DATA IS CHANNEL 2 TIE BUS REGISTER
1496  43B4  DB  23        10.0        IN      CH3TIE                  ;READ CH 3 TIE BUS REG
1497  43B6  A7             4.0        ANA     A                       ;ANY ERROR BITS
1498  43B7  CA  C2  43     10.0        JZ      CH3CON                  ;NO , CONTINUE
1499  43BA                             ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  43BA  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43BC  7F              4.0                 MOV     A,A             ;RETRY LINK
1500  43BD                             ERRB    TEST1,CH3CON,3
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  43BD  CD  12  28     18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0006                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43C0  06                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43C1  03                                .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  43C2  CD  15  28     18.0        CH3CON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43C5  DA  00  43     10.0                JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1501                             ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1502                             ;<CHANNEL 3 FAILED
1503                             ;<ACTUAL DATA IS CHANNEL 3 TIE BUS REGISTER
```

```
1505  43C8  DB  24        10.0        IN      CH4TIE                  ;READ CH 4 TIE BUS REG
1506  43CA  A7             4.0        ANA     A                       ;ANY ERROR BITS
1507  43CB  CA  D6  43     10.0        JZ      CH4CON                  ;NO , CONTINUE
1508  43CE                             ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  43CE  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43D0  7F              4.0                 MOV     A,A             ;RETRY LINK
1509  43D1                             ERRB    TEST1,CH4CON,3
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  43D1  CD  12  28      19.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0007                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43D4  07                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43D5  03                                .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  43D6  CD  15  28      18.0        CH4CON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43D9  DA  00  43      10.0                JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1510                         ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1511                         ;<CHANNEL 4 FAILED
1512                         ;<ACTUAL DATA IS CHANNEL 4 TIE BUS REGISTER
1513  43DC  DB  25        10.0        IN      CH5TIE                  ;READ CH 5 TIE BUS REG
1514  43DE  A7             4.0        ANA     A                       ;ANY ERROR BITS
1515  43DF  CA  EA  43     10.0        JZ      CH5CON                  ;NO , CONTINUE
1516  43E2                             ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  43E2  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43E4  7F              4.0                 MOV     A,A             ;RETRY LINK
1517  43E5                             ERRB    TEST1,CH5CON,3
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  43E5  CD  12  28      18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0008                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43E8  08                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43E9  03                                .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  43EA  CD  15  28      18.0        CH5CON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  43ED  DA  00  43      10.0                JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1518                         ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1519                         ;<CHANNEL 5 FAILED
1520                         ;<ACTUAL DATA IS CHANNEL 5 TIE BUS REGISTER
```

```
1522  43F0  DB  26        10.0          IN      CH6TIE                  ;READ CH 6 TIE BUS REG
1523  43F2  A7             4.0          ANA     A                       ;ANY ERROR BITS
1524  43F3  CA  FE  43     10.0          JZ      CH6CON                  ;NO , CONTINUE
1525  43F6                               ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  43F6  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  43F8  7F              4.0                   MOV     A,A             ;RETRY LINK
1526  43F9                               ERRB    TEST1,CH6CON,3
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  43F9  CD  12  28     18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0009                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  43FC  09                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  43FD  03                                 .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  43FE  CD  15  28     18.0          CH6CON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4401  DA  00  43     10.0                  JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1527                            ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1528                            ;<CHANNEL 6 FAILED
1529                            ;<ACTUAL DATA IS CHANNEL 6 TIE BUS REGISTER
1530  4404  DB  27        10.0          IN      CH7TIE                  ;READ CH 7 TIE BUS REG
1531  4406  A7             4.0          ANA     A                       ;ANY ERROR BITS
1532  4407  CA  12  44     10.0          JZ      CH7CON                  ;NO , CONTINUE
1533  440A                               ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  440A  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  440C  7F              4.0                   MOV     A,A             ;RETRY LINK
1534  440D                               ERRB    TEST1,CH7CON,3
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  440D  CD  12  28     18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        000A                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4410  0A                                 .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4411  03                                 .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  4412  CD  15  28     18.0          CH7CON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4415  DA  00  43     10.0                  JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1535                            ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1536                            ;<CHANNEL 7 FAILED
1537                            ;<ACTUAL DATA IS CHANNEL 7 TIE BUS REGISTER
```

```
1539  4418  DB  28          10.0        IN      CHPTIE                  ;READ PARITY TIE BUS REG
1540  441A  A7               4.0        ANA     A                       ;ANY ERROR BITS
1541  441B  CA  26  44       10.0        JZ      CHPCON                  ;NO , CONTINUE
1542  441E                               ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  441E  D3  94           10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4420  7F               4.0                 MOV     A,A             ;RETRY LINK
1543  4421                               ERRB    TEST1,CHPCON,3
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4421  CD  12  28       18.0                CALL    ER_PB           ;PROCESS ERROR - DO 2.3
 (1)      000B                           MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4424  0B                          .BYTE   MSGN                    ;MESSAGE NUMBER ID
 (1)  4425  03                          .BYTE   3                       ;PRINT ROUTINE NUMBER
 (1)  4426  CD  15  28       18.0    CHPCON::    CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4429  DA  00  43       10.0                JC      TEST1           ;LOOP ADDRESS IF LOOP SPECIFIED
1544                           ;>LOOP WRITE / READ ERROR WITH PLL DISABLED
1545                           ;<PARITY CHANNEL FAILED
1546                           ;<ACTUAL DATA IS PARITY CHANNEL TIE BUS REGISTER
1547  442C                             ENDTST  TEST1
 (1)                           .TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  442C                        REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  442C  CD  06  28       18.0                CALL    REQST
 (2)  442F  00                          .BYTE                           ;DATA PATTERN NUMBER
 (2)  4430  00  00                      .WORD                           ;SYSTEM "" COUNT
 (2)  4432  00  00                      .WORD                           ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4434  00                          .BYTE                           ;DATA COMPARE FLAG IF =1
 (2)  4435  07                          .BYTE   7                       ;REQUEST CODE
 (1)  4436  3A  9A  4F       13.0        LDA     ITERA                   ;GET ITERATION COUNT
 (1)  4439  3D               4.0        DCR     A                       ;DOWNCOUNT
 (1)  443A  32  9A  4F       13.0        STA     ITERA                   ;SAVE COUNT
 (1)  443D  F2  00  43       10.0        JP      TEST1                   ;DO TEST UNTIL TILL = 0
1548
```

```
1550                                .SBTTL  TEST 2 - PHASE LOCKED LOOP TEST PART 2
1551  4440                          ST
 (1)                                ; ***************************************************************************
 (1)                                ;*TEST TITLE
 (1)                                ;*---- -----
1552                                ;*PHASE LOCKED LOOP TEST PART 2
1553  4440                          SD
 (1)                                ; ***************************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
1554                                ;*THIS TEST ISSUES A LOOP WRITE TO READ WITH
1555                                ;*WITH THE PHASE LOCKED LOOP MODULES (M8950)
1556                                ;*ENABLED.  IF ANY ERRORS OCCUR, AN ATTEMPT
1557                                ;*IS MADE TO ISOLATE THE FAILING CHANNEL
1558                                ;*I.E. M8950 AT FAULT
1559  4440                          SP
 (1)                                ; ***************************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
1560                                ;*BGNTST
1561                                ;*  CLEAR ALL TEMP RAM
1562                                ;*  CLEAR XFRCTL SO WE USE PLL'S
1563                                ;*  ISSUE A LOOP WRITE TO READ IN GCR
1564                                ;*  IF ANY ERRORS OCCURED
1565                                ;*  :  THEN
1566                                ;*  :    READ ALL TRACK TIE REGISTERS AND
1567                                ;*  :    PRINT TRACKS AT FAULT
1568                                ;*  :  ELSE
1569                                ;*  :    CONTINUE
1570                                ;*  ENDIF
1571                                ;*ENDTST
1572  4440                          SE
 (1)                                ; ***************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1573                                ;*RPM6 MICRO TEST 02
1574                                ;*RPM6 MICRO ERROR 01
1575                                ;*RPM6-PHASE LOCKED LOOPS TEST PART 2
1576                                ;*M8950 CHANNEL (N)
1577                                ;*LOOP WRITE / READ ERROR WITH PLL ENABLED
1578                                ;*ACTUAL = AAAA
1579                                ;*EXPECTED = EEEE
1580                                ;*BYTE-SCLK COUNT = LLL
1581                                ;*DATA FORMAT = MM
1582                                ;*SKIP COUNT = NN
1583                                ;*ACTUAL DATA IS CHANNEL (N) TIE BUS REGISTER
1584  4440                          S
 (1)                                ; ***************************************************************************
```

```
1586  4440                                  TEST2:  TESTX   @2                      ;INITIALIZE TEST NUMBER 2
 (1)  4440  3E  02              7.0                        MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4442  CD  03  28         18.0                        CALL    TSET            ;SETUP THE TEST
1587                                    ;%RPM6-PHASE LOCKED LOOP TEST PART 2
1588   .                                ;&M8950 CHANNEL (N)
1589  4445  AF                4.0              XRA     A                       ;CLEAR A
1590  4446  32  09  49        13.0              STA     WFFLG                   ;ALLOW WRITE FAIL BIT CHECKING
1591  4449  32  07  49        13.0              STA     XFRCTL                  ;ALLOW PLL TO BE ENABLED
1592  444C  CD  34  47        18.0              CALL    GCRSUB                  ;SET UP FOR GCR LWR
1593  444F  CD  EA  47        18.0              CALL    LWR
1594  4452  D2  11  45        10.0              JNC     ENDT2                   ;IF NO ERROR GO TO NEXT TEST
1595  4455  3E  03              7.0              MVI     A,R.STNM+R.STPC         ;STOP READ PATH SO WE CAN
1596                                                                            ;READ REGISTERS 20-27 , 40-47
1597  4457  D3  09             10.0              OUT     RPCTL
1598  4459  AF                4.0              XRA     A                       ;
1599  445A                                     ROUT    EDATA                   ;WRITE EXPECTED DATA
 (1)  445A  D3  95             10.0                      OUT     EDATA         ;WRITE AC INTO EDATA
 (1)  445C  7F                 4.0                      MOV     A,A            ;RETRY LINK
1600  445D  DB  20             10.0              IN      CHOTIE                  ;READ CH 0 TIE BUS REG
1601  445F  A7                 4.0              ANA     A                       ;ANY ERROR BITS
1602  4460  CA  6B  44         10.0              JZ      CHOCO2                  ;NO , CONTINUE
1603  4463                                     ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  4463  D3  94             10.0                      OUT     ADATA         ;WRITE AC INTO ADATA
 (1)  4465  7F                 4.0                      MOV     A,A            ;RETRY LINK
1604  4466                                     ERRB    TEST2,CHOCO2,3
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4466  CD  12  28         18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000C                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4469  0C                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  446A  03                                          .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  446B  CD  15  28         18.0              CHOCO2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  446E  DA  40  44         10.0                      JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1605                                    ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1606                                    ;<CHANNEL 0 FAILED
1607                                    ;<ACTUAL DATA IS CHANNEL 0 TIE BUS REGISTER
1608  4471  DB  21             10.0              IN      CH1TIE                  ;READ CH 1 TIE BUS REG
1609  4473  A7                 4.0              ANA     A                       ;ANY ERROR BITS
1610  4474  CA  7F  44         10.0              JZ      CH1CO2                  ;NO , CONTINUE
1611  4477                                     ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  4477  D3  94             10.0                      OUT     ADATA         ;WRITE AC INTO ADATA
 (1)  4479  7F                 4.0                      MOV     A,A            ;RETRY LINK
1612  447A                                     ERRB    TEST2,CH1CO2,3
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  447A  CD  12  28         18.0                      CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000D                                         MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  447D  0D                                          .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  447E  03                                          .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  447F  CD  15  28         18.0              CH1CO2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4482  DA  40  44         10.0                      JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1613                                    ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1614                                    ;<CHANNEL 1 FAILED
1615                                    ;<ACTUAL DATA IS CHANNEL 1 TIE BUS REGISTER
```

```
1617  4485  DB  22        10.0        IN      CH2TIE                  ;READ CH 2 TIE BUS REG
1618  4487  A7            4.0         ANA     A                       ;ANY ERROR BITS
1619  4488  CA  93  44    10.0         JZ      CH2CO2                  ;NO , CONTINUE
1620  448B                             ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  448B  D3  94        10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  448D  7F            4.0                  MOV     A,A             ;RETRY LINK
1621  448E                             ERRB    TEST2,CH2CO2,3
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  448E  CD  12  28    18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000E                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4491  0E                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4492  03                                .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  4493  CD  15  28    18.0        CH2CO2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4496  DA  40  44    10.0                 JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1622                             ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1623                             ;<CHANNEL 2 FAILED
1624                             ;<ACTUAL DATA IS CHANNEL 2 TIE BUS REGISTER
1625  4499  DB  23        10.0        IN      CH3TIE                  ;READ CH 3 TIE BUS REG
1626  449B  A7            4.0         ANA     A                       ;ANY ERROR BITS
1627  449C  CA  A7  44    10.0         JZ      CH3CO2                  ;NO , CONTINUE
1628  449F                             ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  449F  D3  94        10.0                 OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44A1  7F            4.0                  MOV     A,A             ;RETRY LINK
1629  44A2                             ERRB    TEST2,CH3CO2,3
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44A2  CD  12  28    18.0                 CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       000F                               MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44A5  0F                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44A6  03                                .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  44A7  CD  15  28    18.0        CH3CO2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44AA  DA  40  44    10.0                 JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1630                             ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1631                             ;<CHANNEL 3 FAILED
1632                             ;<ACTUAL DATA IS CHANNEL 3 TIE BUS REGISTER
```

```
1634  44AD  DB  24          10.0        IN      CH4TIE                  ;READ CH 4 TIE BUS REG
1635  44AF  A7               4.0        ANA     A                       ;ANY ERROR BITS
1636  44B0  CA  BB  44       10.0        JZ      CH4CO2                  ;NO , CONTINUE
1637  44B3                               ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  44B3  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44B5  7F               4.0                MOV     A,A             ;RETRY LINK
1638  44B6                               ERRB    TEST2,CH4CO2,3
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44B6  CD  12  28       18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0010                          MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44B9  10                           .BYTE   MSGN                    ;MESSAGE NUMBER ID
 (1)  44BA  03                           .BYTE   3                       ;PRINT ROUTINE NUMBER
 (1)  44BB  CD  15  28       18.0        CH4CO2::    CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44BE  DA  40  44       10.0                JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1639                                ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1640                                ;<CHANNEL 4 FAILED
1641                                ;<ACTUAL DATA IS CHANNEL 4 TIE BUS REGISTER
1642  44C1  DB  25          10.0        IN      CH5TIE                  ;READ CH 5 TIE BUS REG
1643  44C3  A7               4.0        ANA     A                       ;ANY ERROR BITS
1644  44C4  CA  CF  44       10.0        JZ      CH5CO2                  ;NO , CONTINUE
1645  44C7                               ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  44C7  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44C9  7F               4.0                MOV     A,A             ;RETRY LINK
1646  44CA                               ERRB    TEST2,CH5CO2,3
 (1)                                     ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44CA  CD  12  28       18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0011                          MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44CD  11                           .BYTE   MSGN                    ;MESSAGE NUMBER ID
 (1)  44CE  03                           .BYTE   3                       ;PRINT ROUTINE NUMBER
 (1)  44CF  CD  15  28       18.0        CH5CO2::    CALL    CKLOP               ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44D2  DA  40  44       10.0                JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1647                                ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1648                                ;<CHANNEL 5 FAILED
1649                                ;<ACTUAL DATA IS CHANNEL 5 TIE BUS REGISTER
```

```
1651  44D5  DB  26        10.0        IN      CH6TIE                  ;READ CH 6 TIE BUS REG
1652  44D7  A7            4.0         ANA     A                       ;ANY ERROR BITS
1653  44D8  CA  E3  44     10.0        JZ      CH6CO2                  ;NO , CONTINUE
1654  44DB                            ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  44DB  D3  94        10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44DD  7F            4.0                 MOV     A,A             ;RETRY LINK
1655  44DE                            ERRB    TEST2,CH6CO2,3
 (1)                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44DE  CD  12  28     18.0                CALL    ERLPB   ;'     ;PROCESS ERROR - DO 2.3
 (1)        0012                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44E1  12                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44E2  03                     .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  44E3  CD  15  28     18.0    CH6CO2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44E6  DA  40  44     10.0                JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1656                        ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1657                        ;<CHANNEL 6 FAILED
1658                        ;<ACTUAL DATA IS CHANNEL 6 TIE BUS REGISTER
1659  44E9  DB  27        10.0        IN      CH7TIE                  ;READ CH 7 TIE BUS REG
1660  44EB  A7            4.0         ANA     A                       ;ANY ERROR BITS
1661  44EC  CA  F7  44     10.0        JZ      CH7CO2                  ;NO , CONTINUE
1662  44EF                            ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  44EF  D3  94        10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44F1  7F            4.0                 MOV     A,A             ;RETRY LINK
1663  44F2                            ERRB    TEST2,CH7CO2,3
 (1)                                ;FLAG ERROR - WITH ACTUAL DA A "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  44F2  CD  12  28     18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0013                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44F5  13                     .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44F6  03                     .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  44F7  CD  15  28     18.0    CH7CO2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  44FA  DA  40  44     10.0                JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1664                        ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1665                        ;<CHANNEL 7 FAILED
1666                        ;<ACTUAL DATA IS CHANNEL 7 TIE BUS REGISTER
```

```
1668  44FD  DB  28        10.0        IN      CHPTIE                  ;READ PARITY TIE BUS REG
1669  44FF  A7             4.0        ANA     A                       ;ANY ERROR BITS
1670  4500  CA  0B  45     10.0        JZ      CHPCO2                  ;NO , CONTINUE
1671  4503                             ROUT    ADATA                   ;WRITE ACTUAL DATA TO CAS
 (1)  4503  D3  94          10.0                OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4505  7F               4.0                MOV     A,A             ;RETRY LINK
1672  4506                             ERRB    TEST2,CHPCO2,3
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4506  CD  12  28     18.0                CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0014                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4509  14                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  450A  03                                .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  450B  CD  15  28     18.0        CHPCO2::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  450E  DA  40  44     10.0                JC      TEST2           ;LOOP ADDRESS IF LOOP SPECIFIED
1673                            ;>LOOP WRITE / READ ERROR WITH PLL ENABLED
1674                            ;<PARITY CHANNEL FAILED
1675                            ;<ACTUAL DATA IS PARITY CHANNEL TIE BUS REGISTER
1676  4511                      ENDT2:  ENDTST  TEST2
 (1)                            ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4511                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4511  CD  06  28     18.0                CALL    REQST
 (2)  4514  00                                .BYTE                   ;DATA PATTERN NUMBER
 (2)  4515  00  00                            .WORD                   ;SYSTEM "" COUNT
 (2)  4517  00  00                            .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4519  00                                .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  451A  07                                .BYTE   7               ;REQUEST CODE
 (1)  451B  3A  9A  4F     13.0                LDA     ITERA           ;GET ITERATION COUNT
 (1)  451E  3D               4.0                DCR     A               ;DOWNCOUNT
 (1)  451F  32  9A  4F     13.0                STA     ITERA           ;SAVE COUNT
 (1)  4522  F2  40  44     10.0                JP      TEST2           ;DO TEST UNTIL TILL = 0
```

```
1678                            .SBTTL  TEST 3 - WRITE FAIL LINES CLEAR TEST
1679  4525                      ST
 (1)                            ;*********************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1680                            ;*WRITE FAIL LINES CLEAR TEST
1681  4525                      SD
 (1)                            ;*********************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1682                            ;*THIS TEST PERFORMS A GCR LOOP WRITE TO READ
1683                            ;*AND THEN CHECKS TO SEE THAT ALL WRITF FAIL LINES
1684                            ;*ARE CLEAR
1685  4525                      SP
 (1)                            ;*********************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1686                            ;*BGNTST
1687                            ;*  GET USER SELECTED TU PORT
1688                            ;*  SET THE RECORD SIZE TO 55.
1689                            ;*  BGNDO
1690                            ;*  :  SET THE WRITE PATH CLOCK TO NORMAL
1691                            ;*  :  SET THE READ PATH CLOCK TO NORMAL
1692                            ;*  :  CLEAR ALL AMTIE LINES
1693                            ;*  :  CLEAR TEMP REGISTER SAVE LOCATIONS
1694                            ;*  :  CALL GCRSUB
1695                            ;*  :  IF ANY WRITE FAIL LINES ARE SET
1696                            ;*  :  :  THEN
1697                            ;*  :  :  :  PRINT ERROR
1698                            ;*  :  :  ELSE
1699                            ;*  :  :  :  CONTINUE
1700                            ;*  :  ENDIF
1701                            ;*  ENDDO
1702                            ;*ENDTST
1703  4525                      SE
 (1)                            ;*********************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1704                            ;*RPM6 MICRO TEST 03
1705                            ;*RPM6 MICRO ERROR 01
1706                            ;*RPM6-WRITE FAIL LINES CLEAR TEST
1707                            ;*M8950'S , M8953
1708                            ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1709                            ;*FATAL ERROR - TEST ABORTED
1710                            ;*
1711                            ;*RPM6 MICRO TEST 03
1712                            ;*RPM6 MICRO ERROR 02
1713                            ;*RPM6-WRITE FAIL LINES CLEAR TEST
1714                            ;*M8950'S,M8953
1715                            ;*WRITE FAIL BITS NOT CLEAR
1716                            ;*ACTUAL = AAAA
1717                            ;*EXPECTED = EEEE
1718                            ;*ACTUAL DATA IS WRITE FAIL REGISTER
1719                            ;*
```

```
1720                                    ;*RPM6 MICRO TEST 03
1721                                    ;*RPM6 MICRO ERROR 03
1722                                    ;*RPM6-WRITE FAIL LINES CLEAR TEST
1723                                    ;*M8950 , M8953
1724                                    ;*WRITE FAIL PARITY BIT NOT CLEAR
1725                                    ;*ACTUAL = AAAA
1726                                    ;*EXPECTED = EEEE
1727                                    ;*
1728                                    ;*RPM6 MICRO TEST 03
1729                                    ;*RPM6 MICRO ERROR 04
1730                                    ;*RPM6-WRITE FAIL LINES CLEAR TEST
1731                                    ;*M8950'S , M8953
1732                                    ;*LOOP WRITE/READ ERROR
1733                                    ;*ACTUAL = AAAA
1734                                    ;*EXPECTED = EEEE
1735                                    ;*BYTE-SCLK COUNT = LLL
1736                                    ;*DATA FORMAT = MM
1737                                    ;*SKIP COUNT = NN
1738                                    ;*ACTUAL DATA IS THE FAILURE CODE
1739                                    ;*EXPECTED DATA IS THE INTERUPT CODE
1740  4525                              S
 (1)                                    ; ***************************************************************************
1741
1742  4525                              TEST3:  TESTX   @3                      ;INITIALIZE THE TEST
 (1)  4525  3E  03          7.0                  MVI     A,@3                    ;DEFINE THE TEST NUMBER
 (1)  4527  CD  03  28     18.0                  CALL    TSET                    ;SETUP THE TEST
1743                                    ;%RPM6-WRITE FAIL LINES CLEAR TEST
1744                                    ;&M8950'S , M8953
1745  452A  AF                4.0               XRA     A                       ;CLEAR A
1746  452B  32  08  49       13.0               STA     PLLFLG                  ;CLEAR PLL TEST FLAG
1747  452E  32  07  49       13.0               STA     XFRCTL                  ;STORE PLL CONTROL BYTE
1748  4531  32  0C  49       13.0               STA     SRPFAL                  ;CLEAR I/O SAVE LOCATIONS
1749  4534  32  0D  49       13.0               STA     SRPATH                  ;
1750  4537  32  FE  48       13.0               STA     TSTAMT                  ;CLEAR AMTIE CONTROL BYTE
1751  453A  32  FF  48       13.0               STA     TSTAMP                  ;CLEAR AMTIE PARITY CONTROL BYTE
1752  453D  3E  FF             7.0               MVI     A,0FF                   ;SET WRITE FAIL TESTING FLAG
1753  453F  32  09  49       13.0               STA     WFFLG                   ;TO PREVENT LWR FROM TESTING WRITE FAIL
1754  4542  CD  34  47       18.0               CALL    GCRSUB                  ;CALL SUBROUTINE LWRSUB
```

```
1756  4545  21  0C  49      10.0          LXI     H,SRPFAL                ;GET RPFAIL CONTENTS
1757  4548  7E                 7.0          MOV     A,M                     ;GET WRITE FAIL BITS
1758  4549  E6  FF               7.0          ANI     0FF                     ;ANY BITS SET ?
1759  454B  CA  5A  45         10.0          JZ      T3ACON                  ;NO , CONTINUE
1760  454E                                   ROUT    ADATA                   ;SAVE ACTUAL DATA
 (1)  454E  D3  94              10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4550  7F                  4.0                  MOV     A,A             ;RETRY LINK
1761  4551  AF                 4.0          XRA     A                       ;CLEAR A
1762  4552                                   ROUT    EDATA                   ;STORE EXPECTED DATA
 (1)  4552  D3  95              10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4554  7F                  4.0                  MOV     A,A             ;RETRY LINK
1763  4555                                   ERRB    TEST3,T3ACON
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4555  CD  12  28         18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0015                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4558  15                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4559  00                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  455A  CD  15  28         18.0          T3ACON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  455D  DA  25  45         10.0                  JC      TEST3           ;LOOP ADDRESS IF LOOP SPECIFIED
1764                                   ;>WRITE FAIL BITS NOT CLEAR
1765                                   ;<ACTUAL DATA IS WRITE FAIL REGISTER
1766  4560  21  0D  49      10.0  TST03B: LXI     H,SRPATH                ;GET WRITE FAIL PARITY BIT
1767  4563  7E                 7.0          MOV     A,M                     ;
1768  4564  E6  01               7.0          ANI     01                      ;IS IT SET
1769  4566  CA  75  45         10.0          JZ      T3AC01                  ;NO , GOOD JUMP
1770  4569                                   ROUT    ADATA                   ;STORE ACTUAL DATA
 (1)  4569  D3  94              10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  456B  7F                  4.0                  MOV     A,A             ;RETRY LINK
1771  456C  AF                 4.0          XRA     A                       ;MAKE EXPECTED BYTE
1772  456D                                   ROUT    EDATA                   ;STORE IN CAS
 (1)  456D  D3  95              10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  456F  7F                  4.0                  MOV     A,A             ;RETRY LINK
1773  4570                                   ERRB    TEST3,T3AC01
 (1)                                       ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4570  CD  12  28         18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0016                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4573  16                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4574  00                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4575  CD  15  28         18.0          T3AC01::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4578  DA  25  45         10.0                  JC      TEST3           ;LOOP ADDRESS IF LOOP SPECIFIED
1774                                   ;>WRITE FAIL PARITY BIT NOT CLEAR
1775  457B                             ENDTST  TEST3
 (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  457B                             REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  457B  CD  06  28         18.0                  CALL    REQST
 (2)  457E  00                                       .BYTE                   ;DATA PATTERN NUMBER
 (2)  457F  00  00                                   .WORD                   ;SYSTEM "" COUNT
 (2)  4581  00  00                                   .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4583  00                                       .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4584  07                                       .BYTE   7               ;REQUEST CODE
 (1)  4585  3A  9A  4F         13.0          LDA     ITERA                   ;GET ITERATION COUNT
 (1)  4588  3D                  4.0          DCR     A                       ;DOWNCOUNT
 (1)  4589  32  9A  4F         13.0          STA     ITERA                   ;SAVE COUNT
 (1)  458C  F2  25  45         10.0          JP      TEST3                   ;DO TEST UNTIL TILL = 0
```

```
1777                                .SBTTL  TEST 4 - WRITE FAIL LINES SET TEST
1778  458F                         ST
 (1)                                ; ************************************************************************
 (1)                                ;*TEST TITLE
 (1)                                ;*---- -----
1779                                ;*WRITE FAIL LINES SET TEST
1780  458F                         SD
 (1)                                ; ************************************************************************
 (1)                                ;*DESCRIPTION
 (1)                                ;*-----------
1781                                ;*THIS TEST PERFORMS A GCR LOOP WRITE TO READ
1782                                ;*WHILE SETTING AN AMTIE BIT FOR EACH CHANNEL
1783                                ;*THEN CHECKING THAT THE APPROPRIATE WRITE FAIL
1784                                ;*BIT IS SET
1785  458F                         SP
 (1)                                ; ************************************************************************
 (1)                                ;*PROCEDURE
 (1)                                ;*---------
1786                                ;*BGNTST
1787                                ;*  GET USER SELECTED TU PORT
1788                                ;*  BGNDO
1789                                ;*  :  BGNDO UNTIL ALL CHANNELS HAVE BEEN TESTED
1790                                ;*  :  :  SET AMTIE BIT FOR CHANNEL (N)
1791                                ;*  :  :  CALL SUBROUTINE GCRSUB
1792                                ;*  :  :  IF THE WRITE FAIL BIT FOR CHANNEL (N) IS SET
1793                                ;*  :  :  :  THEN
1794                                ;*  :  :  :  :  CONTINUE
1795                                ;*  :  :  :  ELSE
1796                                ;*  :  :  :  :  PRINT ERROR
1797                                ;*  :  :  :  :  CONTINUE
1798                                ;*  :  :  ENDIF
1799                                ;*  :  ENDDO
1800                                ;*  ENDDO
1801                                ;*ENDTST
1802  458F                         SE
 (1)                                ; ************************************************************************
 (1)                                ;*ERRORS
 (1)                                ;*------
1803                                ;*RPM6 MICRO TEST 04
1804                                ;*RPM6 MICRO ERROR 01
1805                                ;*RPM6-WRITE FAIL LINES SET TEST
1806                                ;*M8950 CHANNEL (N), M8953
1807                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1808                                ;*FATAL ERROR - TEST ABORTED
1809                                ;*
1810                                ;*RPM6 MICRO TEST 04
1811                                ;*RPM6 MICRO ERROR 02
1812                                ;*RPM6-WRITE FAIL LINES SET TEST
1813                                ;*M8950 CHANNEL (N),M8953
1814                                ;*WRITE FAIL LINE FOR CHANNEL (N) NOT SET
1815                                ;*ACTUAL = AAAA
1816                                ;*EXPECTED = EEEE
1817                                ;*
1818                                ;*RPM6 MICRO TEST 04
```

```
1819                                    ;*RPM6 MICRO ERROR 03
1820                                    ;*RPM6-WRITE FAIL LINES SET TEST
1821                                    ;*M8950 CHANNEL (N),M8953
1822                                    ;*LOOP WRITE/READ ERROR
1823                                    ;*ACTUAL = AAAA
1824                                    ;*EXPECTED = EEEE
1825                                    ;*BYTE-SCLK COUNT = LLL
1826                                    ;*DATA FORMAT = MM
1827                                    ;*SKIP COUNT = NN
1828                                    ;*ACTUAL DATA IS THE FAILURE CODE
1829                                    ;*EXPECTED DATA IS THE INTERUPT CODE
1830  458F                              $
 (1)                                    ; ***************************************************************************
1831  458F                              TEST4:  TESTX   @4                      ;INITIALIZE TEST
 (1)  458F  3E  04            7.0                   MVI     A,@4              ;DEFINE THE TEST NUMBER
 (1)  4591  CD  03  28       18.0                   CALL    TSET              ;SETUP THE TEST
1832                                    ;%RPM6-WRITE FAIL LINES SET TEST
1833                                    ;&M8950 CHANNEL (N),M8953
1834  4594  AF                 4.0     TST04A: XRA     A                       ;CLEAR A
1835  4595  32  0C  49          13.0             STA     SRPFAL                  ;CLEAR I/O SAVE LOCATIONS
1836  4598  32  0D  49          13.0             STA     SRPATH                  ;
1837  459B  32  FF  48          13.0             STA     TSTAMP                  ;CLEAR PARITY AMTIE LINE
1838  459E  3E  01               7.0             MVI     A,@001                  ;GET BIT FOR AMTIE CHANNEL 0
1839  45A0  32  FE  48          13.0             STA     TSTAMT                  ;SET AMTIE CHANNEL 0
1840  45A3  CD  34  47          18.0             CALL    GCRSUB                  ;DO LOOP WRITE / READ
1841  45A6  21  0C  49          10.0             LXI     H,SRPFAL                ;GET WRITE FAIL LINES
1842  45A9  7E                   7.0             MOV     A,M                     ;
1843  45AA  E6  01               7.0             ANI     @001                    ;IS THE WRITE FAIL BIT SET
1844                                                                             ;FOR CHANNEL 0
1845  45AC  C2  BC  45          10.0             JNZ     T4ACON                  ;YES , GOOD CONTINUE
1846  45AF                             ROUT    ADATA                           ;STORE ACTUAL DATA IN CAS
 (1)  45AF  D3  94            10.0                     OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  45B1  7F                 4.0                     MOV     A,A             ;RETRY LINK
1847  45B2  3E  01               7.0     MVI     A,@001                          ;
1848  45B4                             ROUT    EDATA                           ;STORE EXPECTED DATA
 (1)  45B4  D3  95            10.0                     OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  45B6  7F                 4.0                     MOV     A,A             ;RETRY LINK
1849  45B7                             ERRB    TST04A,T4ACON
 (1)                                 ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  45B7  CD  12  28       18.0                     CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0017                           MSGN    =       MSGN+1                  ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45BA  17                             .BYTE   MSGN                    ;MESSAGE NUMBER ID
 (1)  45BB  00                             .BYTE                           ;PRINT ROUTINE NUMBER
 (1)  45BC  CD  15  28       18.0     T4ACON::        CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45BF  DA  94  45       10.0                     JC      TST04A          ;LOOP ADDRESS IF LOOP SPECIFIED
1850                                ;>WRITE FAIL LINE FOR CHANNEL 0 NOT SET
```

```
1852  45C2  AF                    4.0  TST04B: XRA     A                        ;CLEAR A
1853  45C3  32  0C  49           13.0          STA     SRPFAL                   ;CLEAR I/O SAVE LOCATIONS
1854  45C6  32  0D  49           13.0          STA     SRPATH                   ;
1855  45C9  3E  02                7.0          MVI     A,@002                   ;
1856  45CB  32  FE  48           13.0          STA     TSTAMT                   ;SET AMTIE CHANNEL 1
1857  45CE  CD  34  47           18.0          CALL    GCRSUB                   ;DO LOOP WRITE / READ
1858  45D1  21  0C  49           10.0          LXI     H,SRPFAL                 ;GET WRITE FAIL LINES
1859  45D4  7E                    7.0          MOV     A,M                      ;
1860  45D5  E6  02                7.0          ANI     @002                     ;IS THE WRITE FAIL BIT SET
1861                                                                            ;FOR CHANNEL 1
1862  45D7  C2  E7  45           10.0          JNZ     T4BCON                   ;YES , GOOD CONTINUE
1863  45DA                                     ROUT    ADATA                    ;STORE ACTUAL DATA IN CAS
 (1)  45DA  D3  94               10.0                  OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  45DC  7F                    4.0                  MOV     A,A              ;RETRY LINK
1864  45DD  3E  02                7.0          MVI     A,@002                   ;
1865  45DF                                     ROUT    EDATA                    ;STORE EXPECTED DATA
 (1)  45DF  D3  95               10.0                  OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  45E1  7F                    4.0                  MOV     A,A              ;RETRY LINK
1866  45E2                                     ERRB    TST04B,T4BCON
 (1)                                         ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  45E2  CD  12  28           18.0                  CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)        0018                                       MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  45E5    18                                       .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  45E6    00                                       .BYTE                    ;PRINT ROUTINE NUMBER
 (1)  45E7  CD  15  28           18.0          T4BCON::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  45EA  DA  C2  45           10.0                  JC      TST04B           ;LOOP ADDRESS IF LOOP SPECIFIED
1867                                     ;>WRITE FAIL LINE FOR CHANNEL 1 NOT SET
1868  45ED  AF                    4.0  TST04C: XRA     A                        ;CLEAR A
1869  45EE  32  0C  49           13.0          STA     SRPFAL                   ;CLEAR I/O SAVE LOCATIONS
1870  45F1  32  0D  49           13.0          STA     SRPATH                   ;
1871  45F4  3E  04                7.0          MVI     A,@004                   ;
1872  45F6  32  FE  48           13.0          STA     TSTAMT                   ;SET AMTIE CHANNEL 2
1873  45F9  CD  34  47           18.0          CALL    GCRSUB                   ;DO LOOP WRITE / READ
1874  45FC  21  0C  49           10.0          LXI     H,SRPFAL                 ;GET WRITE FAIL LINES
1875  45FF  7E                    7.0          MOV     A,M                      ;
1876  4600  E6  04                7.0          ANI     @004                     ;IS THE WRITE FAIL BIT SET
1877                                                                            ;FOR CHANNEL 2
1878  4602  C2  12  46           10.0          JNZ     T4CCON                   ;YES , GOOD CONTINUE
1879  4605                                     ROUT    ADATA                    ;STORE ACTUAL DATA IN CAS
 (1)  4605  D3  94               10.0                  OUT     ADATA            ;WRITE AC INTO ADATA
 (1)  4607  7F                    4.0                  MOV     A,A              ;RETRY LINK
1880  4608  3E  04                7.0          MVI     A,@004                   ;
1881  460A                                     ROUT    EDATA                    ;STORE EXPECTED DATA
 (1)  460A  D3  95               10.0                  OUT     EDATA            ;WRITE AC INTO EDATA
 (1)  460C  7F                    4.0                  MOV     A,A              ;RETRY LINK
1882  460D                                     ERRB    TST04C,T4CCON
 (1)                                         ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  460D  CD  12  28           18.0                  CALL    ERLPB            ;PROCESS ERROR - DO 2.3
 (1)        0019                                       MSGN    =       MSGN+1           ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4610    19                                       .BYTE   MSGN             ;MESSAGE NUMBER ID
 (1)  4611    00                                       .BYTE                    ;PRINT ROUTINE NUMBER
 (1)  4612  CD  15  28           18.0          T4CCON::        CALL    CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4615  DA  ED  45           10.0                  JC      TST04C           ;LOOP ADDRESS IF LOOP SPECIFIED
1883                                     ;>WRITE FAIL LINE FOR CHANNEL 2 NOT SET
```

```
1885  4618  AF                  4.0  TST04D: XRA   A                    ;CLEAR A
1886  4619  32  0C  49          13.0         STA   SRPFAL                ;CLEAR I/O SAVE LOCATIONS
1887  461C  32  0D  49          13.0         STA   SRPATH                ;
1888  461F  3E  08               7.0         MVI   A,@010               ;
1889  4621  32  FE  48          13.0         STA   TSTAMT               ;SET AMTIE CHANNEL 3
1890  4624  CD  34  47          18.0         CALL  GCRSUB               ;DO LOOP WRITE / READ
1891  4627  21  0C  49          10.0         LXI   H,SRPFAL             ;GET WRITE FAIL LINES
1892  462A  7E                   7.0         MOV   A,M                  ;
1893  462B  E6  08               7.0         ANI   @010                 ;IS THE WRITE FAIL BIT SET
1894                                                                    ;FOR CHANNEL 3
1895  462D  C2  3D  46          10.0         JNZ   T4DCON               ;YES , GOOD CONTINUE
1896  4630                                   ROUT  ADATA                ;STORE ACTUAL DATA IN CAS
 (1)  4630  D3  94               10.0               OUT   ADATA         ;WRITE AC INTO ADATA
 (1)  4632  7F                    4.0               MOV   A,A           ;RETRY LINK
1897  4633  3E  08               7.0         MVI   A,@010               ;
1898  4635                                   ROUT  EDATA                ;STORE EXPECTED DATA
 (1)  4635  D3  95               10.0               OUT   EDATA         ;WRITE AC INTO EDATA
 (1)  4637  7F                    4.0               MOV   A,A           ;RETRY LINK
1899  4638                                   ERRB  TST04D,T4DCON
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4638  CD  12  28          18.0                     CALL  ERLPB       ;PROCESS ERROR - DO 2.3
 (1)      001A                                         MSGN  =     MSGN+1       ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  463B  1A                                         .BYTE MSGN         ;MESSAGE NUMBER ID
 (1)  463C  00                                         .BYTE              ;PRINT ROUTINE NUMBER
 (1)  463D  CD  15  28          18.0               T4DCON::  CALL  CKLOP       ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4640  DA  18  46          10.0                     JC    TST04D       ;LOOP ADDRESS IF LOOP SPECIFIED
1900                           ;>WRITE FAIL LINE FOR CHANNEL 3 NOT SET
1901  4643  AF                  4.0  TST04E: XRA   A                    ;CLEAR A
1902  4644  32  0C  49          13.0         STA   SRPFAL                ;CLEAR I/O SAVE LOCATIONS
1903  4647  32  0D  49          13.0         STA   SRPATH                ;
1904  464A  3E  10               7.0         MVI   A,@020               ;
1905  464C  32  FE  48          13.0         STA   TSTAMT               ;SET AMTIE CHANNEL 4
1906  464F  CD  34  47          18.0         CALL  GCRSUB               ;DO LOOP WRITE / READ
1907  4652  21  0C  49          10.0         LXI   H,SRPFAL             ;GET WRITE FAIL LINES
1908  4655  7E                   7.0         MOV   A,M                  ;
1909  4656  E6  10               7.0         ANI   @020                 ;IS THE WRITE FAIL BIT SET
1910                                                                    ;FOR CHANNEL 4
1911  4658  C2  68  46          10.0         JNZ   T4ECON               ;YES , GOOD CONTINUE
1912  465B                                   ROUT  ADATA                ;STORE ACTUAL DATA IN CAS
 (1)  465B  D3  94               10.0               OUT   ADATA         ;WRITE AC INTO ADATA
 (1)  465D  7F                    4.0               MOV   A,A           ;RETRY LINK
1913  465E  3E  10               7.0         MVI   A,@020               ;
1914  4660                                   ROUT  EDATA                ;STORE EXPECTED DATA
 (1)  4660  D3  95               10.0               OUT   EDATA         ;WRITE AC INTO EDATA
 (1)  4662  7F                    4.0               MOV   A,A           ;RETRY LINK
1915  4663                                   ERRB  TST04E,T4ECON
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4663  CD  12  28          18.0                     CALL  ERLPB       ;PROCESS ERROR - DO 2.3
 (1)      001B                                         MSGN  =     MSGN+1       ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4666  1B                                         .BYTE MSGN         ;MESSAGE NUMBER ID
 (1)  4667  00                                         .BYTE              ;PRINT ROUTINE NUMBER
 (1)  4668  CD  15  28          18.0               T4ECON::  CALL  CKLOP       ;CHECK LOOP FUNCTION - DO 2.2
 (1)  466B  DA  43  46          10.0                     JC    TST04E       ;LOOP ADDRESS IF LOOP SPECIFIED
1916                           ;>WRITE FAIL LINE FOR CHANNEL 4 NOT SET
```

```
1918  466E  AF          4.0  TST04I: XRA   A                    ;CLEAR A
1919  466F  32 0C 49     13.0        STA   SRPFAL                ;CLEAR I/O SAVE LOCATIONS
1920  4672  32 0D 49     13.0        STA   SRPATH                ;
1921  4675  3E 20        7.0         MVI   A,@040               ;
1922  4677  32 FE 48     13.0        STA   TSTAMT               ;SET AMTIE CHANNEL 5
1923  467A  CD 34 47     18.0        CALL  GCRSUB               ;DO LOOP WRITE / READ
1924  467D  21 0C 49     10.0        LXI   H,SRPFAL             ;GET WRITE FAIL LINES
1925  4680  7E           7.0         MOV   A,M                  ;
1926  4681  E6 20        7.0         ANI   @040                 ;IS THE WRITE FAIL BIT SET
1927                                                           ;FOR CHANNEL 5
1928  4683  C2 93 46     10.0        JNZ   T4FCON               ;YES , GOOD CONTINUE
1929  4686                          ROUT  ADATA                ;STORE ACTUAL DATA IN CAS
 (1)  4686  D3 94         10.0              OUT   ADATA          ;WRITE AC INTO ADATA
 (1)  4688  7F           4.0               MOV   A,A            ;RETRY LINK
1930  4689  3E 20        7.0         MVI   A,@040               ;
1931  468B                          ROUT  EDATA                ;STORE EXPECTED DATA
 (1)  468B  D3 95         10.0              OUT   EDATA          ;WRITE AC INTO EDATA
 (1)  468D  7F           4.0               MOV   A,A            ;RETRY LINK
1932  468E                          ERRB  TST04F,T4FCON
 (1)                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  468E  CD 12 28     18.0                   CALL  ERLPB       ;PROCESS ERROR - DO 2.3
 (1)       001C                                MSGN  =     MSGN+1      ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4691  1C                                 .BYTE MSGN        ;MESSAGE NUMBER ID
 (1)  4692  00                                 .BYTE             ;PRINT ROUTINE NUMBER
 (1)  4693  CD 15 28     18.0          T4FCON::       CALL  CKLOP      ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4696  DA 6E 46     10.0                   JC    TST04F      ;LOOP ADDRESS IF LOOP SPECIFIED
1933                          ;>WRITE FAIL LINE FOR CHANNEL 5 NOT SET
1934  4699  AF          4.0  TST04G: XRA   A                    ;CLEAR A
1935  469A  32 0C 49     13.0        STA   SRPFAL               ;CLEAR I/O SAVE LOCATIONS
1936  469D  32 0D 49     13.0        STA   SRPATH               ;
1937  46A0  3E 40        7.0         MVI   A,@100               ;
1938  46A2  32 FE 48     13.0        STA   TSTAMT               ;SET AMTIE CHANNEL 6
1939  46A5  CD 34 47     18.0        CALL  GCRSUB               ;DO LOOP WRITE / READ
1940  46A8  21 0C 49     10.0        LXI   H,SRPFAL             ;GET WRITE FAIL LINES
1941  46AB  7E           7.0         MOV   A,M                  ;
1942  46AC  E6 40        7.0         ANI   @100                 ;IS THE WRITE FAIL BIT SET
1943                                                           ;FOR CHANNEL 6
1944  46AE  C2 BE 46     10.0        JNZ   T4GCON               ;YES , GOOD CONTINUE
1945  46B1                          ROUT  ADATA                ;STORE ACTUAL DATA IN CAS
 (1)  46B1  D3 94         10.0              OUT   ADATA          ;WRITE AC INTO ADATA
 (1)  46B3  7F           4.0               MOV   A,A            ;RETRY LINK
1946  46B4  3E 40        7.0         MVI   A,@100               ;
1947  46B6                          ROUT  EDATA                ;STORE EXPECTED DATA
 (1)  46B6  D3 95         10.0              OUT   EDATA          ;WRITE AC INTO EDATA
 (1)  46B8  7F           4.0               MOV   A,A            ;RETRY LINK
1948  46B9                          ERRB  TST04G,T4GCON
 (1)                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  46B9  CD 12 28     18.0                   CALL  ERLPB       ;PROCESS ERROR - DO 2.3
 (1)       001D                                MSGN  =     MSGN+1      ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46BC  1D                                 .BYTE MSGN        ;MESSAGE NUMBER ID
 (1)  46BD  00                                 .BYTE             ;PRINT ROUTINE NUMBER
 (1)  46BE  CD 15 28     18.0          T4GCON::       CALL  CKLOP      ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46C1  DA 99 46     10.0                   JC    TST04G      ;LOOP ADDRESS IF LOOP SPFCIFIED
1949                          ;>WRITE FAIL LINE FOR CHANNEL 6 NOT SET
```

```
1951  46C4  AF          4.0  TST04H: XRA   A                    ;CLEAR A
1952  46C5  32 0C 49     13.0        STA   SRPFAL                ;CLEAR I/O SAVE LOCATIONS
1953  46C8  32 0D 49     13.0        STA   SRPATH                ;
1954  46CB  3E 80         7.0         MVI   A,@200               ;
1955  46CD  32 FE 48     13.0        STA   TSTAMT               ;SET AMTIE CHANNEL 7
1956  46D0  CD 34 47     18.0        CALL  GCRSUB               ;DO LOOP WRITE / READ
1957  46D3  21 0C 49     10.0        LXI   H,SRPFAL             ;GET WRITE FAIL LINES
1958  46D6  7E            7.0        MOV   A,M                  ;
1959  46D7  E6 80         7.0        ANI   @200                 ;IS THE WRITE FAIL BIT SET
1960                                                            ;FOR CHANNEL 7
1961  46D9  C2 E9 46     10.0        JNZ   T4HCON               ;YES , GOOD CONTINUE
1962  46DC                           ROUT  ADATA                ;STORE ACTUAL DATA IN CAS
 (1)  46DC  D3 94         10.0              OUT   ADATA         ;WRITE AC INTO ADATA
 (1)  46DE  7F            4.0               MOV   A,A           ;RETRY LINK
1963  46DF  3E 80         7.0        MVI   A,@200               ;
1964  46E1                           ROUT  EDATA                ;STORE EXPECTED DATA
 (1)  46E1  D3 95         10.0              OUT   EDATA         ;WRITE AC INTO EDATA
 (1)  46E3  7F            4.0               MOV   A,A           ;RETRY LINK
1965  46E4                           ERRB  TST04H,T4HCON
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA ''ADATA'' AND EXPECTED DATA ''EDATA'' VALID
 (1)  46E4  CD 12 28     18.0                   CALL  ERLPB       ;PROCESS ERROR - DO 2.3
 (1)        001E                                MSGN  =     MSGN+1      ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  46E7  1E                                  .BYTE MSGN        ;MESSAGE NUMBER ID
 (1)  46E8  00                                  .BYTE             ;PRINT ROUTINE NUMBER
 (1)  46E9  CD 15 28     18.0          T4HCON::        CALL  CKLOP       ;CHECK LOOP FUNCTION - DO 2.2
 (1)  46EC  DA C4 46     10.0                  JC    TST04H      ;LOOP ADDRESS IF LOOP SPECIFIED
1966                           ;>WRITE FAIL LINE FOR CHANNEL 7 NOT SET
```

```
1968  46EF  AF                4.0  TST04I: XRA     A                       ;CLEAR A
1969  46F0  32  0C  49        13.0          STA     SRPFAL                  ;CLEAR I/O SAVE LOCATIONS
1970  46F3  32  0D  49        13.0          STA     SRPATH                  ;
1971  46F6  32  FE  48        13.0          STA     TSTAMT                  ;CLEAR AMTIES
1972  46F9  3E  01             7.0          MVI     A,@001                  ;
1973  46FB  32  FF  48        13.0          STA     TSTAMP                  ;SET AMTIE PARITY LINE
1974  46FE  CD  34  47        18.0          CALL    GCRSUB                  ;DO LOOP WRITE / READ
1975  4701  21  0D  49        10.0          LXI     H,SRPATH                ;GET PARITY WRITE FAIL LINE
1976  4704  7E                7.0          MOV     A,M                     ;
1977  4705  E6  01             7.0          ANI     @001                    ;IS THE WRITE FAIL BIT SET
1978                                                                       ;FOR THE PARITY LINE
1979  4707  C2  17  47        10.0          JNZ     T4ICON                  ;YES , GOOD CONTINUE
1980  470A                                  ROUT    ADATA                   ;STORE ACTUAL DATA IN CAS
 (1)  470A  D3  94             10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  470C  7F                 4.0                  MOV     A,A             ;RETRY LINK
1981  470D  3E  01             7.0          MVI     A,@001                  ;
1982  470F                                  ROUT    EDATA                   ;STORE EXPECTED DATA
 (1)  470F  D3  95             10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  4711  7F                 4.0                  MOV     A,A             ;RETRY LINK
1983  4712                                  ERRB    TST04I,T4ICON
 (1)                                      ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4712  CD  12  28        18.0                  CALL    ERLPB             ;PROCESS ERROR - DO 2.3
 (1)        001F                                    MSGN    =       MSGN+1            ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4715  1F                                      .BYTE   MSGN              ;MESSAGE NUMBER ID
 (1)  4716  00                                      .BYTE                     ;PRINT ROUTINE NUMBER
 (1)  4717  CD  15  28        18.0          T4ICON::        CALL    CKLOP             ;CHECK LOOP FUNCTION - DO 2.2
 (1)  471A  DA  EF  46        10.0                  JC      TST04I            ;LOOP ADDRESS IF LOOP SPECIFIED
1984                                  ;>WRITE FAIL LINE FOR PARITY NOT SET
1985  471D                                  ENDTST  TEST4
 (1)                                      ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  471D                                    REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  471D  CD  06  28        18.0                                    CALL    REQST
 (2)  4720  00                                                        .BYTE                   ;DATA PATTERN NUMBER
 (2)  4721  00  00                                                    .WORD                   ;SYSTEM "" COUNT
 (2)  4723  00  00                                                    .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4725  00                                                        .BYTE                   ;DATA COMPARE FLAG IF -1
 (2)  4726  07                                                        .BYTE   7               ;REQUEST CODE
 (1)  4727  3A  9A  4F        13.0                  LDA     ITERA             ;GET ITERATION COUNT
 (1)  472A  3D                 4.0                  DCR     A                 ;DOWNCOUNT
 (1)  472B  32  9A  4F        13.0                  STA     ITERA             ;SAVE COUNT
 (1)  472E  F2  8F  45        10.0                  JP      TEST4             ;DO TEST UNTIL TILL = 0
1986  4731  C3  18  28        10.0  EXIT:   JMP     TSTEND
1987
```

```
                                        .SBTTL  SUBROUTINE GCRSUB

                                        ;*SUBROUTINE: GCRSUB
                                        ;*
                                        ;*DESCRIPTION:
                                        ;*
                                        ;*      THIS SUBROUTINE INITIALIZES ALL VARIABLES IN ORDER
                                        ;*      TO RUN THE LOOP WRITE / READ CODE
                                        ;*
                                        ;*      SET TU PORT UNDER TEST
                                        ;*      CLEAR ALL TU PORTS
                                        ;*      SET RECORD SIZE IN CAS
                                        ;*      SET FORMAT IN CAS
                                        ;*      LOAD DATA PATTERN IN DDR
                                        ;*      CALL LWR
                                        ;*
                                        ;*VARIABLES:
                                        ;*      PLLFLG -- IF SET TO NON ZERO WILL NOT ALLOW LWR TO BE CALLED
                                        ;*
                                        ;*SUBROUTINES CALLED:
                                        ;*      CLEAR
                                        ;*      LWR

 4734  06  00          7.0  GCRSUB: MVI     B,@0
 4736  3A  00  49       13.0         LDA     UNITMP          ;DID THE USER SPECIFY TU
 4739  E6  01           7.0         ANI     @01             ;PORT 0?
 473B  C2  51  47       10.0         JNZ     FOUND           ;YES-GO USE PORT #0
 473E  04               4.0         INR     B               ;NO-UPDATE POINTER TO PORT #1
 473F  3A  00  49       13.0         LDA     UNITMP          ;DID THE USER SPECIFY TU
 4742  E6  02           7.0         ANI     @02             ;PORT 1?
 4744  C2  51  47       10.0         JNZ     FOUND           ;YES-GO USE PORT #1
 4747  04               4.0         INR     B               ;NO-UPDATE POINTER TO PORT #2
 4748  3A  00  49       13.0         LDA     UNITMP          ;DID THE USER SPECIFY TU
 474B  E6  04           7.0         ANI     @04             ;PORT 2?
 474D  C2  51  47       10.0         JNZ     FOUND           ;YES-GO USE PORT #2
 4750  04               4.0         INR     B               ;NO-ASSUME PORT #3
 4751  CD  BF  47       18.0  FOUND: CALL    CLEAR
 4754  DB  E0           10.0         IN      INTSTA
 4756  E6  80           7.0         ANI     BIT7
 4758  B0               4.0         ORA     B
 4759  D3  E0           10.0         OUT     MBSEL           ;SELECT THE DESIRED TU-PORT AND MASSBUS

 475B  AF               4.0  T1L01: XRA     A               ;CLEAR THE DATA BUS CONTROL
 475C  32  28  4F       13.0         STA     DBUS            ;MASK TO KEEP THE MASSBUS CLEAN
                                                             ;
 475F  2A  01  49       16.0         LHLD    SIZE            ;GET THE RECORD SIZE
 4762  22  23  4F       16.0         SHLD    BYTFL
 4765  7D               4.0         MOV     A,L             ;GET THE BYTE COUNT LOW
 4766                                ROUT    RO5L            ;SAVE IN BYTE COUNT LOW
 (1)   4766  D3  8A      10.0                 OUT     RO5L    ;WRITE AC INTO RO5L
 (1)   4768  7F          4.0                  MOV     A,A     ;RETRY LINK
 4769  7C               4.0         MOV     A,H             ;GET THE BYTE COUNT HIGH
 476A                                ROUT    RO5H            ;SAVE IN BYTE COUNT HIGH
 (1)   476A  D3  8B      10.0                 OUT     RO5H    ;WRITE AC INTO RO5H
```

```
  (1)  476C  7F                    4.0                MOV     A,A             ;RETRY LINK
2040                                                                           ;
2041   476D  3A  03  49            13.0    LDA     FORMT                       ;GET THE FORMAT
2042   4770  32  25  4F            13.0    STA     FORMAT                      ;NO SKIP COUNT
2043   4773                               ROUT    R02H                        ;SAVE IN THE CAS
  (1)  4773  D3  85                10.0            OUT     R02H            ;WRITE AC INTO R02H
  (1)  4775  7F                    4.0             MOV     A,A             ;RETRY LINK
2044                                                                           ;
```

```
2046  4776  3E  88        7.0        MVI   A,DDRCO          ;CLEAR DDR IN, SO DDR
2047  4778  D3  DB       10.0        OUT   DDRCTL           ;WRITES TO TAPE
2048                                                       ;
2049  477A  2A  04  49    16.0        LHLD  KWDAT            ;GET THE WRITE DATA
2050  477D  29           10.0        DAD   H                ;SHIFT IT LEFT 1 BIT POSITION
2051  477E  17            4.0        RAL                    ;SAVE THE CARRY
2052  477F  29           10.0        DAD   H                ;SHIFT IT LEFT ANOTHER BIT POSITION
2053  4780  17            4.0        RAL                    ;SAVE THAT CARRY TOO.
2054  4781  E6  03        7.0        ANI   $3               ;SAVE ONLY THE CARRY BITS
2055  4783  47            4.0        MOV   B,A              ;SAVE IN REGISTER B
2056  4784  7C            4.0        MOV   A,H              ;
2057  4785  D3  D9       10.0        OUT   DDRB             ;LOAD <13:6> IN DDR B
2058  4787  3A  06  49    13.0        LDA   KWDAT+2          ;GET BITS <17:16>
2059  478A  85            4.0        ADD   L                        ;MERGE WITH <5:0>
2060  478B  D3  D8       10.0        OUT   DDRA             ;LOAD INTO DDR A
2061  478D  AC            4.0        XRA   H                ;COMPUTE PARITY ON ALL BITS
2062  478E  A8            4.0        XRA   B                ;
2063  478F  78            4.0        MOV   A,B              ;GET BITS <15:14> AGAIN
2064  4790  E2  95  47    10.0        JPO   1$               ;JUMP IF ODD PARITY
2065  4793  F6  04        7.0        ORI   BIT2             ;SET THE PARITY BIT TO MAKE IT ODD
2066  4795  D3  DA       10.0  1$:   OUT   DDRC             ;LOAD <P:14> IN DDRC
2067                                                       ;
2068  4797  21  08  49    10.0        LXI   H,PLLFLG         ;GET PHASE LOCKED LOOP FLAG
2069  479A  7E            7.0        MOV   A,M              ;
2070  479B  A7            4.0        ANA   A                ;ARE WE DOING THE PLL TEST
2071  479C  C2  BE  47    10.0        JNZ   QUIT             ;YES , EXIT AND LET PLL TEST CONTROL
2072                                                       ;THE REST
2073  479F  CD  EA  47    18.0        CALL  LWR              ;CALL LOOP WRITE/READ - TU PORT - GCR
2074                                                       ;
2075  47A2  D2  B8  47    10.0  3$:   JNC   T1CN1                ;CONTINUE IF NO ERROR
2076  47A5  78            4.0        MOV   A,B              ;GET THE FAILURE CODE
2077  47A6  E6  FC        7.0        ANI   $FC              ;MASK OUT THE MEANINGFUL BITS
2078  47A8  0F            4.0        RRC                    ;POSITION FOR OUTPUT
2079  47A9  0F            4.0        RRC                    ;
2080  47AA                           ROUT  ADATA                    ;STORE IN THE CAS
 (1)  47AA  D3  94       10.0                OUT   ADATA            ;WRITE AC INTO ADATA
 (1)  47AC  7F            4.0                MOV   A,A              ;RETRY LINK
2081  47AD  79            4.0        MOV   A,C              ;GET THE INTERRUPT CODE
2082  47AE  E6  3F        7.0        ANI   $3F              ;MASK OUT THE MEANINGFUL BITS
2083  47B0                           ROUT  EDATA                    ;STORE IN THE CAS
 (1)  47B0  D3  95       10.0                OUT   EDATA            ;WRITE AC INTO EDATA
 (1)  47B2  7F            4.0                MOV   A,A              ;RETRY LINK
2084  47B3                           ERRB  T1LO1,T1CN1,3
 (1)                               ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  47B3  CD  12  28    18.0                CALL  ERLPB            ;PROCESS ERROR - DO 2.3
 (1)      0020                    MSGN  =     MSGN+1                 ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  47B6  20                    .BYTE MSGN                   ;MESSAGE NUMBER ID
 (1)  47B7  03                    .BYTE 3                      ;PRINT ROUTINE NUMBER
 (1)  47B8  CD  15  28    18.0  T1CN1:: CALL  CKLOP            ;CHECK LOOP FUNCTION - DO 2.2
 (1)  47BB  DA  5B  47    10.0        JC    T1LO1              ;LOOP ADDRESS IF LOOP SPECIFIED
2085                     ;>LOOP WRITE/READ ERROR
2086                     ;<ACTUAL DATA IS THE FAILURE CODE
2087                     ;<EXPECTED DATA IS THE INTERRUPT CODE
2088  47BE  C9           10.0  QUIT:  RET
```

```
2090                                    .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
2091  47BF                              SSUB
 (1)                                    ;*****************************************************************************
 (1)                                    ;*SUBROUTINE TITLE
 (1)                                    ;*---------- -----
2092                                    ;*CLEAR ALL TU PORTS
2093  47BF                              SD
 (1)                                    ;*****************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
2094                                    ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
2095                                    ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
2096                                    ;*AND LOOP MODES.
2097  47BF                              SP
 (1)                                    ;*****************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
2098                                    ;*BGNSUB
2099                                    ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
2100                                    ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
2101                                    ;*     CLEAR PORT SELECT FOR TRANSPORT
2102                                    ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
2103                                    ;*     CLEAR PORT DIAGNOSTIC CONTROL
2104                                    ;*     CLEAR PORT AMTIE WORD
2105                                    ;*ENDSUB
2106  47BF                              S
 (1)                                    ;*****************************************************************************
2107
2108  47BF  F5              12.0  CLEAR:  PUSH    PSW                 ;SAVE THE SELECTED PORT #
2109  47C0  C5              12.0        PUSH    B
2110  47C1  06  00           7.0        MVI     B,0                 ;START TO CLEAR AT PORT #0
2111  47C3  DB  E0          10.0  CLRLP: IN      INTSTA              ;GET MB SELECT INFO
2112  47C5  E6  80           7.0        ANI     BIT7                ;SAVE ONLY THE MASSBUS SELECT BIT
2113  47C7  B0               4.0        ORA     B                   ;ADD IN THE SELECTED PORT #
2114  47C8  D3  E0          10.0        OUT     MBSEL               ;RESET TO THIS PORT
2115  47CA  3E  80           7.0        MVI     A,@200              ;LOAD MTA REGISTER #0 SELECT CODE
2116  47CC  D3  40          10.0        OUT     TCMD                ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
2117  47CE  AF               4.0        XRA     A                   ;CLEAR TU COMMAND A
2118  47CF  D3  40          10.0        OUT     TCMD                ;
2119  47D1  3E  81           7.0        MVI     A,@201              ;LOAD MTA REGISTER #1 SELECT CODE
2120  47D3  D3  40          10.0        OUT     TCMD                ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
2121  47D5  3E  00           7.0        MVI     A,SELCLR            ;LOAD TU "CLEAR SELECT" COMMAND
2122  47D7  D3  40          10.0        OUT     TCMD                ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
2123  47D9  AF               4.0        XRA     A
2124  47DA  D3  44          10.0        OUT     TAMT                ;CLEAR AMTIE WORD
2125  47DC  D3  48          10.0        OUT     PDIAG               ;CLEAR DIAG CONTROL WORD
2126  47DE  D3  4C          10.0        OUT     PENAB               ;CLEAR PORT ENABLE WORD
2127  47E0  04               4.0        INR     B                   ;POINT TO THE NEXT PORT TO CLEAR
2128  47E1  78               4.0        MOV     A,B
2129  47E2  FE  04           7.0        CPI     4                   ;DONE?
2130  47E4  C2  C3  47      10.0        JNZ     CLRLP               ;NO - CLEAR THIS PORT ALSO
2131  47E7  C1              10.0        POP     B                   ;RESET B & C
2132  47E8  F1              10.0        POP     PSW                 ;ALL DONE
2133  47E9  C9              10.0        RET                         ;EXIT
```

```
                                        .SBTTL  LWR - GCR LOOP WRITE TO READ AT PORT BOARD

                                ;*SUBROUTINE: LWR - GCR LOOP WRITE TO READ AT PORT BOARD
                                ;*
                                ;*DESCRIPTION:
                                ;*
                                ;*      THIS SUBROUTINE SETS UP THE TM78 HARDWARE REGISTERS
                                ;*      TO LOOP THE WRITE PATH TO THE READ PATH THROUGH
                                ;*      THE M8955 (TU78 PORT MODULE)
                                ;*      THEN A GCR DATA TRANSFER IS ISSUED
                                ;*      ALL STATUS REGISTERS ARE CHECKED FOR VALID
                                ;*      TERMINATION DATA
                                ;*      IF ANY ERRORS OCCUR THIS SUBROUTINE
                                ;*      WILL RETURN WITH THE "C" BIT SET
                                ;*      AND AN ERROR CODE IN THE REGISTER PAIR B ANN C
                                ;*
                                ;*VARIABLES:
                                ;*      WFFLG -- IF SET TO NON ZERO DISABLES WRITE FAIL CHECKING
                                ;*
                                ;*SUBROUTINES CALLED:
                                ;*      WRTSET -- SET UP WRITE PATH (LOCATED IN OPERATIONAL MICROCODE)
                                ;*
                                ;*


47EA  3E  21          7.0  LWR:    MVI     A,P.AMTP+P.LWR          ;SET LWR AT PORT AND AMTIE PAR BIT
47EC  D3  48         10.0          OUT     PDIAG

                        ;SET THE AMTIES TO SIMULATE IN A GAP BETWEEN RECORDS

47EE  3E  FF          7.0          MVI     A,0FF                   ;SET ALL AMTIE BITS
47F0  D3  44         10.0          OUT     TAMT
47F2  3E  21          7.0          MVI     A,P.LWR+P.AMTP          ;INCLUDING THE PARITY LINE
47F4  D3  48         10.0          OUT     PDIAG                   ;WRITE IT OUT
                        ;INIT THE WMC

47F6  AF              4.0          XRA     A                       ;CLEAR TRACK ENABLE REGISTER
47F7  D3  D2         10.0          OUT     TRKENA                  ; TO COMPLETELY RESET
47F9  D3  D2         10.0          OUT     TRKENA                  ; THE WRITE PATH
47FB  CD  0C  08     18.0          CALL    WRTSET                  ;SET UP ALL WMC REGISTERS
47FE  3E  0D          7.0          MVI     A,RCLRT                 ;ISSUE CLEAR ALL COMMAND TO RMC
4800  D3  0B         10.0          OUT     RCMD                    ;TO PUT THE PLO'S AT REST
                                                                   ;EARLY, TO SIMULATE TAPE START TIME

4802  3A  07  49     13.0          LDA     XFRCTL                  ;GET TRANSFER CONTROL WORD
4805  D3  09         10.0  LWR1:   OUT     RPCTL                   ;LOAD RMC CONTROL REGISTER

                        ;START WRITING

4807  3E  14          7.0          MVI     A,P.WPEN+P.RPEN         ;SET WRITE AND READ ENABLE
4809  D3  4C         10.0  LWR2:   OUT     PENAB
480B  3E  D8          7.0          MVI     A,W.ENAB+X.ENAB+W.WRITE+W.GCR
480D  D3  D3         10.0          OUT     WMCCTL                  ;ENABLE THE WMC AND XMC
```

```
2189
2190                                    ;START READ PATH READING
2191
2192  480F  3E  07          7.0         MVI     A,READG             ;SETUP RMC READ COMMAND
2193  4811  D3  0B         10.0         OUT     RCMD
2194
2195                                    ;LOAD AMTIE BITS
2196
2197  4813  3A  FE  48      13.0         LDA     TSTAMT              ;GET AMTIE WORD
2198  4816  D3  44          10.0         OUT     TAMT
2199  4818  3A  FF  48      13.0         LDA     TSTAMP              ;GET AMTIE PARITY BIT
2200  481B  F6  20           7.0         ORI     P.LWR               ;KEEP LWR BIT
2201  481D  D3  48          10.0         OUT     PDIAG               ;WRITE OUT PORT REGISTER
2202                                    ;WATCH FOR XMC DONE
2203                                    ;TIME OUT AFTER 332.8 MS.
2204
2205  481F  01  8C  57      10.0         LXI     B,22412.            ;CLEAR B,C. SET OPERATION TIMER
2206
2207  4822  DB  D0          10.0  LWR8A:  IN      WMCSTA              ;GET WMC,XMC DONE WORD
2208  4824  A7             4.0         ANA     A                   ;MANIPULATE CONDITION CODES
2209                                                                ;SEE IF "XMC IS DONE (AC = POSITIVE)
2210  4825  F2  3D  48      10.0         JP      LWR9                ;YES! WRITE DATA XFR IS FINISHED'
2211  4828  0B             6.0         DCX     B                   ;DECREMENT LOOP COUNTER
2212  4829  78             4.0         MOV     A,B                 ;LOOP UNTIL DONE OR TIMED OUT
2213  482A  B1             4.0         ORA     C
2214  482B  C2  22  48      10.0         JNZ     LWR8A               ;LOOP UNTIL DONE OR TIMED OUT
2215
2216                                    ;IF WE GET HERE THE READ-AFTER-WRITE HAS TIMED OUT
2217                                    ;QUIT!
2218
2219  482E  01  18  48      10.0         LXI     B,130+F22           ;SET READ-AFTER-WRITE TIME OUT INTERRUPT CODE
2220  4831  DB  E0          10.0         IN      INTSTA
2221  4833  1F             4.0         RAR
2222  4834  D2  B1  48      10.0         JNC     LWREX
2223  4837  01  18  1C      10.0         LXI     B,130+F07
2224  483A  C3  B1  48      10.0         JMP     LWREX               ;ERROR. GO STOP MOTION!
2225
2226                                    ;XMC IS FINISHED. WRITE DATA XFR IS DONE.
2227
2228                                    ;SET ALL AMTIE BITS TO SATISFY RMC
2229
2230  483D  3E  05           7.0  LWR9:   MVI     A,5                 ;DELAY TO ALLOW READ PATH TO
2231  483F  3D             4.0  LWR9L:  DCR     A                   ; DETECT LONG POSTAMBLE ERROR
2232  4840  C2  3F  48      10.0         JNZ     LWR9L               ; AND ALLOW PLO'S TO DRIFT
2233  4843  3E  FF           7.0         MVI     A,0FF               ;SET ALL AMTIE BITS
2234  4845  D3  44          10.0         OUT     TAMT                ;BITS <7:0>
2235  4847  3E  21           7.0         MVI     A,P.LWR+P.AMTP  ;SET PARITY AMITE BIT
2236  4849  D3  48          10.0         OUT     PDIAG               ;WRITE
2237
2238  484B  3E  00           7.0         MVI     A,0                 ;DELAY ABOUT 1 MS. TO ALLOW
2239  484D  3D             4.0  LWR10:  DCR     A                   ;  READ PATH TO TIME OUT
2240  484E  C2  4D  48      10.0         JNZ     LWR10               ;  0.1 IN. GAP AND FINISH
2241                                    ;CHECK ERROR BITS
2242
```

```
2243  4851  3E 9B        7.0          MVI   A,DDRCO+DDRAIN+DDRBIN+DDRCIN    ;SET DDR IN
2244  4853  D3 DB       10.0          OUT   DDRCTL                          ;SO M.RDPE CAN BE READ IN WMCERR REG
2245  4855  DB DA       10.0          IN    WMCERR                          ;CHECK WMC ERROR BITS
2246  4857  32 0A 49    13.0          STA   SWMCER                          ;COPY REGISTER TO MEMORY
2247  485A  E6 38        7.0          ANI   W.ERR+W.ROME+M.RDPE
2248  485C  C2 BC 48    10.0          JNZ   LWRER2                          ;ERROR IF ANY SET
2249
2250                                ;CHECK IF FINAL RMC STATUS IS ACCEPTABLE
2251
2252  485F  DB 02       10.0          IN    RSTAT                           ;READ RMC STATUS WORD
2253  4861  32 0B 49    13.0          STA   SRSTAT                          ;STORE IN MEMORY
2254  4864  FE B1        7.0          CPI   PL                              ;CHECK IF OKAY
2255  4866  DA B6 48    10.0          JC    LWRER1                          ,IF NOT, ERROR
2256
2257
2258  4869  21 09 49    10.0          LXI   H,WFFLG          ;WRITE FAIL TEST ?
2259  486C  7E           7.0          MOV   A,M                             ;GET THE FLAG
2260  486D  A7           4.0          ANA   A                               ;SET CONDITION CODE
2261  486E  CA 83 48    10.0          JZ    LWRCHK                          ;NO , DO THIS ERROR CHECKING
2262
2263                                ;WRITE FAIL TEST FLAG IS SET USE THIS CODE
2264  4871  DB 00       10.0          IN    RPFAIL                          ;LOOK AT WRITE FAIL BITS
2265  4873  32 0C 49    13.0          STA   SRPFAL                          ;STORE IN RAM
2266  4876  DB 01       10.0          IN    RPATH                           ;LOOK AT WRITE FAIL PARITY BIT
2267  4878  32 0D 49    13.0          STA   SRPATH                          ;STORE IN RAM
2268  487B  DB 1A       10.0          IN    ECCSTA                          ;CHECK ECC STATUS BITS
2269  487D  32 0E 49    13.0          STA   SECCST                          ;STORE IN RAM
2270  4880  C3 9E 48    10.0          JMP   LWRXX                           ;EXIT NORMALLY
2271
2272                                ;NORMAL LOOP WRITE TO READ CODE
2273
2274  4883  DB 00       10.0  LWRCHK: IN    RPFAIL                          ;LOOK AT WRITE FAIL BITS
2275  4885  32 0C 49    13.0        STA   SRPFAL                          ;STORE IN RAM
2276  4888  A7           4.0        ANA   A                               ;
2277  4889  C2 CA 48    10.0        JNZ   LWRER3                          ;ERROR IF NOT ZERO
2278  488C  DB 01       10.0        IN    RPATH                           ;LOOK AT WRITE FAIL PARITY BIT
2279  488E  32 0D 49    13.0        STA   SRPATH                          ;STORE IN RAM
2280  4891  1F           4.0        RAR                                   ;SHIFT TO "C" BIT
2281  4892  DA CA 48    10.0        JC    LWRER3                          ;ERROR IF NOT ZERO
2282
2283                                ;CHECK ERROR BITS FOR GCR
2284
2285  4895  DB 1A       10.0  LWRNXA: IN    ECCSTA                          ;CHECK ECC STATUS BITS
2286  4897  32 0E 49    13.0        STA   SECCST                          ;STORE IN RAM
2287  489A  A7           4.0        ANA   A                               ;LOOK FOR ANY ERROR BITS
2288  489B  C2 EA 48    10.0        JNZ   LWRER7                          ;ERROR IF ANY SET
2289
2290                                ;NO ERROR EXIT
2291
2292  489E  DB DA       10.0  LWRXX:  IN    WMCERR                          ;READ MASSBUS PARITY ERROR BIT
2293  48A0  32 0A 49    13.0        STA   SWMCER                          ;STA IN RAM
2294  48A3  E6 40        7.0        ANI   M.PE
2295  48A5  C2 F8 48    10.0        JNZ   LWRER8                          ;JUMP IF SET
2296  48A8  3E 84        7.0        MVI   A,P.INTEN+P.RPEN                ;CLEAR DATA PATH ENABLE
```

```
2297  48AA  D3  4C        10.0          OUT     PENAB
2298  48AC  01  01  00     10.0          LXI     B,101                ;SET UP DONE INTERRUPT CODE
2299  48AF  AF             4.0          XRA     A                    ;CLEAR CARRY
2300  48B0  C9            10.0          RET
2301
2302                                  ;ERROR EXIT
2303
2304  48B1  AF             4.0  LWREX:  XRA     A
2305  48B2  D3  4C        10.0          OUT     PENAB
2306  48B4  37             4.0          STC                          ;SET CARRY TO INDICATE ERROR
2307  48B5  C9            10.0          RET
2308                                  ;ERROR EXITS
2309
2310  48B6  01  15  20     10.0  LWRER1: LXI     B,125+F10            ;RSTAT CONTAINS ERROR CODE
2311  48B9  C3  B1  48     10.0          JMP     LWREX
2312
2313  48BC  01  18  4C     10.0  LWRER2: LXI     B,130+F23            ;WMC ROM PE OR RD PE SET IN WMCERR
2314  48BF  E6  20          7.0          ANI     W.ERR                ;REGISTER
2315  48C1  CA  B1  48     10.0          JZ      LWREX
2316  48C4  01  18  0C     10.0          LXI     B,130+F03
2317  48C7  C3  B1  48     10.0          JMP     LWREX
2318
2319  48CA  01  15  18     10.0  LWRER3: LXI     B,125+F06            ;AT LEAST ONE WRITE FAIL BIT IS SET IN RPFAIL
2320  48CD  C3  B1  48     10.0          JMP     LWREX                ;OR RPATH REGISTERS
2321
2322  48D0  01  15  10     10.0  LWRER4: LXI     B,125+F04            ;AMTIE, PNTR MISMATCH, UNCORRECTABLE, 2 TRK ERR
2323  48D3  E6  40          7.0          ANI     E.RPE                ;OR SINGLE TRACK ERROR SET IN ECCSTA REGISTER
2324  48D5  CA  B1  48     10.0          JZ      LWREX
2325  48D8  01  18  18     10.0          LXI     B,130+F06
2326  48DB  C3  B1  48     10.0          JMP     LWREX
2327
2328  48DE  01  18  44     10.0  LWRER5: LXI     B,130+F21            ;XL PE SET IN INTSTA REGISTER
2329  48E1  C3  B1  48     10.0          JMP     LWREX
2330
2331  48E4  01  15  24     10.0  LWRER6: LXI     B,125+F11            ;CRC CHARACTERS FROM WMC AND RMC DID NOT MATCH
2332  48E7  C3  B1  48     10.0          JMP     LWREX
2333
2334  48EA  01  15  14     10.0  LWRER7: LXI     B,125+F05            ;AT LEAST ONE BIT SET IN ECCSTA REGISTER
2335  48ED  E6  40          7.0          ANI     E.RPE
2336  48EF  CA  B1  48     10.0          JZ      LWREX
2337  48F2  01  18  18     10.0          LXI     B,130+F06
2338  48F5  C3  B1  48     10.0          JMP     LWREX
2339
2340  48F8  01  15  28     10.0  LWRER8: LXI     B,125+F12            ;MASSBUS DATA BUS PARITY ERROR
2341  48FB  C3  B1  48     10.0          JMP     LWREX
2342
2343                                  .SBTTL  PROGRAM VARIABLES
2344
2345  48FE  00                      TSTAMT: .BYTE   0                ;AMTIE BYTE TO LOAD INTO TAMI
2346  48FF  00                      TSTAMP: .BYTE   0                ;AMTIE PARITY BYTE
2347  4900  00                      UNITMP: .BYTE   0                ;UNIT MAP
2348  4901  00  00                  SIZE:   .WORD   0                ;RECORD SIZE
2349  4903  00                      FORMT:  .BYTE   0                ;DATA FORMAT
2350  4904  00                      KWDAT:  .BYTE   0                ;18 BITS OF INPUT DATA
```

```
2351  4905  00                        .BYTE   0
2352  4906  00                        .BYTE   0
2353  4907  00              XFRCTL:   .BYTE   0             ;RPCTL SOFTWARE REGISTER
2354  4908  00              PLLFLG:   .BYTE   0             ;PHASE LOCKED LOOP TEST FLAG
2355  4909  00              WFFLG:    .BYTE   0             ;WRITE FAIL TEST FLAG
2356
2357                        ;TEMPORARY REGISTER SAVE LOCATIONS
2358
2359  490A  00              SWMCER:   .BYTE   0             ;WMCERR SAVE LOCATION
2360  490B  00              SRSTAT:   .BYTE   0             ;RSTAT SAVE LOCATION
2361  490C  00              SRPFAL:   .BYTE   0             ;RPFAIL SAVE LOCATION
2362  490D  00              SRPATH:   .BYTE   0             ;RPATH SAVE LOCATION
2363  490E  00              SECCST:   .BYTE   0             ;ECCSTA SAVE LOCATION
2364        0000            .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE    4F9B          ATTCD    4F97         AXNUM    4F91
B      =%0000          BADST  = 0090          BIT0   = 0001         BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008         BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080         BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00           BSAVE    4F9C          BYTCNT= 00D4          BYTEH    4F24
BYTEL    4F23          C      =%0001          CASCT    4F21         CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088         CBUSST= 00A1
CBYTH  = 008B          CBYTL  = 008A          CDG1H  = 0087         CDG1L  = 0086
CDG2H  = 0093          CDG2L  - 0092          CDG3H  = 0095         CDG3L  = 0094
CDVTH  = 008D          CDVTL  = 008C          CHPCON   4426  G      CHPCO2   450B  G
CHPTIE= 0028           CH0CON   4386  G       CH0CO2   446B  G      CH0TIE= 0020
CH1CON   439A  G       CH1CO2   447F  G       CH1TIE= 0021          CH2CON   43AE  G
CH2CO2   4493  G       CH2TIE= 0022           CH3CON   43C2  G      CH3CO2   44A7  G
CH3TIE= 0023           CH4CON   43D6  G       CH4CO2   44BB  G      CH4TIE= 0024
CH5CON   43EA  G       CH5CO2   44CF  G       CH5TIE= 0025          CH6CON   43FE  G
CH6CO2   44E3  G       CH6TIE= 0026           CH7CON   4412  G      CH7CO2   44F7  G
CH7TIE= 0027           CKLOP  = 2815          CLEAR    47BF         CLKCTL= 00F0
CLOCK    4F26          CLRLP    47C3          CMC0H  = 0099         CMC0L  = 0098
CMC1H  = 009B          CMC1L  = 009A          CMC2H  = 009D         CMC2L  = 009C
CMC3H  = 009F          CMC3L  = 009E          CMINH  = 0097         CMINL  = 0096
CNTCTL= 00D7           CRCWRD= 0018           CSAVE    4F9D         CSRLH  = 0091
CSRLL  = 0090          CTCH   = 0085          CTCL   = 0084         CTSTH  = 008F
CTSTL  = 008E          CXCTH  = 0081          CXCTL  = 0080         CXINH  = 0083
CXINL  = 0082          C.     = 0001          C.AVAI= 0080          C.DP   = 0C08
C.DSE  = 0010          C.DTU  = 0003          C.DVA  = 0008         C.FAIL= 00FC
C.FMT  = 0070          C.FNCT= 003E           C.GO   = 0001         C.INTC= 00FE
C.MAIN= 0020           C.NSA  = 0080          C.RCT  = 00FC         C.SER  = 0080
C.SHR  = 0040          C.SKPC= 000F           C.TAPE= 0040          C.WCS  = 0002
D      =%0002          DATACT= 00D0           DBUS     4F28         DBUSCT= 00C0
DBUSST= 00C0           DDRA   = 00D8          DDRAIN= 0010          DDRB   = 00D9
DDRBIN= 0002           DDRC   = 00DA          DDRCIN= 0001          DDRCO  = 0088
DDRCTL= 00DB           DIAFLG   4F22          DIAGPG= 4300          DIAGRM= 4F90
DIARD  = 0008          DONE1  = 0045          DONINT= 0010          DSAVE    4F9E
DSE    = 0006          DUMMY    431E  G       D.ATH0= 0001          D.ATH1= 0002
D.EOTD= 0010           D.LAGC= 0020           D.NOTW= 0040          D.NTHR= 0004
D.TACH= 0008           D.WR4  = 0080          E      =%0003         ECCBAD= 0042
ECCCOR= 0019           ECCOK  = 0041          ECCSTA= 001A          ECCTST= 000E
EDATA  = 0095          ENDI2    4511          EOTCLR= 0003          ERFLG    4F93
ERLP   = 2809          ERLPA  = 280F          ERLPB  = 2812         ERLPE  = 280C
ERNUM    4F90          ERRCNT= 00D6           ESAVE    4F9F         EXIT     4731
E.ACRC= 0010           E.AMT  = 0020          E.CDP  = 0080         E.CRC  = 0080
E.PNTR= 0008           E.RPE  = 0040          E.STEC= 0001          E.TTEC= 0002
E.UNC  = 0004          FIFORD= 006A           FORMAT   4F25         FORMT    4903
FOUND    4751          FWDTST= 0061           F03    = 0C00         F04    = 1000
F05    = 1400          F06    = 1800          F07    = 1C00         F10    = 2000
F11    = 2400          F12    = 2800          F21    = 4400         F22    = 4800
F23    = 4C00          GCRID  = 0089          GCRSET= 0002          GCRSUB   4734
GOODTM= 0092           H      =%0004          HLSAVE   4FA0         IE     = 0008
INTSTA= 00E0           ITERA    4F9A          I.PWR  = 0020         I.RMPE= 0040
I01    = 0001          I25    = 0015          I30    = 0018         I5.5   = 0010
I6.5   = 0020          I7.5   = 0040          KCALL  = 005F         KCLR   = 007B
KDEP   = 003F          KENAB  = 0078          KEXAM  = 003E         KEYBRD= 00C8
```

```
KEY1   = 0078    KEY10  = 006D    KEY11  = 006E    KEY12  = 006F
KEY13  = 005C    KEY14  = 005D    KEY15  = 005E    KEY16  = 005F
KEY17  = 003C    KEY18  = 003D    KEY19  = 003E    KEY2   = 0079
KEY20  = 003F    KEY3   = 007A    KEY4   = 007B    KEY5   = 0074
KEY6   = 0075    KEY7   = 0076    KEY8   = 0077    KEY9   = 006C
KINTA  = 006F    KLDAD  = 003D    KN0    = 003C    KN1    = 005C
KN2    = 005D    KN3    = 005E    KN4    = 006C    KN5    = 006D
KN6    = 006E    KN7    = 0074    KN8    = 007D    KN9    = 0076
KU2    = 0079    KU3    = 007A    KU8    = 0077    KWDAT    4904
L      =%0005    LBLANK= 000F    LCE    = 000B    LCH    = 000C
LCL    = 000D    LCP    = 000E    LC0    = 0000    LC1    = 0001
LC2    = 0002    LC3    = 0003    LC4    = 0004    LC5    = 0005
LC6    = 0006    LC7    = 0007    LC8    = 0008    LC9    = 0009
LDLEDA= 00CA    LDLEDB= 00CB    LDLEDC= 00CC    LDLEDD= 00CD
LDLEDE= 00CE    LDLEDF= 00CF    LKDIAG= 2800    LKKBD  = 004C
LKKEY  = 0049    LKLWMG= 0058    LKLWMP= 0055    LKLWPG= 0052
LKLWPP= 004F    LKMOD7= 0046    LKOPR  = 0046    LPFLG    4F94
LPNUM    4F92    LWR      47EA    LWRCHK   4883    LWRER1   48B6
LWRER2   48BC    LWRER3   48CA    LWRER4   48D0    LWRER5   48DE
LWRER6   48E4    LWRER7   48EA    LWRER8   48F8    LWREX    48B1
LWRNXA   4895    LWRXX    489E    LWR1     4805    LWR10    484D
LWR2     4809    LWR8A    4822    LWR9     483D    LWR9L    483F
M      =%0006    MBSEL  = 00E0    MB.A   = 0008    MB.B   = 0004
MEMTOP= 4FFF    MINUS  = 000A    MM     = 8000    MSE    = 0008
MSGN   = 0020    MTACLR= 0000    MT.ARA= 0020    MT.CPE= 0080
MT.DSE= 0001    MT.FWD= 0040    MT.INH= 0008    MT.LWR= 0004
MT.MOT= 0002    MT.NWT= 0080    MT.PEC= 0040    MT.PSB= 0004
MT.PS0= 0001    MT.PS1= 0002    MT.REV= 0020    MT.WRT= 0010
MT.Z   = 0008    M.ATA  = 0080    M.CAPE= 0020    M.CONT= 0080
M.DEM  = 0020    M.EBL  = 0010    M.EXC  = 0008    M.FAIL= 0008
M.ILR  = 0010    M.INIT= 0010    M.OCC  = 0020    M.ONLI= 0001
M.PE   = 0040    M.PORT= 0080    M.RDEN= 0002    M.RDPE= 0008
M.RUN  = 0004    M.SCLK= 0001    M.TRA  = 0040    M.UNIT= 0007
M.WCLK= 0040    M.WCLN= 0080    M.WREN= 0080    M5.5   = 0001
M6.5   = 0002    M7.5   = 0004    NOTCAP= 0088    OKAY   = 00FF
OPRRAM= 4300    OPSTRT= 0058    OPVER  = 0040    PADCNT= 00D5
PADCRC= 0080    PDIAG  = 0048    PEID   = 008A    PENAB  = 004C
PESET  = 0001    PL       = 00B1    PLLFLG   4908    PRDD   = 004C
PRENF  = 009C    PS       = 00B2    PSTAT  = 0048    PSW    =%0009
P.AMTP= 0001    P.BCTC= 0040    P.CMDP= 0020    P.INTE= 0080
P.LCS  = 0040    P.LWR  = 0020    P.RDP  = 0002    P.RPEN= 0004
P.RPST= 0002    P.RPOE= 0020    P.RP1E= 0010    P.RP2E= 0020
P.RP3E= 0010    P.SING= 0080    P.STAT= 0002    P.STPE= 0080
P.TACH= 0008    P.TUPR= 0010    P.WCSD= 0004    P.WDS  = 0040
P.WFLP= 0001    P.WPEN= 0010    P.WPOE= 0008    P.WP1E= 0004
P.WP2E= 0008    P.WP3E= 0004    P.5VOK= 0002    QUE    = 281B
QUEM   = 281E    QUIT     47BE    RAMT   = 0010    RARA   = 0006
RARAI  = 0004    RCHBD0= 0043    RCHBD1= 0047    RCHOK  = 0046
RCHTST= 000C    RCLRT  = 000D    RCMD   = 000B    RCMLP  = 0003
RCONT  = 0080    RDATA  = 0017    RDCLK  = 0010    RDON   = 0011
READG  = 0007    REND   = 0014    REQST  = 2806    RESCHR= 00D1
REVTST= 0064    REWIND= 0004    RFIFOL= 0008    RGCLK  = 0002
RGCRI  = 0003    RIBG   = 0001    RILL   = 0012    RINST  = 000C
RMCTST= 0008    RMK2   = 0013    RNOP   = 0000    RPATH  = 0001
```

```
RPBAD = 0044            RPCHI = 0001            RPCLK = 0003            RPCTL = 0009
RPEI  = 0002            RPFAIL= 0000            RPF1  = 009D            RPF2  = 009E
RPOK  = 0043            RPOSTN= 0016            RPSTA = 0015            RRCMT = 000A
RSTAT = 0002            RTIEB = 000A            RTIER = 0030            RTM   = 0005
RUNKI = 0009            RUPTST= 005E            RWDUNL= 0005            R.AMT = 0001
R.BOP = 0008            R.DATA= 0040            R.DON = 0002            R.DRDY= 0010
R.END = 0010            R.ILL = 0004            R.JVOK= 0004            R.MK2 = 0008
R.PLOD= 0008            R.PLOO= 0010            R.PLO1= 0020            R.POST= 0020
R.STNM= 0002            R.STOP= 0004            R.STPC= 0001            R.TBJN= 0080
R.TSTD= 0040            R.VOK = 0080            R00H  = 0081            R00L  = 0080
R01H  = 0083            R01L  = 0082            R02H  = 0085            R02L  = 0084
R03H  = 0087            R03L  = 0086            R04H  = 0089            R04L  = 0088
R05H  = 008B            R05L  = 008A            R06H  = 008D            R06L  = 008C
R07H  - 008F            R07L  = 008E            R10H  = 0091            R10L  = 0090
R11H  = 0093            R11L  = 0092            R12H  = 0095            R12L  = 0094
R13H  = 0097            R13L  = 0096            R14H  = 0099            R14L  = 0098
R15H  = 009B            R15L  = 009A            R16H  = 009D            R16L  = 009C
R17H  = 009F            R17L  = 009E            R7.5  = 0010            SECCST  490E
SELCLR= 0000            SETATA= 00A1            SID   = 0080            SIZE    4901
SOD   = 0080            SOE   = 0040            SP    =%0008            SRPATH  490D
SRPFAL  490C            SRSTAT  490B            SSCLK = 0040            SSTEP = 0005
STACK = 4FFF            STATRM= 4F20            STPCT   4F20            STRSP = 5000
SWMCER  490A            TADR00= 0080            TADR01= 0081            TADR02= 0082
TADR03= 0083            TADR04= 0084            TADR05= 0085            TADR06= 0086
TADR07= 0087            TADR10= 0088            TADR11= 0089            TADR12= 008A
TADR13= 008B            TAMI  = 0044            TASEL = 0080            TCMD  = 0040
TC.FWD= 0040            TC.INH= 0008            TC.LWR= 0004            TC.REV= 0020
TC.WRT= 0010            TEMP    4F99            TEST1   4300            TEST2   4440
TEST3   4525            TEST4   458F            TMF   = 0099            TMRDY = 0040
TRKENA= 00D2            TSET  = 2803            TSTAMP  48FF            TSTAMT  48FE
TSTEND= 2818            TSTS  = 0040            TST01   4324            TST01A  4370
TST03B  4560            TST04A  4594            TST04B  45C2            TST04C  45ED
TST04D  4618            TST04E  4643            TST04F  466E            TST04G  4699
TST04H  46C4            TST04I  46EF            TUSEL0= 00D1            TUSEL1= 00D2
TU78  = 0010            T.ATH0= 0001            T.ATH1= 0002            T.BOT = 0004
T.EOT = 0002            T.FPT = 0001            T.NTHR= 0004            T.ONL = 0020
T.PES = 0008            T.PSBJ= 0020            T.PSOJ= 0008            T.PS1J= 0010
T.RDY = 0080            T.RDYC= 0040            T.RWD = 0010            T.SCLK= 0002
T1CN1   47B8  G         T1CON1  436A  G         T1LO1   475B            T3ACON  455A  G
T3ACO1  4575  G         T4ACON  45BC  G         T4BCON  45E7  G         T4CCON  4612  G
T4DCON  463D  G         T4ECON  4668  G         T4FCON  4693  G         T4GCON  46BE  G
T4HCON  46E9  G         T4ICON  4717  G         UIBG  = 00A1            UNITMP  4900
VALFC   4F98            VALTB   4F95            VELTST= 005B            WDR.P = 0010
WFFLG   4909            WMCCTL= 00D3            WMCERR= 00DA            WMCSTA= 00D0
WRTCLK= 000U            WRTDAT= 00D3            WRTSET= 080C            W.ACRC= 0004
W.CRC = 0008            W.DIAG= 0002            W.DONN= 0040            W.ECC = 0010
W.ENAB= 0080            W.ERR = 0020            W.FMT = 0070            W.GCR = 0010
W.LEFT= 0004            W.ONES= 0020            W.RESI= 0002            W.REV = 0004
W.ROME= 0010            W.RST = 0001            W.SKIP= 000F            W.WRIT= 0008
W.XFER= 0020            X     =%000A            XFRCTL  4907            X.DONN= 0080
X.ENAB= 0040            X.PEPE= 0002            X.ROME= 0001            X.WCLK= 0001
Y     =%000B            .     = 490F
```

```
 ERRORS DETECTED:  0

 *RPM6.A78/PTP,RPM6=NLIST,PARAM,MACRO,LIST,RPM6
 RUN-TIME: 4 7 0 SECONDS
 CORE USED:  10K
```

3	"MEMORY MAP" EXTENDED ADDRESS BIT
19	READ MICRO CONTROLLER CONTROL & STATUS WORDS
139	TRANSPORT/PORT COMMAND + STATUS WORDS
380	COMMON ADDRESS SPACE REGISTERS
444	MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485	MASSBUS CONTROL WORD DEFINITIONS
557	WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709	KEYBOARD/DISPLAY CONTROL + STATUS WORD
797	MISCELLANEOUS VARIABLE DEFINITIONS
801	RAM ADDRESS DEFINITIONS
819	LINKAGE TABLE ADDRESSES
842	BIT DEFINITIONS
856	RIM AND SIM INSTRUCTION BIT DEFINITIONS
877	INTERRUPTABLE INSTRUCTION MACROS
924	DIAGNOSTIC VARIABLES
945	COMMON VARIABLE STORAGE
959	MACRO CALL DEFINITIONS
1047	TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101	8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1333	TEST 1 - LOOP WRITE/READ - TU PORT - GCR - NORMAL W/R CLOCKS
1414	TEST 2 - LOOP WRITE/READ - TU PORT - GCR - +10% W/R CLOCKS
1473	TEST 3 - LOOP WRITE/READ - TU PORT - GCR - -20% W -30% R CLOCKS
1531	TEST 4 - LOOP WRITE/READ - TU PORT - PE - NORMAL W/R CLOCKS
1591	TEST 5 - LOOP WRITE/READ - TU PORT - PE - +10% W/R CLOCKS
1650	TEST 6 - LOOP WRITE/READ - TU PORT - PE - -20% W -30% R CLOCKS
1709	SUBROUTINE PESUB
1797	SUBROUTINE CLEAR ALL TU PORTS
1842	PROGRAM VARIABLES

```
                ;*************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;                "BYTE/SCLK COUNT NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALJE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                MM = DATA FORMAT FROM CAS REGISTER 2
                ;                NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;                BYTE-SCLK COUNT = LLL
                ;                DATA FORMAT = MM
                ;                SKIP COUNT  = NN
                ;                WHERE:
                ;                LLL = AS ABOVE
                ;                MM  = AS ABOVE
                ;                NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;                TRANSITION COUNT = LLL
                ;                WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;                EXPECTED 18 BITS =      E EEEEEE
                ;                ACTUAL 18 BITS =        A AAAAAA
                ;
                ;                WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;                BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;                OF CAS REG 15 LOW BYTE.
                ;
                ;                ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;                AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;                17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;                TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;                AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;                "SUBGROUP NUMBER = LLL"
                ;                WHERE:
                ;                LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                      (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       10 -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ;*************************************************************************
```

```
1016                    ; ***************************************************************************
1017                    ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
1018                    ;TO CAUSE SOME HOST CPU ACTION.
1019                    ;
1020                    ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
1021                    ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
1022                    ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
1023                    ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 73
1024                    ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
1025                    ;               - HOST RESPONSE CODE 31 OR 33
1026                    ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
1027                    ;               - HOST RESPONSE CODE 31 OR 33
1028                    ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
1029                    ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
1030                    ;                       BIT0 = 1 TEST PORT 0
1031                    ;                       BIT1 = 1 TEST PORT 1
1032                    ;                       BIT2 = 1 TEST PORT 2
1033                    ;                       BIT3 = 1 TEST PORT 3
1034                    ; ***************************************************************************
1035                    ; ***************************************************************************
1036                    ;DATA PATTERN CODES
1037                    ;
1038                    ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
1039                    ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
1040                    ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
1041                    ;               18 BITS OF ALL 1'S
1042                    ;               18 BITS OF ALL 0'S
1043                    ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
1044                    ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
1045                    ; ***************************************************************************
1329    4300                    .-      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE  TUP2 - TAPE UNIT PORT TEST PART #2
                                .SBTTL  TEST 1 - LOOP WRITE/READ - TU PORT - GCR - NORMAL W/R CLOCKS
                                ;ID     TUP2-TAPE UNIT PORT CONTROLLER PART #2
1335  4300                      ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
1336                            ;*GCR - LOOP WRITE/READ - TU PORT - NORMAL W/R CLOCKS
1337  4300                      SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
1338                            ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
1339                            ;*GCR FORMAT WITH NORMAL WRITE AND READ CLOCKS
1340                            ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
1341  4300                      SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
1342                            ;*BGNTST
1343                            ;*  GET USER SELECTED TU PORT
1344                            ;*  SET THE RECORD SIZE TO 128
1345                            ;*  BGNDO
1346                            ;*  :  SET THE WRITE PATH CLOCK TO NORMAL
1347                            ;*  :  SET THE READ PATH CLOCK TO NORMAL
1348                            ;*  :  CALL SUBROUTINE GCRSUB
1349                            ;*  :  ADD 128 TO THE RECORD SIZE
1350                            ;*  :  DO UNTIL THE RECORD SIZE=640
1351                            ;*  ENDDO
1352                            ;*  SET RECORD SIZE TO 1112.
1353                            ;*  CALL SUBROUTINE GCRSUB
1354                            ;*ENDTST
1355  4300                      SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
1356                            ;*TUP2 MICRO TEST 01
1357                            ;*TUP2 MICRO ERROR 01
1358                            ;*TUP2-LOOP WRITE/READ - TU PORT - GCR - NORMAL W/R CLOCKS
1359                            ;*ALL MODULES
1360                            ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
1361                            ;*FATAL ERROR - TEST ABORTED
1362                            ;*
1363                            ;*TUP2 MICRO TEST 01
1364                            ;*TUP2 MICRO ERROR 02
1365                            ;*TUP2-LOOP WRITE/READ - TU PORT - GCR - NORMAL W/R CLOCKS
1366                            ;*ALL MODULES
1367                            ;*LOOP WRITE/READ ERROR
1368                            ;*ACTUAL = AAAA
1369                            ;*EXPECTED = EEEE
1370                            ;*BYTE-SCLK COUNT = LLL
1371                            ;*DATA FORMAT = MM
1372                            ;*SKIP COUNT = NN
1373                            ;*ACTUAL DATA IS THE FAILURE CODE
```

```
                                        ;*EXPECTED DATA IS THE INTERRUPT CODE
4300                                    ;
 (1)                                    ; ************************************************************************
                                        ;
4300  C3  07  43      10.0              JMP     TEST1
4303  00                                .BYTE   0               ;DUMMY BYTE
4304  00                                .BYTE   0               ;DUMMY BYTE
4305  00                                .BYTE   0               ;DUMMY BYTE
4306  00                          SPEED:.BYTE   0               ;CLOCK CONTROL WORD
4307                              TEST1:TESTX   @1              ;INITIALIZE THE TEST
 (1)  4307  3E  01       7.0                        MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4309  CD  03  28   18.0                       CALL    TSET            ;SETUP THE TEST
                                        ;%TUP2-LOOP WRITE/READ - TU PORT - GCR - NORMAL W/R CLOCKS
                                        ;&ALL MODULES
430C                                    REQ     @7,0,0,0,0
 (1)  430C  CD  06  28   18.0                       CALL    REQST
 (1)  430F  00                                      .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4310  00  00                                  .WORD   0               ;SYSTEM "0" COUNT
 (1)  4312  00  00                                  .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  4314  00                                      .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1)  4315  07                                      .BYTE   @7              ;REQUEST CODE
4316                                    RIN     R12L
 (1)  4316  DB  94       10.0                       IN      R12L            ;READ R12L INTO AC
 (1)  4318  7F            4.0                       MOV     A,A             ;RETRY LINK
4319  32  77  45      13.0              STA     UNITMP
431C  A7                 4.0              ANA     A
431D  C2  2B  43      10.0              JNZ     TST01X
4320                                    ERR     EXIT,DUMMY
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4320  CD  09  28   18.0                       CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)      0001                                      MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4323  01                                      .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4324  00                                      .BYTE
 (1)  4325  CD  15  28   18.0                DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4328  DA  A6  44   10.0                       JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
                                        ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                        ;>FATAL ERROR - TEST ABORTED
```

```
1394  432B  21  80  00   10.0  TST01X: LXI   H,128              ;SET UP THE BYTE COUNT
1395  432E  22  78  45   16.0          SHLD  SIZE               ;
1396                                                           ;
1397  4331  3E  10        7.0  TST01Y: MVI   A,@020             ;SET UP FOR NORMAL CLOCKS
1398  4333  32  06  43   13.0          STA   SPEED
1399  4336  CD  B0  44   18.0          CALL  GCRSUB        ;CALL SUBROUTINE LWRSUB
1400                                                           ;
1401  4339  2A  78  45   16.0          LHLD  SIZE               ;GET THE SIZE
1402  433C  01  80  00   10.0          LXI   B,128              ;LOAD THE INCREMENT VALUE
1403  433F  09           10.0          DAD   B                  ;INCREMENT THE BYTE COUNT
1404  4340  22  78  45   16.0          SHLD  SIZE               ;SAVE THE SIZE FOR LATER
1405  4343  7C           4.0          MOV   A,H                ;DO UNTIL THE BYTE COUNT = 768
1406  4344  FE  03       7.0          CPI   $3                 ;
1407  4346  C2  31  43   10.0          JNZ   TST01Y             ;
1408  4349  21  58  04   10.0          LXI   H,1112.            ;SET UP THE BYTE COUNT
1409  434C  22  78  45   16.0          SHLD  SIZE               ;
1410  434F  CD  B0  44   18.0          CALL  GCRSUB             ;CALL THE LWRSUB SUBROUTINE
1411  4352                            ENDTST TST01X
 (1)                          ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4352                    REQ   7                  ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4352  CD  06  28   18.0                    CALL  REQST
 (2)  4355  00                                   .BYTE                ;DATA PATTERN NUMBER
 (2)  4356  00  00                               .WORD                ;SYSTEM "" COUNT
 (2)  4358  00  00                               .WORD                ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  435A  00                                   .BYTE                ;DATA COMPARE FLAG IF =1
 (2)  435B  07                                   .BYTE  7             ;REQUEST CODE
 (1)  435C  3A  9A  4F   13.0          LDA   ITERA              ;GET ITERATION COUNT
 (1)  435F  3D           4.0          DCR   A                  ;DOWNCOUNT
 (1)  4360  32  9A  4F   13.0          STA   ITERA              ;SAVE COUNT
 (1)  4363  F2  2B  43   10.0          JP    TST01X             ;DO TEST UNTIL TILL = 0
1412
```

```
                                .SBTTL  TEST 2 - LOOP WRITE/READ - TU PORT - GCR - +10% W/R CLOCKS

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*GCR - LOOP WRITE/READ - TU PORT +10% W/R CLOCKS
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*GCR FORMAT WITH +10% WRITE AND READ CLOCKS
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO +10%
                                ;*  :  SET THE READ PATH CLOCK TO +10%
                                ;*  :  CALL SUBROUTINE GCRSUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE=640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL SUBROUTINE GCRSUB
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP2 MICRO TEST 02
                                ;*TUP2 MICRO ERROR 02
                                ;*TUP2-LOOP WRITE/READ - TU PORT - GCR - +10% W/R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
                                S
                                ; ************************************************************************

                                TEST02: TESTX   @2              ;INITIALIZE THE TEST
 (1)  4366  3E  02          7.0                 MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1)  4368  CD  03  28     18.0                 CALL    TSET            ;SETUP THE TEST
                                ;%TUP2-LOOP WRITE/READ - TU PORT - GCR - +10% W/R CLOCKS
```

```
1453                                    ;&ALL MODULES
1454  436B  21  80  00      10.0               LXI     H,128           ;SETUP THE BYTE COUNT
1455  436E  22  78  45      16.0               SHLD    SIZE            ;
1456                                           ;
1457  4371  3E  0A           7.0  TST02X:      MVI     A,@012          ;SET W/R CLOCKS TO +10%
1458  4373  32  06  43      13.0               STA     SPEED           ;
1459  4376  CD  B0  44      18.0               CALL    GCRSUB          ;CALL SUBROUTINE LWRSUB
1460  4379  2A  78  45      16.0               LHLD    SIZE            ;GET THE BYTE COUNT
1461  437C  01  80  00      10.0               LXI     B,128           ;SET UP INCREMENT VALUE
1462  437F  09              10.0               DAD     B               ;INCREMENT THE BYTE COUNT
1463  4380  22  78  45      16.0               SHLD    SIZE            ;SAVE THE NEW BYTE COUNT
1464  4383  7C               4.0               MOV     A,H             ;DO UNTIL THE BYTE COUNT=1
1465  4384  FE  03           7.0               CPI     $3
1466  4386  C2  71  43      10.0               JNZ     TST02X
1467  4389  21  58  04      10.0               LXI     H,1112.                 ;SET UP THE BYTE COUNT
1468  438C  22  78  45      16.0               SHLD    SIZE                    ;
1469  438F  CD  B0  44      18.0               CALL    GCRSUB                  ;CALL THE LWRSUB SUBROUTINE
1470  4392                                     ENDTST  TEST02
 (1)                                     ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4392                                     REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4392  CD  06  28      18.0                                CALL    REQST
 (2)  4395  00                                                  .BYTE           ;DATA PATTERN NUMBER
 (2)  4396  00  00                                              .WORD           ;SYSTEM "" COUNT
 (2)  4398  00  00                                              .WORD           ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  439A  00                                                  .BYTE           ;DATA COMPARE FLAG IF =1
 (2)  439B  07                                                  .BYTE   7       ;REQUEST CODE
 (1)  439C  3A  9A  4F      13.0               LDA     ITERA           ;GET ITERATION COUNT
 (1)  439F  3D               4.0               DCR     A               ;DOWNCOUNT
 (1)  43A0  32  9A  4F      13.0               STA     ITERA           ;SAVE COUNT
 (1)  43A3  F2  66  43      10.0               JP      TEST02          ;DO TEST UNTIL TILL - 0
1471
```

```
                                .SBTTL  TEST 3 - LOOP WRITE/READ - TU PORT - GCR - -20% W -30% R CLOCKS

43A6                            ST
 (1)                            ; ***************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*GCR - LOOP WRITE/READ - TU PORT - -20%W -30%R CLOCKS
43A6                            SD
 (1)                            ; ***************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*GCR FORMAT WITH -20% WRITE CLOCK AND -30% READ CLOCK
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
43A6                            SP
 (1)                            ; ***************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO -20%
                                ;*  :  SET THE READ PATH CLOCK TO -30%
                                ;*  :  CALL SUBROUTINE GCRSUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE=640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL SUBROUTINE GCRSUB
                                ;*ENDTST
43A6                            SE
 (1)                            ; ***************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*TUP2 MICRO TEST 03
                                ;*TUP2 MICRO ERROR 02
                                ;*TUP2-LOOP WRITE/READ - TU PORT - GCR - -20% W -30% R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
43A6                            S
 (1)                            ; ***************************************************************************

43A6                            TEST03: TESTX   @3              ;INITIALIZE THE TEST
 (1)  43A6  3E  03        7.0                   MVI     A,@3             ;DEFINE THE TEST NUMBER
 (1)  43A8  CD  03  28   18.0                   CALL    TSET             ;SETUP THE TEST
                                ;%TUP2-LOOP WRITE/READ - TU PORT - GCR - -20% W -30% R CLOCKS
```

```
1512                                    ;&ALL MODULES
1513  43AB  21  80  00      10.0         LXI     H,128          ;SET UP THE BYTE COUNT
1514  43AE  22  78  45      16.0         SHLD    SIZE           ;
1515                                    ;
1516  43B1  3E  1B           7.0  TST03X: MVI     A,@033         ;SET -20% W AND -30% R CLOCKS
1517  43B3  32  06  43      13.0         STA     SPEED          ;
1518  43B6  CD  B0  44      18.0         CALL    GCRSUB         ;CALL SUBROUTINE LWRSUB
1519  43B9  2A  78  45      16.0         LHLD    SIZE           ;GET THE BYTE COUNT
1520  43BC  01  80  00      10.0         LXI     B,128          ;SET UP THE INCREMENT VALUE
1521  43BF  09              10.0         DAD     B              ;INCREMENT THE BYTE COUNT
1522  43C0  22  78  45      16.0         SHLD    SIZE           ;SAVE THE NEW BYTE COUNT
1523  43C3  7C               4.0         MOV     A,H            ;DO UNTIL THE BYTE COUNT=
1524  43C4  FE  03           7.0         CPI     $3
1525  43C6  C2  B1  43      10.0         JNZ     TST03X
1526  43C9  21  58  04      10.0         LXI     H,1112.                ;SET UP THE BYTE COUNT
1527  43CC  22  78  45      16.0         SHLD    SIZE                   ;
1528  43CF  CD  B0  44      18.0         CALL    GCRSUB                 ;CALL THE LWRSUB SUBROUTINE
1529  43D2                               ENDTST  TEST03
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43D2                               REQ     7              ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43D2  CD  06  28      18.0                         CALL    REQST
 (2)  43D5  00                                           .BYTE                   ;DATA PATTERN NUMBER
 (2)  43D6  00  00                                       .WORD                   ;SYSTEM "" COUNT
 (2)  43D8  00  00                                       .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43DA  00                                           .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  43DB  07                                           .BYTE   7               ;REQUEST CODE
 (1)  43DC  3A  9A  4F      13.0         LDA     ITERA          ;GET ITERATION COUNT
 (1)  43DF  3D               4.0         DCR     A              ;DOWNCOUNT
 (1)  43E0  32  9A  4F      13.0         STA     ITERA          ;SAVE COUNT
 (1)  43E3  F2  A6  43      10.0         JP      TEST03         ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST 4 - LOOP WRITE/READ - TU PORT - PE - NORMAL W/R CLOCKS

                        ST
                        ; ***************************************************************************
                        ;*TEST TITLE
                        ;*---- -----
                        ;*PE - LOOP WRITE/READ - TU PORT - NORMAL W/R CLOCKS
                        SD
                        ; ***************************************************************************
                        ;*DESCRIPTION
                        ;*-----------
                        ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                        ;*PE FORMAT WITH NORMAL WRITE AND READ CLOCKS
                        ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
                        SP
                        ; ***************************************************************************
                        ;*PROCEDURE
                        ;*---------
                        ;*BGNTST
                        ;*  GET USER SELECTED TU PORT
                        ;*  SET THE RECORD SIZE TO 128
                        ;*  BGNDO
                        ;*  :  SET THE WRITE PATH CLOCK TO NORMAL
                        ;*  :  SET THE READ PATH CLOCK TO NORMAL
                        ;*  :  CALL SUBROUTINE PESUB
                        ;*  :  ADD 128 TO THE RECORD SIZE
                        ;*  :  DO UNTIL THE RECORD SIZE=640
                        ;*  ENDDO
                        ;*  SET RECORD SIZE TO 1112.
                        ;*  CALL SUBROUTINE PESUB
                        ;*ENDTST
                        SE
                        ; ***************************************************************************
                        ;*ERRORS
                        ;*------
                        ;*TUP2 MICRO TEST 04
                        ;*TUP2 MICRO ERROR 02
                        ;*TUP2-LOOP WRITE/READ - TU PORT - PE - NORMAL W/R CLOCKS
                        ;*ALL MODULES
                        ;*LOOP WRITE/READ ERROR
                        ;*ACTUAL = AAAA
                        ;*EXPECTED = EEEE
                        ;*BYTE-SCLK COUNT = LLL
                        ;*DATA FORMAT = MM
                        ;*SKIP COUNT = NN
                        ;*ACTUAL DATA IS THE FAILURE CODE
                        ;*EXPECTED DATA IS THE INTERRUPT CODE
                        S
                        ; ***************************************************************************

43E6                    TEST04: TESTX   @4              ;INITIALIZE THE TEST
43E6  3E  04              7.0                   MVI     A,@4            ;DEFINE THE TEST NUMBER
43E8  CD  03  28         18.0                   CALL    TSET            ;SETUP THE TEST
                        ;%TUP2-LOOP WRITE/READ - TU PORT - PE - NORMAL W/R CLOCKS
```

```
1570                                    ;&ALL MODULES
1571  43EB  21  80  00      10.0         LXI     H,128                   ;SET UP THE BYTE COUNT
1572  43EE  22  78  45      16.0         SHLD    SIZE                    ;
1573                                                                    ;
1574  43F1  3E  10            7.0  TST04X: MVI     A,@020                  ;SET UP FOR NORMAL CLOCKS
1575  43F3  32  06  43      13.0         STA     SPEED
1576  43F6  CD  A9  44      18.0         CALL    PESUB           ;CALL SUBROUTINE PESUB
1577                                                                    ;
1578  43F9  2A  78  45      16.0         LHLD    SIZE                    ;GET THE SIZE
1579  43FC  01  80  00      10.0         LXI     B,128                   ;LOAD THE INCREMENT VALUE
1580  43FF  09              10.0         DAD     B                       ;INCREMENT THE BYTE COUNT
1581  4400  22  78  45      16.0         SHLD    SIZE                    ;SAVE THE SIZE FOR LATER
1582  4403  7C               4.0         MOV     A,H                     ;DO UNTIL THE BYTE COUNT = 768
1583  4404  FE  03            7.0         CPI     $3                      ;
1584  4406  C2  F1  43      10.0         JNZ     TST04X                  ;
1585  4409  21  58  04      10.0         LXI     H,1112.                 ;SET UP THE BYTE COUNT
1586  440C  22  78  45      16.0         SHLD    SIZE                    ;
1587  440F  CD  A9  44      18.0         CALL    PESUB                   ;CALL SUBROUTINE PESUB
1588  4412                               ENDTST  TEST04
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4412                               REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4412  CD  06  28      18.0                                 CALL    REQST
 (2)  4415  00                                                   .BYTE                   ;DATA PATTERN NUMBER
 (2)  4416  00  00                                               .WORD                   ;SYSTEM "" COUNT
 (2)  4418  00  00                                               .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  441A  00                                                   .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  441B  07                                                   .BYTE   7               ;REQUEST CODE
 (1)  441C  3A  9A  4F      13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1)  441F  3D               4.0         DCR     A               ;DOWNCOUNT
 (1)  4420  32  9A  4F      13.0         STA     ITERA           ;SAVE COUNT
 (1)  4423  F2  E6  43      10.0         JP      TEST04          ;DO TEST UNTIL TILL = 0
1589
```

```
                                .SBTTL  TEST 5 - LOOP WRITE/READ - TU PORT - PE - +10% W/R CLOCKS

                                ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*PE - LOOP WRITE READ - TU PORT +10% W/R CLOCKS
                                SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*PE FORMAT WITH +10% WRITE AND READ CLOCKS
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
                                SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO +10%
                                ;*  :  SET THE READ PATH CLOCK TO +10%
                                ;*  :  CALL SUBROUTINE PESUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE=640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL SUBROUTINE PESUB
                                ;*ENDTST
                                SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*TUP2 MICRO TEST 05
                                ;*TUP2 MICRO ERROR 02
                                ;*TUP2-LOOP WRITE/READ - TU PORT - PE - +10% W/R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
                                S
                                ; ***************************************************************************

                                TEST05: TESTX   @5              ;INITIALIZE THE TEST
 (1)  4426  3E  05          7.0                 MVI     A,@5            ;DEFINE THE TEST NUMBER
 (1)  4428  CD  03  28     18.0                 CALL    TSET            ;SETUP THE TEST
                                ;%TUP2-LOOP WRITE/READ - TU PORT - PE - +10% W/R CLOCKS
```

```
1630                                    ;&ALL MODULES
1631  442B  21  80  00      10.0           LXI     H,128           ;SETUP THE BYTE COUNT
1632  442E  22  78  45      16.0           SHLD    SIZE            ;
1633                                                             ;
1634  4431  3E  0A           7.0  TST05X:  MVI     A,@012          ;SET W/R CLOCKS TO +10%
1635  4433  32  06  43      13.0           STA     SPEED           ;
1636  4436  CD  A9  44      18.0           CALL    PESUB           ;CALL SUBROUTINE PESUB
1637  4439  2A  78  45      16.0           LHLD    SIZE            ;GET THE BYTE COUNT
1638  443C  01  80  00      10.0           LXI     B,128           ;SET UP INCREMENT VALUE
1639  443F  09              10.0           DAD     B               ;INCREMENT THE BYTE COUNT
1640  4440  22  78  45      16.0           SHLD    SIZE            ;SAVE THE NEW BYTE COUNT
1641  4443  7C               4.0           MOV     A,H             ;DO UNTIL THE BYTE COUNT=1
1642  4444  FE  03           7.0           CPI     $3
1643  4446  C2  31  44      10.0           JNZ     TST05X
1644  4449  21  58  04      10.0           LXI     H,1112.                 ;SET UP THE BYTE COUNT
1645  444C  22  78  45      16.0           SHLD    SIZE                    ;
1646  444F  CD  A9  44      18.0           CALL    PESUB                   ;CALL SUBROUTINE PESUB
1647  4452                                  ENDTST  TEST05
 (1)                                  ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4452                                REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4452  CD  06  28      18.0                            CALL    REQST
 (2)  4455  00                                              .BYTE           ;DATA PATTERN NUMBER
 (2)  4456  00  00                                          .WORD           ;SYSTEM "" COUNT
 (2)  4458  00  00                                          .WORD           ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  445A  00                                              .BYTE           ;DATA COMPARE FLAG IF =1
 (2)  445B  07                                              .BYTE   7       ;REQUEST CODE
 (1)  445C  3A  9A  4F      13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1)  445F  3D               4.0           DCR     A               ;DOWNCOUNT
 (1)  4460  32  9A  4F      13.0           STA     ITERA           ;SAVE COUNT
 (1)  4463  F2  26  44      10.0           JP      TEST05          ;DO TEST UNTIL TILL - 0
1648
```

```
1650                                    .SBTTL  TEST 6 - LOOP WRITE/READ - TU PORT - PE - -20% W -30% R CLOCKS
1651
1652  4466                              ST
 (1)                                    ; ************************************************************************
 (1)                                    ;*TEST TITLE
 (1)                                    ;*---- -----
1653                                    ;*PE - LOOP WRITE/READ - TU PORT - -20%W -30%R CLOCKS
1654  4466                              SD
 (1)                                    ; ************************************************************************
 (1)                                    ;*DESCRIPTION
 (1)                                    ;*-----------
1655                                    ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
1656                                    ;*PE FORMAT WITH -20% WRITE CLOCK AND -30% READ CLOCK
1657                                    ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
1658  4466                              SP
 (1)                                    ; ************************************************************************
 (1)                                    ;*PROCEDURE
 (1)                                    ;*---------
1659                                    ;*BGNTST
1660                                    ;*  GET USER SELECTED TU PORT
1661                                    ;*  SET THE RECORD SIZE TO 128
1662                                    ;*  BGNDO
1663                                    ;*  :  SET THE WRITE PATH CLOCK TO -20%
1664                                    ;*  :  SET THE READ PATH CLOCK TO -30%
1665                                    ;*  :  CALL SUBROUTINE PESUB
1666                                    ;*  :  ADD 128 TO THE RECORD SIZE
1667                                    ;*  :  DO UNTIL THE RECORD SIZE=640
1668                                    ;*  ENDDO
1669                                    ;*  SET RECORD SIZE TO 1112.
1670                                    ;*  CALL SUBROUTINE PESUB
1671                                    ;*ENDTST
1672  4466                              SE
 (1)                                    ; ************************************************************************
 (1)                                    ;*ERRORS
 (1)                                    ;*------
1673                                    ;*TUP2 MICRO TEST 06
1674                                    ;*TUP2 MICRO ERROR 02
1675                                    ;*TUP2-LOOP WRITE/READ - TU PORT - PE - -20% W -30% R CLOCKS
1676                                    ;*ALL MODULES
1677                                    ;*LOOP WRITE/READ ERROR
1678                                    ;*ACTUAL = AAAA
1679                                    ;*EXPECTED = EEEE
1680                                    ;*BYTE-SCLK COUNT = LLL
1681                                    ;*DATA FORMAT = MM
1682                                    ;*SKIP COUNT = NN
1683                                    ;*ACTUAL DATA IS THE FAILURE CODE
1684                                    ;*EXPECTED DATA IS THE INTERRUPT CODE
1685  4466                              S
 (1)                                    ; ************************************************************************
1686
1687  4466                              TEST06: TESTX   @6                ;INITIALIZE THE TEST
 (1)  4466  3E  06            7.0                 MVI     A,@6             ;DEFINE THE TEST NUMBER
 (1)  4468  CD  03  28       18.0                 CALL    TSET             ;SETUP THE TEST
1688                                    ;%TUP2-LOOP WRITE/READ - TU PORT - PE - -20% W -30% R CLOCKS
```

```
1689                                    ;&ALL MODULES
1690  446B  21  80  00      10.0           LXI     H,128           ;SET UP THE BYTE COUNT
1691  446E  22  78  45      16.0           SHLD    SIZE            ;
1692                                                             ;
1693  4471  3E  1B           7.0  TST06X:  MVI     A,@033          ;SET -20% W AND -30% R CLOCKS
1694  4473  32  06  43      13.0           STA     SPEED           ;
1695  4476  CD  A9  44      18.0           CALL    PESUB           ;CALL SUBROUTINE PESUB
1696  4479  2A  78  45      16.0           LHLD    SIZE            ;GET THE BYTE COUNT
1697  447C  01  80  00      10.0           LXI     B,128           ;SET UP THE INCREMENT VALUE
1698  447F  09              10.0           DAD     B               ;INCREMENT THE BYTE COUNT
1699  4480  22  78  45      16.0           SHLD    SIZE            ;SAVE THE NEW BYTE COUNT
1700  4483  7C               4.0           MOV     A,H             ;DO UNTIL THE BYTE COUNT=
170?  4484  FE  03           7.0           CPI     $3
170?  4486  C2  71  44      10.0           JNZ     TST06X
1703  4489  21  58  04      10.0           LXI     H,1112.                 ;SET UP THE BYTE COUNT
1704  448C  22  78  45      16.0           SHLD    SIZE                    ;
1705  448F  CD  A9  44      18.0           CALL    PESUB                   ;CALL SUBROUTINE PESUB
1706  4492                                 ENDTST  TEST06
 (1)                                 ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4492                           REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4492  CD  06  28      18.0                    CALL    REQST
 (2)  4495  00                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  4496  00  00                                  .WORD                   ;SYSTEM "" COUNT
 (2)  4498  00  00                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  449A  00                                      .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  449B  07                                      .BYTE   7               ;REQUEST CODE
 (1)  449C  3A  9A  4F      13.0           LDA     ITERA           ;GET ITERATION COUNT
 (1)  449F  3D               4.0           DCR     A               ;DOWNCOUNT
 (1)  44A0  32  9A  4F      13.0           STA     ITERA           ;SAVE COUNT
 (1)  44A3  F2  66  44      10.0           JP      TEST06          ;DO TEST UNTIL TILL = 0
1707  44A6  C3  18  28      10.0  EXIT:    JMP     TSTEND
```

```
                                        .SBTTL  SUBROUTINE PESUB

                                PESUB:  XRA     A               ;CLEAR THE ACCUMULATOR
                                        STA     PGCTL           ;SET THE PE FLAG
                                        JMP     PEGCOM

                                GCRSUB: MVI     A,@377          ;SET THE GCR FLAG
                                        STA     PGCTL

                                PEGCOM: MVI     B,@0
                                        LDA     UNITMP          ;DID THE USER SPECIFY TU
                                        ANI     @01             ;PORT 0?
                                        JNZ     FOUND           ;YES-GO USE PORT #0
                                        INR     B               ;NO-UPDATE POINTER TO PORT #1
                                        LDA     UNITMP          ;DID THE USER SPECIFY TU
                                        ANI     @02             ;PORT 1?
                                        JNZ     FOUND           ;YES-GO USE PORT #1
                                        INR     B               ;NO-UPDATE POINTER TO PORT #2
                                        LDA     UNITMP          ;DID THE USER SPECIFY TU
                                        ANI     @04             ;PORT 2?
                                        JNZ     FOUND           ;YES-GO USE PORT #2
                                        INR     B               ;NO-ASSUME PORT #3
                                FOUND:  CALL    CLEAR
                                        IN      INTSTA
                                        ANI     BIT7
                                        ORA     B
                                        OUT     MBSEL           ;SELECT THE DESIRED TU-PORT AND MASSBUS

                                T1L01:  XRA     A               ;CLEAR THE DATA BUS CONTROL
                                        STA     DBUS            ;MASK TO KEEP THE MASSBUS CLEAN
                                                                ;
                                        LHLD    SIZE            ;GET THE RECORD SIZE
                                        SHLD    BYTEL           ;TO 128 BYTES
                                        MOV     A,L             ;GET THE BYTE COUNT LOW
                                        ROUT    RO5L            ;SAVE IN BYTE COUNT LOW
 (1)                                            OUT     RO5L            ;WRITE AC INTO RO5L
 (1)                                            MOV     A,A             ;RETRY LINK
                                        MOV     A,H             ;GET THE BYTE COUNT HIGH
                                        ROUT    RO5H            ;SAVE IN BYTE COUNT HIGH
 (1)                                            OUT     RO5H            ;WRITE AC INTO RO5H
 (1)                                            MOV     A,A             ;RETRY LINK
                                                                ;
                                        LDA     FORMT           ;GET THE FORMAT
                                        STA     FORMAT          ;NO SKIP COUNT
                                        ROUT    RO2H            ;SAVE IN THE CAS
 (1)                                            OUT     RO2H            ;WRITE AC INTO RO2H
 (1)                                            MOV     A,A             ;RETRY LINK
                                                                ;
                                        MVI     A,DDRCO         ;CLEAR DDR IN, SO DDR
                                        OUT     DDRCTL          ;WRITES TO TAPE
                                                                ;
                                        LHLD    KWDAT           ;GET THE WRITE DATA
                                        DAD     H               ;SHIFT IT LEFT 1 BIT POSITION
                                        RAL                     ;SAVE THE CARRY
```

```
1757  4500  29              10.0          DAD     H               ;SHIFT IT LEFT ANOTHER BIT POSITION
1758  4501  17               4.0          RAL                     ;SAVE THAT CARRY TOO.
1759  4502  E6  03           7.0          ANI     $3              ;SAVE ONLY THE CARRY BITS
1760  4504  47               4.0          MOV     B,A             ;SAVE IN REGISTER B
1761  4505  7C               4.0          MOV     A,H             ;
1762  4506  D3  D9          10.0          OUT     DDRB            ;LOAD <13:6> IN DDR B
1763  4508  3A  7D  45      13.0          LDA     KWDAT+2         ;GET BITS <17:16>
1764  450B  85               4.0          ADD     L               ;MERGE WITH <5:0>
1765  450C  D3  D8          10.0          OUT     DDRA            ;LOAD INTO DDR A
1766  450E  AC               4.0          XRA     H               ;COMPUTE PARITY ON ALL BITS
1767  450F  A8               4.0          XRA     B               ;
1768  4510  78               4.0          MOV     A,B             ;GET BITS <15:14> AGAIN
1769  4511  E2  16  45      10.0          JPO     1$              ;JUMP IF ODD PARITY
1770  4514  F6  04           7.0          ORI     BIT2            ;SET THE PARITY BIT TO MAKE IT ODD
1771  4516  D3  DA          10.0  1$:     OUT     DDRC            ;LOAD <P:14> IN DDRC
1772                                                              ;
1773  4518  21  03  43      10.0          LXI     H,SPEED-3       ;POINT TO THE CLOCK
1774  451B  22  26  4F      16.0          SHLD    CLOCK           ;SPEED TABLE
1775  451E  3A  76  45      13.0          LDA     PGCTL           ;GET THE PE/GCR FLAG
1776  4521  A7               4.0          ANA     A               ;SET CONDITION BITS
1777  4522  C2  2B  45      10.0          JNZ     2$              ;GO DO GCR
1778  4525  CD  4F  00      18.0          CALL    LKLWPP          ;ELSE-DO PE
1779  4528  C3  2E  45      10.0          JMP     3$              ;
1780  452B  CD  52  00      18.0  2$:     CALL    LKLWPG          ;CALL LOOP WRITE/READ - TU PORT - GCR
1781                                                              ;
1782  452E  D2  44  45      10.0  3$:     JNC     T1CN1                   ;CONTINUE IF NO ERROR
1783  4531  78               4.0          MOV     A,B                     ;GET THE FAILURE CODE
1784  4532  E6  FC           7.0          ANI     $FC                     ;MASK OUT THE MEANINGFUL BITS
1785  4534  0F               4.0          RRC                             ;POSITION FOR OUTPUT
1786  4535  0F               4.0          RRC                             ;
1787  4536                               ROUT    ADATA                   ;STORE IN THE CAS
 (1)  4536  D3  94          10.0                  OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4538  7F               4.0                  MOV     A,A             ;RETRY LINK
1788  4539  79               4.0          MOV     A,C                     ;GET THE INTERRUPT CODE
1789  453A  E6  3F           7.0          ANI     $3F                     ;MASK OUT THE MEANINGFUL BITS
1790  453C                               ROUT    EDATA                   ;STORE IN THE CAS
 (1)  453C  D3  95          10.0                  OUT     EDATA           ;WRITE AC INTO EDATA
 (1)  453E  7F               4.0                  MOV     A,A             ;RETRY LINK
1791  453F                               ERRB    T1LO1,T1CN1,3
 (1)                                   ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  453F  CD  12  28      18.0                  CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)       0002                                   MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4542  02                                    .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4543  03                                    .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  4544  CD  15  28      18.0          T1CN1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4547  DA  DC  44      10.0                  JC      T1LO1           ;LOOP ADDRESS IF LOOP SPECIFIED
1792                                ;>LOOP WRITE/READ ERROR
1793                                ;<ACTUAL DATA IS THE FAILURE CODE
1794                                ;<EXPECTED DATA IS THE INTERRUPT CODE
1795  454A  C9              10.0          RET
```

```
                                .SBTTL  SUBROUTINE CLEAR ALL TU PORTS
                                SSUB
                                ; ************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*CLEAR ALL TU PORTS
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                ;*SUBSYSTEM AND TO CLEAR ALL TU PORT PENDING INTERRUPTS, AMTIE LINES
                                ;*AND LOOP MODES.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*     SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                ;*     CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                ;*     CLEAR PORT SELECT FOR TRANSPORT
                                ;*     CLEAR PORT PARITY ERRORS & ENABLE WORD
                                ;*     CLEAR PORT DIAGNOSTIC CONTROL
                                ;*     CLEAR PORT AMTIE WORD
                                ;*ENDSUB
                                S
                                ; ************************************************************************

454B  F5              12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
454C  C5              12.0          PUSH    B
454D  06  00           7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
454F  DB  E0          10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
4551  E6  80           7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
4553  B0               4.0          ORA     B               ;ADD IN THE SELECTED PORT #
4554  D3  E0          10.0          OUT     MBSEL           ;RESET TO THIS PORT
4556  3E  80           7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
4558  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
455A  AF               4.0          XRA     A               ;CLEAR TU COMMAND A
455B  D3  40          10.0          OUT     TCMD            ;
455D  3E  81           7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
455F  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
4561  3E  00           7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
4563  D3  40          10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
4565  AF               4.0          XRA     A
4566  D3  44          10.0          OUT     TAMT            ;CLEAR AMTIE WORD
4568  D3  48          10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
456A  D3  4C          10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
456C  04               4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
456D  78               4.0          MOV     A,B
456E  FE  04           7.0          CPI     4               ;DONE?
4570  C2  4F  45      10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
4573  C1              10.0          POP     B               ;RESET B & C
4574  F1              10.0          POP     PSW             ;ALL DONE
4575  C9              10.0          RET                     ;EXIT
```

```
1842                                    .SBTTL  PROGRAM VARIABLES
1843
1844  4576   00                         PGCTL:  .BYTE   0               ;PE/GCR CONTROL FLAG
1845  4577   00                         UNITMP: .BYTE   0               ;UNIT MAP
1846  4578   00    00                   SIZE:   .WORD   0               ;RECORD SIZE
1847  457A   00                         FORMT:  .BYTE   0               ;DATA FORMAT
1848  457B   00                         KWDAT:  .BYTE   0               ;18 BITS OF INPUT DATA
1849  457C   00                                 .BYTE   0
1850  457D   00                                 .BYTE   0
1851        0000                                .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE    4F9B          ATTCD    4F97         AXNUM    4F91
B      =%0000          BADST  = 0090          BIT0   = 0001         BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008         BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080         BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F0C           BSAVE    4F9C          BYTCNT= 00D4          BYTEH    4F24
BYTEL    4F23          C      =%0001          CASCT    4F21         CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088         CBUSST= 00A1
CBYTH  = 008B          CBYTL  = 008A          CDG1H  = 0087         CDG1L  = 0086
CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095         CDG3L  = 0094
CDVTH  = 008D          CDVTL  = 008C          CHPTIE= 0028          CHOTIE= 0020
CH1TIE= 0021           CH2TIE= 0022           CH3TIE= 0023          CH4TIE= 0024
CH5TIE= 0025           CH6TIE= 0026           CH7TIE= 0027          CKLOP  = 2815
CLEAR    454B          CLKCTL= 00F0           CLOCK    4F26         CLRLP    454F
CMC0H  = 0099          CMC0L  = 0098          CMC1H  = 009B         CMC1L  = 009A
CMC2H  = 009D          CMC2L  = 009C          CMC3H  = 009F         CMC3L  = 009E
CMINH  = 0097          CMINL  = 0096          CNTCTL= 00D7          CRCWRD= 0018
CSAVE    4F9D          CSRLH  = 0091          CSRLL  = 0090         CTCH   = 0085
CTCL   = 0084          CTSTH  = 008F          CTSTL  = 008E         CXCTH  = 0081
CXCTL  = 0080          CXINH  = 0083          CXINL  = 0082         C.     = 0001
C.AVAI= 0080           C.DP   = 0008          C.DSE  = 0010         C.DTU  = 0003
C.DVA  = 0008          C.FAIL= 00FC           C.FMT  = 0070         C.FNCT= 003E
C.GO   = 0001          C.INTC= 00FE           C.MAIN= 0020          C.NSA  = 0080
C.RCI  = 00FC          C.SER  = 0080          C.SHR  = 0040         C.SKPC= 000F
C.TAPE= 0040           C.WCS  = 0002          D      =%0002         DATACT= 00D0
DBUS     4F28          DBUSCT= 00C0           DBUSST= 00C0          DDRA   = 00D8
DDRAIN= 0010           DDRB   = 00D9          DDRBIN= 0002          DDRC   = 00DA
DDRCIN= 0001           DDRCO  = 0088          DDRCTL= 00DB          DIAFLG   4F22
DIAGPG= 4300           DIAGRM= 4F90           DIARD  = 000B         DONE1  = 0045
DONINT= 0010           DSAVE    4F9E          DSE    = 0006         DUMMY    4325   G
D.ATH0= 0001           D.ATH1= 0002           D.EOTD= 0010          D.LAGC= 0020
D.NOIW= 0040           D.NTHR= 0004           D.TACH= 0008          D.WR4  = 0080
E      =%0003          ECCBAD= 0042           ECCCOR= 0019          ECCOK  = 0041
ECCSTA= 001A           ECCTST= 000E           EDATA  = 0095         EOTCLR= 0003
ERFLG    4F93          ERLP   = 2809          ERLPA  = 280F         ERLPB  = 2812
ERLPE  = 280C          ERNUM    4F90          ERRCNT= 00D6          ESAVE    4F9F
EXIT     44A6          E.ACRC= 0010           E.AMT  = 0020         E.CDP  = 0080
E.CRC  = 0080          E.PNTR= 0008           E.RPE  = 0040         E.STEC= 0001
E.TTEC= 0002           E.UNC  = 0004          FIFORD= 006A          FORMAT   4F25
FORMT    457A          FOUND    44D2          FWDTST= 0061          GCRID  = 0089
GCRSET= 0002           GCRSUB   44B0          GOODTM= 0092          H      =%0004
HLSAVE   4FA0          IE     = 0008          INTSTA= 00E0          ITERA    4F9A
I.PWR  = 0020          I.RMPE= 0040           I5.5   = 0010         I6.5   = 0020
I7.5   = 0040          KCALL  = 005F          KCLR   = 007B         KDEP   = 003F
KENAB  = 0078          KEXAM  = 003E          KEYBRD= 00C8          KEY1   = 0078
KEY10  = 006D          KEY11  = 006E          KEY12  = 006F         KEY13  = 005C
KEY14  = 005D          KEY15  = 005E          KEY16  = 005F         KEY17  = 003C
KEY18  = 003D          KEY19  = 003E          KEY2   = 0079         KEY20  = 003F
KEY3   = 007A          KEY4   = 007B          KEY5   = 0074         KEY6   = 0075
KEY7   = 0076          KEY8   = 0077          KEY9   = 006C         KINTA  = 006F
KLDAD  = 003D          KN0    = 003C          KN1    = 005C         KN2    = 005D
KN3    = 005E          KN4    = 006C          KN5    = 006D         KN6    = 006E
KN7    = 0074          KN8    = 0075          KN9    = 0076         KU2    = 0079
```

```
KU3    = 007A          KU8    = 0077          KWDAT   457B          L      =%0005
LBLANK= 000F           LCE    = 000B          LCH    = 000C          LCL    = 000D
LCP    = 000E          LC0    = 0000          LC1    = 0001          LC2    = 0002
LC3    = 0003          LC4    = 0004          LC5    = 0005          LC6    = 0006
LC7    = 0007          LC8    = 0008          LC9    = 0009          LDLEDA= 00CA
LDLEDB= 00CB           LDLEDC= 00CC           LDLEDD= 00CD           LDLEDE= 00CE
LDLEDF= 00CF           LKDIAG= 2800           LKKBD  = 004C          LKKEY  = 0049
LKLWMG= 0058           LKLWMP= 005F           LKLWPG= 0052           LKLWPP= 004F
LKMOD7= 0046           LKOPR  = 0046          LPFLG   4F94          LPNUM   4F92
M      =%0006          MBSEL  = 00E0          MB.A   = 0008          MB.B   = 0004
MEMTOP= 4FFF           MINUS  = 000A          MM     = 8000          MSE    = 0008
MSGN   = 0002          MTACLR= 0000           MT.ARA= 0020           MT.CPE= 0080
MT.DSE= 0001           MT.FWD= 0040           MT.INH= 0008           MT.LWR= 0004
MT.MOT= 0002           MT.NWT= 0080           MT.PEC= 0040           MT.PSB= 0004
MT.PS0= 0001           MT.PS1= 0002           MT.REV= 0020           MT.WRT= 0010
MT.Z   = 0008          M.ATA  = 0080          M.CAPE= 0020           M.CONT= 0080
M.DEM  = 0020          M.EBL  = 0010          M.EXC  = 0008          M.FAIL= 0008
M.ILR  = 0010          M.INIT= 0010           M.OCC  = 0020          M.ONLI= 0001
M.PE   = 0040          M.PORT= 0080           M.RDEN= 0002           M.RDPE= 0008
M.RUN  = 0004          M.SCLK= 0001           M.TRA  = 0040          M.UNIT= 0007
M.WCLK= 0040           M.WCLN= 0080           M.WREN= 0080           M5.5   = 0001
M6.5   = 0002          M7.5   = 0004          NOTCAP= 0088           OKAY   = 00FF
OPRRAM= 4300           OPSTRT= 0058           OPVER  = 0040          PADCNT= 00D5
PADCRC= 0080           PDIAG  = 0048          PEGCOM  44B5          PEID   = 008A
PENAB  = 004C          PESET  = 0001          PESUB   44A9          PGCTL   4576
PL     = 00B1          PRDD   = 004C          PRENF  = 009C          PS     = 00B2
PSTAT  = 0048          PSW    =%0009          P.AMTP= 0001           P.BCTC= 0040
P.CMDP= 0020           P.INTE= 0080           P.LCS  = 0040          P.LWR  = 0020
P.RDP  = 0002          P.RPEN= 0004           P.RPST= 0002           P.RPOE= 0020
P.RP1E= 0010           P.RP2E= 0020           P.RP3E= 0010           P.SING= 0080
P.STAT= 0002           P.STPE= 0080           P.TACH= 0008           P.TUPR= 0010
P.WCSP= 0004           P.WDS  = 0040          P.WFLP= 0001           P.WPE'= 0010
P.WPUE= 0008           P.WP1E= 0004           P.WP2E= 0008           P.WP3E= 0004
P.5VOK= 0002           QUE    = 281B          QUEM   = 281E          RAMT   = 0010
RARA   = 0006          RARAI  = 0004          RCHBD0= 0048           RCHBD1= 0047
RCHOK  = 0046          RCHTST= 000C           RCLRT  = 000D          RCMD   = 000B
RCMLP  = 0003          RCONT  = 0080          RDATA  = 0017          RDCLK  = 0010
RDON   = 0011          READG  = 0007          REND   = 0014          REQST  = 2806
RESCHR= 00D1           REVTST= 0064           REWIND= 0004           RFIFOL= 0008
RGCLK  = 0002          RGCRI  = 0003          RIBG   = 0001          RILL   = 0012
RINST  = 000C          RMCTST= 0008           RMK2   = 0010          RNOP   = 0000
RPATH  = 0001          RPBAD  = 0044          RPCHI  = 0001          RPCLK  = 0003
RPCTL  = 0009          RPEI   = 0002          RPFAIL= 0000           RPF1   = 009D
RPF2   = 009E          RPOK   = 0043          RPOSTN= 0016           RPSTA  = 0015
RRCMT  = 000A          RSTAT  = 0002          RTIEB  = 000A          RTIER  = 0030
RTM    = 0005          RUNKI  = 0009          RUPTST= 005E           RWDUNL= 0005
R.AMT  = 0001          R.BOP  = 0008          R.DATA= 0040           R.DON  = 0002
R.DRDY= 0010           R.END  = 0010          R.ILL  = 0004          R.JVOK= 0004
R.MK2  = 0008          R.PLOD= 0008           R.PLO0= 0010           R.PLO1= 0020
R.POST= 0020           R.STNM= 0002           R.STOP= 0004           R.STPC= 0001
R.TBJN= 0080           R.TSTD= 0040           R.VOK  = 0080          R00H   = 0081
R00L   = 0080          R01H   = 0083          R01L   = 0082          R02H   = 0085
R02L   = 0084          R03H   = 0087          R03L   = 0086          R04H   = 0089
R04L   = 0088          R05H   = 008B          R05L   = 008A          R06H   = 008D
```

```
R06L  = 008C          R07H  = 008F          R07L  = 008E          R10H  = 0091
R10L  = 0090          R11H  = 0093          R11L  = 0092          R12H  = 0095
R12L  = 0094          R13H  = 0097          R13L  = 0096          R14H  = 0099
R14L  = 0098          R15H  = 009B          R15L  = 009A          R16H  = 009D
R16L  = 009C          R17H  = 009F          R17L  = 009E          R7.5  = 0010
SELCLR= 0000          SETATA= 00A1          SID   = 0080          SIZE    4578
SOD   = 0080          SOE   = 0040          SP    =%0008          SPEED   4306
SSCLK = 0040          SSTEP = 0005          STACK = 4FFF          STATRM= 4F20
STPCT   4F20          STRSP = 5000          TADR00= 0080          TADR01= 0081
TADR02= 0082          TADR03= 0083          TADR04= 0084          TADR05= 0085
TADR06= 0086          TADR07= 0087          TADR10= 0088          TADR11= 0089
TADR12= 008A          TADR13= 008B          TAMT  = 0044          TASEL = 0080
TCMD  = 0040          TC.FWD= 0040          TC.INH= 0008          TC.LWR= 0004
TC.REV= 0020          TC.WRT= 0010          TEMP    4F99          TEST02  4366
TEST03  43A6          TEST04  43E6          TEST05  4426          TEST06  4466
TEST1   4307          TMF   = 0099          TMRDY = 0040          TRKENA= 00D2
TSET  = 2803          TSTEND= 2818          TSTS  = 0040          TST01X  432B
TST01Y  4331          TST02X  4371          TST03X  43B1          TST04X  43F1
TST05X  4431          TST06X  4471          TUSEL0= 00D1          TUSEL1= 00D2
TU78  = 0010          T.ATH0= 0001          T.ATH1= 0002          T.BOT = 0004
T.EOT = 0002          T.FPT = 0001          T.NTHR= 0004          T.ONL = 0020
T.PES = 0008          T.PSBJ= 0020          T.PSOJ= 0008          T.PS1J= 0010
T.RDY = 0080          T.RDYO= 0040          T.RWD = 0010          T.SCLK= 0002
T1CN1   4544  G       T1LO1   44DC          UIBG  = 00A1          UNITMP  4577
VALFC   4F98          VALTB   4F95          VELTST= 005B          WDR.P = 0010
WMCCTL= 00D3          WMCERR= 00DA          WMCSTA= 00D0          WRTCLK= 0000
WRTDAT= 00D3          W.ACRC= 0004          W.CRC = 0008          W.DIAG= 0002
W.DONN= 0040          W.ECC = 0010          W.ENAB= 0080          W.ERR   0020
W.FMT = 0070          W.GCR = 0010          W.LEFT= 0004          W.ONES= 0020
W.RESI= 0002          W.REV = 0004          W.ROME= 0010          W.RST = 0001
W.SKIP= 000F          W.WRIT= 0008          W.XFER= 0020          X     =%000A
X.DONN= 0080          X.ENAB= 0040          X.PEPE= 0002          X.ROME= 0001
X.WCLK= 0001          Y     =%000B          .     = 457E


 ERRORS DETECTED:  0

 *TUP2.A78/PTP,TUP2=NLIST,PARAM,MACRO,LIST,TUP2
 RUN-TIME: 4 4 0 SECONDS
 CORE USED:  10K
```

3	"MEMORY MAP" EXTENDED ADDRESS BIT
19	READ MICRO CONTROLLER CONTROL & STATUS WORDS
139	TRANSPORT/PORT COMMAND + STATUS WORDS
380	COMMON ADDRESS SPACE REGISTERS
444	MASSBUS REGISTER ASSIGNMENTS FOR DIAGNOSTIC USE
485	MASSBUS CONTROL WORD DEFINITIONS
557	WRITE MICRO CONTROLLER CONTROL + STATUS WORDS
709	KEYBOARD/DISPLAY CONTROL + STATUS WORD
797	MISCELLANEOUS VARIABLE DEFINITIONS
801	RAM ADDRESS DEFINITIONS
819	LINKAGE TABLE ADDRESSES
842	BIT DEFINITIONS
856	RIM AND SIM INSTRUCTION BIT DEFINITIONS
877	INTERRUPTABLE INSTRUCTION MACROS
924	DIAGNOSTIC VARIABLES
945	COMMON VARIABLE STORAGE
959	MACRO CALL DEFINITIONS
1047	TM78 MICRO-DIAGNOSTIC MACRO CALLS
1101	8080 MACRO CALLS - ERROR AND TEST INITIALIZATION
1334	TEST 1 - LOOP WRITE/READ - MTA - GCR - NORMAL W/R CLOCKS
1414	TEST 2 - LOOP WRITE/READ - MTA - GCR - +10% W/R CLOCKS
1473	TEST 3 - LOOP WRITE/READ - MTA - GCR - -20% W -30% R CLOCKS
1531	TEST 4 - LOOP WRITE/READ - MTA - PE - NORMAL W/R CLOCKS
1591	TEST 5 - LOOP WRITE/READ - MTA - PE - +10% W/R CLOCKS
1650	TEST 6 - LOOP WRITE/READ - MTA - PE - -20% W -30% R CLOCKS
1708	TEST7 - MTA VELOCITY TESTS
1840	SUBROUTINE PESUB
1935	SUBROUTINE CLEAR ALL MTAS
1980	PROGRAM VARIABLES

```
                ; ************************************************************************
                ;PRINT ROUTINES DEFINED - USED IN THE "ERR", "ERRE", "ERRA", OR "ERRB"
                ;ERROR MACRO CALLS
                ;
                ;       1  -    REQUEST HOST CPU TO PRINT:
                ;               "BYTE/SCLK COUNT NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;       2  -    REQUEST HOST CPU TO PRINT:
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               MM = DATA FORMAT FROM CAS REGISTER 2
                ;               NN = SKIP COUNT FROM CAS REGISTER 2
                ;
                ;       3  -    REQUEST HOST CPU TO PRINT:
                ;               BYTE-SCLK COUNT = LLL
                ;               DATA FORMAT = MM
                ;               SKIP COUNT  = NN
                ;               WHERE:
                ;               LLL = AS ABOVE
                ;               MM  = AS ABOVE
                ;               NN  = AS ABOVE
                ;
                ;       4  -    REQUEST HOST TO PRINT:
                ;               TRANSITION COUNT = LLL
                ;               WHERE: LLL = COUNT FROM CAS REGISTER 05
                ;
                ;       5  -    REQUEST HOST CPU TO PRINT:
                ;               EXPECTED 18 BITS =      E EEEEEE
                ;               ACTUAL 18 BITS =        A AAAAAA
                ;
                ;               WHERE EXPECTED BITS 15-0 ARE IN CAS REG 14 AND
                ;               BITS 17-16 ARE IN REG 15...PRINT FLAG IS SIGN BIT
                ;               OF CAS REG 15 LOW BYTE.
                ;
                ;               ACTUAL 18 BITS DATA CONTAINED IN REG 16 (BITS 15-0)
                ;               AND IN REG 17 (BITS 17-16)...PRINT FLAG IS CAS REG
                ;               17 LOW BYTE SIGN BIT.
                ;
                ;       6  -    REQUEST THE HOST CPU TO PRINT BOTH ROUTINE #1 AND #5
                ;               TO REPORT THE BYTE/SCLK COUNT AND 18 BITS OF EXPECTED
                ;               AND/OR ACTUAL DATA.
                ;
                ;       7  -    REQUEST HOST CPU TO PRINT:
                ;               "SUBGROUP NUMBER = LLL"
                ;               WHERE:
                ;               LLL = THE VALUE STORED IN CAS REGISTER 5
                ;                     (THE BYTE COUNT REGISTER 16 BITS).
                ;
                ;      10  -    REQUEST THE HOST TO PRINT ROUTINES #1, #2, & #5
                ; ************************************************************************
```

```
                        ;*****************************************************************
                        ;DIAGNOSTIC HOST REQUEST CODES - USED BY THE "REQ" MACRO CALL
                        ;TO CAUSE SOME HOST CPU ACTION.

                        ;       1  -    WRITE XFR PE MODE - ACCEPT HOST RESPONSE CODE 61 OR 65
                        ;       2  -    WRITE XFR GCR MODE - ACCEPT HOST RESPONSE CODE 63 OR 67
                        ;       3  -    READ XFR - FORWARD DIRECTION - HOST RESPONSE CODE 71
                        ;       4  -    READ XFR - REVERSE DIRECTION - HOST RESPONSE CODE 77
                        ;       5  -    REQUEST A HOST END STATUS CHECK - NO ERROR EXPECTED
                        ;               - HOST RESPONSE CODE 31 OR 33
                        ;       6  -    REQUEST A HOST END STATUS CHECK - EXPECT AN ERROR CONDITION
                        ;               - HOST RESPONSE CODE 31 OR 33
                        ;       7  -    REQUEST PORT TEST MASK INPUT FROM HOST CPU
                        ;               HOST LOADS CAS REGISTER 12 LOW BYTE WITH FOLLOWING INFO:
                        ;                       BIT0 = 1 TEST PORT 0
                        ;                       BIT1 = 1 TEST PORT 1
                        ;                       BIT2 = 1 TEST PORT 2
                        ;                       BIT3 = 1 TEST PORT 3
                        ;*****************************************************************
                        ;*****************************************************************
                        ;DATA PATTERN CODES

                        ;       1  -    MBDPAR (MASSBUS DATA PARITY) - 37 18 BIT WORDS
                        ;               FLOAT A 1 THRU ALL 18 POSITIONS STARTING AT BIT 0
                        ;               FLOAT A 0 THRU ALL 18 POSITIONS STARTING AT BIT 18
                        ;               18 BITS OF ALL 1'S
                        ;               18 BITS OF ALL 0'S
                        ;               18 BITS OF ALTERNATING BIT PATTERN (252525)
                        ;               18 BITS OF COMPLIMENT ALT. BIT DATA (525252)
                        ;*****************************************************************
  4300                          .=      DIAGPG  ;START OF ALL DIAGNOSTIC TESTING
```

```
                                .TITLE 'MTA4 - MAG TAPE ADAPTER TEST PART #4
                                ;ID     MTA4-MAG TAPE ADAPTER CONTROLLER PART #4
                                .SBTTL  TEST 1 - LOOP WRITE/READ - MTA - GCR - NORMAL W/R CLOCKS
4300                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*GCR - LOOP WRITE/READ - MTA - NORMAL W/R CLOCKS
4300                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*GCR FORMAT WITH NORMAL WRITE AND READ CLOCKS
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
4300                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT (MTA)
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO NORMAL
                                ;*  :  SET THE READ PATH CLOCK TO NORMAL
                                ;*  :  CALL SUBROUTINE GCRSUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE=640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL GCRSUB SUBROUTINE
                                ;*ENDTST
4300                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA4 MICRO TEST 0'
                                ;*MTA4 MICRO ERR  01
                                ;*MTA4-LOOP WRI'E/READ - MTA - GCR - NORMAL W/R CLOCKS
                                ;*ALL MODULES
                                ;*OPERATOR ERROR NO TM78 UNITS SPECIFIED
                                ;*FATAL ERROR - TEST ABORTED
                                ;*
                                ;*MTA4 MICRO TEST 01
                                ;*MTA4 MICRO ERROR 02
                                ;*MTA4-LOOP WRITE/REAI - MTA - GCR - NORMAL W/R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
```

```
1374                                    ;*EXPECTED DATA IS THE INTERRUPT CODE
1375  4300                              S
 (1)                                    ; ************************************************************************
1376
1377  4300  C3  07  43        10.0            JMP     TEST1
1378  4303  00                                .BYTE   0               ;DUMMY BYTE
1379  4304  00                                .BYTE   0               ;DUMMY BYTE
1380  4305  00                                .BYTE   0               ;DUMMY BYTE
1381  4306  00                    SPEED:      .BYTE   0               ;CLOCK CONTROL WORD
1382  4307                        TEST1:      TESTX   @1              ;INITIALIZE THE TEST
 (1)  4307  3E  01                 7.0                       MVI     A,@1            ;DEFINE THE TEST NUMBER
 (1)  4309  CD  03  28            18.0                      CALL    TSET            ;SETUP THE TEST
1383                                    ;%MTA4-LOOP WRITE/READ - MTA - GCR - NORMAL W/R CLOCKS
1384                                    ;&ALL MODULES
1385  430C                              REQ     @7,0,0,0,0
 (1)  430C  CD  06  28            18.0                      CALL    REQST
 (1)  430F  00                                              .BYTE   0               ;DATA PATTERN NUMBER
 (1)  4310  00  00                                          .WORD   0               ;SYSTEM "0" COUNT
 (1)  4312  00  00                                          .WORD   0               ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (1)  4314  00                                              .BYTE   0               ;DATA COMPARE FLAG IF =1
 (1)  4315  07                                              .BYTE   @7              ;REQUEST CODE
1386  4316                              RIN     R12L
 (1)  4316  DB  94                10.0                      IN      R12L            ;READ R12L INTO AC
 (1)  4318  7F                     4.0                      MOV     A,A             ;RETRY LINK
1387  4319  32  3F  46            13.0              STA     UNITMP
1388  431C  A7                      4.0              ANA     A
1389  431D  C2  2B  43            10.0              JNZ     TST01X
1390  4320                              ERR     EXIT,DUMMY
 (1)                                    ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  4320  CD  09  28            18.0                      CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0001                                            MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4323    01                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4324    00                                            .BYTE
 (1)  4325  CD  15  28            18.0              DUMMY:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  4328  DA  62  45            10.0                      JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1391                                    ;>OPERATOR ERROR NO TM78 UNITS SPECIFIED
1392                                    ;>FATAL ERROR - TEST ABORTED
```

```
1394  432B  21  80  00      10.0  TST01X: LXI   H,128                ;SET UP THE BYTE COUNT
1395  432E  22  40  46      16.0          SHLD  SIZE                 ;
1396                                                                 ;
1397  4331  3E  10           7.0  TST01Y: MVI   A,@020               ;SET UP FOR NORMAL CLOCKS
1398  4333  32  06  43      13.0          STA   SPEED
1399  4336  CD  6C  45      18.0          CALL  GCRSUB          ;CALL SUBROUTINE LWRSUB
1400                                                                 ;
1401  4339  2A  40  46      16.0          LHLD  SIZE                 ;GET THE SIZE
1402  433C  01  80  00      10.0          LXI   B,128                ;LOAD THE INCREMENT VALUE
1403  433F  09              10.0          DAD   B                    ;INCREMENT THE BYTE COUNT
1404  4340  22  40  46      16.0          SHLD  SIZE                 ;SAVE THE SIZE FOR LATER
1405  4343  7C               4.0          MOV   A,H                  ;DO UNTIL THE BYTE COUNT = 768
1406  4344  FE  03           7.0          CPI   $3                   ;
1407  4346  C2  31  43      10.0          JNZ   TST01Y               ;
1408  4349  21  58  04      10.0          LXI   H,1112.              ;SET UP THE BYTE COUNT
1409  434C  22  40  46      16.0          SHLD  SIZE                 ;
1410  434F  CD  6C  45      18.0          CALL  GCRSUB               ;CALL SUBROUTINE GCRSUB
1411  4352                                ENDTST TST01X
  (1)                       ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  4352                               REQ   7                ;FAKE CALL TO KEEP TEST ALIVE
  (2)  4352  CD  06  28      18.0                          CALL    REQST
  (2)  4355  00                                            .BYTE                ;DATA PATTERN NUMBER
  (2)  4356  00  00                                        .WORD                ;SYSTEM "" COUNT
  (2)  4358  00  00                                        .WORD                ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  435A  00                                            .BYTE                ;DATA COMPARE FLAG IF =1
  (2)  435B  07                                            .BYTE   7            ;REQUEST CODE
  (1)  435C  3A  9A  4F      13.0          LDA   ITERA                ;GET ITERATION COUNT
  (1)  435F  3D               4.0          DCR   A                    ;DOWNCOUNT
  (1)  4360  32  9A  4F      13.0          STA   ITERA                ;SAVE COUNT
  (1)  4363  F2  2B  43      10.0          JP    TST01X               ;DO TEST UNTIL TILL   0
1412                                  .
```

```
                                .SBTTL  TEST 2 - LOOP WRITE/READ - MTA - GCR - +10% W/R CLOCKS

4366                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*GCR - LOOP WRITE/READ - MTA +10% W/R CLOCKS
4366                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*GCR FORMAT WITH +10% WRITE AND READ CLOCKS
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
4366                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT (MTA)
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO +10%
                                ;*  :  SET THE READ PATH CLOCK TO +10%
                                ;*  :  CALL SUBROUTINE GCRSUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE=640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL SUBROUTINE GCRSUB
                                ;*ENDTST
4366                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA4 MICRO TEST 02
                                ;*MTA4 MICRO ERROR 02
                                ;*MTA4-LOOP WRITE/READ - MTA - GCR - +10% W/R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
4366                            S
 (1)                            ; ************************************************************************

4366                            TEST02: TESTX   @2              ;INITIALIZE THE TEST
 (1) 4366  3E  02          7.0                   MVI     A,@2            ;DEFINE THE TEST NUMBER
 (1) 4368  CD  03  28     18.0                   CALL    TSET            ;SETUP THE TEST
                                ;%MTA4-LOOP WRITE/READ - MTA - GCR - +10% W/R CLOCKS
```

```
1453                                    ;&ALL MODULES
1454  436B  21  80  00      10.0        LXI     H,128           ;SETUP THE BYTE COUNT
1455  436E  22  40  46      16.0        SHLD    SIZE            ;
1456                                    ;
1457  4371  3E  0A           7.0  TST02X: MVI     A,@012          ;SET W/R CLOCKS TO +10%
1458  4373  32  06  43      13.0        STA     SPEED           ;
1459  4376  CD  6C  45      18.0        CALL    GCRSUB          ;CALL SUBROUTINE LWRSUB
1460  4379  2A  40  46      16.0        LHLD    SIZE            ;GET THE BYTE COUNT
1461  437C  01  80  00      10.0        LXI     B,128           ;SET UP INCREMENT VALUE
1462  437F  09              10.0        DAD     B               ;INCREMENT THE BYTE COUNT
1463  4380  22  40  46      16.0        SHLD    SIZE            ;SAVE THE NEW BYTE COUNT
1464  4383  7C               4.0        MOV     A,H             ;DO UNTIL THE BYTE COUNT=1
1465  4384  FE  03           7.0        CPI     $3
1466  4386  C2  71  43      10.0        JNZ     TST02X
1467  4389  21  58  04      10.0        LXI     H,1112.                 ;SET UP THE BYTE COUNT
1468  438C  22  40  46      16.0        SHLD    SIZE                    ;
1469  438F  CD  6C  45      18.0        CALL    GCRSUB                  ;CALL SUBROUTINE GCRSUB
1470  4392                            ENDTST  TEST02
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4392                          REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4392  CD  06  28      18.0                            CALL    REQST
 (2)  4395  00                                              .BYTE                   ;DATA PATTERN NUMBER
 (2)  4396  00  00                                          .WORD                   ;SYSTEM "" COUNT
 (2)  4398  00  00                                          .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  439A  00                                              .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  439B  07                                              .BYTE   7               ;REQUEST CODE
 (1)  439C  3A  9A  4F      13.0        LDA     ITERA           ;GET ITERATION COUNT
 (1)  439F  3D               4.0        DCR     A               ;DOWNCOUNT
 (1)  43A0  32  9A  4F      13.0        STA     ITERA           ;SAVE COUNT
 (1)  43A3  F2  66  43      10.0        JP      TEST02          ;DO TEST UNTIL TILL = 0
1471
```

```
                                .SBTTL  TEST 3 - LOOP WRITE/READ - MTA - GCR - -20% W -30% R CLOCKS

43A6                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*----- -----
                                ;*GCR - LOOP WRITE/READ - MTA - -20%W -30%R CLOCKS
43A6                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*GCR FORMAT WITH -20% WRITE CLOCK AND -30% READ CLOCK
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
43A6                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT (MTA)
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO -20%
                                ;*  :  SET THE READ PATH CLOCK TO -30%
                                ;*  :  CALL SUBROUTINE GCRSUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE-640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL GCRSUB SUBROUTINE
                                ;*ENDTST
43A6                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA4 MICRO TEST 03
                                ;*MTA4 MICRO ERROR 02
                                ;*MTA4-LOOP WRITE/READ - MTA - GCR - -20% W -30% R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
43A6                            S
 (1)                            ; ************************************************************************

43A6                            TEST03: TESTX   @3              ;INITIALIZE THE TEST
 (1) 43A6  3E  03          7.0                       MVI     A,@3            ;DEFINE THE TEST NUMBER
 (1) 43A8  CD  03  28     18.0                       CALL    TSET            ;SETUP THE TEST
                                ;%MTA4-LOOP WRITE/READ - MTA - GCR - -20% W -30% R CLOCKS
```

```
1512                                    ;&ALL MODULES
1513  43AB  21  80  00      10.0               LXI     H,128           ;SET UP THE BYTE COUNT
1514  43AE  22  40  46      16.0               SHLD    SIZE            ;
1515                                           ;
1516  43B1  3E  1B           7.0   TST03X:     MVI     A,@033          ;SET -20% W AND -30% R CLOCKS
1517  43B3  32  06  43      13.0               STA     SPEED           ;
1518  43B6  CD  6C  45      18.0               CALL    GCRSUB          ;CALL SUBROUTINE LWRSUB
1519  43B9  2A  40  46      16.0               LHLD    SIZE            ;GET THE BYTE COUNT
1520  43BC  01  80  00      10.0               LXI     B,128           ;SET UP THE INCREMENT VALUE
1521  43BF  09              10.0               DAD     B               ;INCREMENT THE BYTE COUNT
1522  43C0  22  40  46      16.0               SHLD    SIZE            ;SAVE THE NEW BYTE COUNT
1523  43C3  7C               4.0               MOV     A,H             ;DO UNTIL THE BYTE COUNT=
1524  43C4  FE  03           7.0               CPI     $3
1525  43C6  C2  B1  43      10.0               JNZ     TST03X
1526  43C9  21  58  04      10.0               LXI     H,1112.                 ;SET UP THE BYTE COUNT
1527  43CC  22  40  46      16.0               SHLD    SIZE                    ;
1528  43CF  CD  6C  43      18.0               CALL    GCRSUB                  ;CALL SUBROUTINE GCRSUB
1529  43D2                                     ENDTST  TEST03
 (1)                                     ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  43D2                                     REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  43D2  CD  06  28      18.0                                 CALL    REQST
 (2)  43D5  00                                                   .BYTE           ;DATA PATTERN NUMBER
 (2)  43D6  00  00                                               .WORD           ;SYSTEM '"' COUNT
 (2)  43D8  00  00                                               .WORD           ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  43DA  00                                                   .BYTE           ;DATA COMPARE FLAG IF =1
 (2)  43DB  07                                                   .BYTE   7       ;REQUEST CODE
 (1)  43DC  3A  9A  4F      13.0               LDA     ITERA           ;GET ITERATION COUNT
 (1)  43DF  3D               4.0               DCR     A               ;DOWNCOUNT
 (1)  43E0  32  9A  4F      13.0               STA     ITERA           ;SAVE COUNT
 (1)  43E3  F2  A6  43      10.0               JP      TEST03          ;DO TEST UNTIL TILL - 0
```

```
                                .SBTTL  TEST 4 - LOOP WRITE/READ - MTA - PE - NORMAL W/R CLOCKS

43E6                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*PE - LOOP WRITE/READ - MTA - NORMAL W/R CLOCKS
43E6                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*PE FORMAT WITH NORMAL WRITE AND READ CLOCKS
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
43E6                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT (MTA)
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO NORMAL
                                ;*  :  SET THE READ PATH CLOCK TO NORMAL
                                ;*  :  CALL SUBROUTINE PESUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE=640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL PESUB SUBROUTINE
                                ;*ENDTST
43E6                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*MTA4 MICRO TEST 04
                                ;*MTA4 MICRO ERROR 02
                                ;*MTA4-LOOP WRITE/READ - MTA - PE - NORMAL W/R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
43E6                            S
 (1)                            ; ************************************************************************

43E6                            TEST04: TESTX   @4              ;INITIALIZE THE TEST
 (1)  43E6  3E   04       7.0                   MVI     A,@4             ;DEFINE THE TEST NUMBER
 (1)  43E8  CD   03   28 18.0                   CALL    TSET             ;SETUP THE TEST
                                ;%MTA4-LOOP WRITE/READ - MTA - PE - NORMAL W/R CLOCKS
```

```
1570                                    ;&ALL MODULES
1571  43EB  21  80  00      10.0         LXI     H,128                   ;SET UP THE BYTE COUNT
1572  43EE  22  40  46      16.0         SHLD    SIZE                    ;
1573                                                                    ;
1574  43F1  3E  10            7.0  TST04X: MVI     A,@020                  ;SET UP FOR NORMAL CLOCKS
1575  43F3  32  06  43      13.0         STA     SPEED
1576  43F6  CD  65  45      18.0         CALL    PESUB           ;CALL SUBROUTINE PESUB
1577                                                                    ;
1578  43F9  2A  40  46      16.0         LHLD    SIZE                    ;GET THE SIZE
1579  43FC  01  80  00      10.0         LXI     B,128                   ;LOAD THE INCREMENT VALUE
1580  43FF  09              10.0         DAD     B                       ;INCREMENT THE BYTE COUNT
1581  4400  22  40  46      16.0         SHLD    SIZE                    ;SAVE THE SIZE FOR LATER
1582  4403  7C               4.0         MOV     A,H                     ;DO UNTIL THE BYTE COUNT = 768
1583  4404  FE  03            7.0         CPI     $3                      ;
1584  4406  C2  F1  43      10.0         JNZ     TST04X                  ;
1585  4409  21  58  04      10.0         LXI     H,1112.                 ;SET UP THE BYTE COUNT
1586  440C  22  40  46      16.0         SHLD    SIZE                    ;
1587  440F  CD  65  45      18.0         CALL    PESUB                   ;CALL SUBROUTINE PESUB
1588  4412                               ENDTST  TEST04
 (1)                                ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4412                          REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4412  CD  06  28      18.0                    CALL    REQST
 (2)  4415  00                                      .BYTE                   ;DATA PATTERN NUMBER
 (2)  4416  00  00                                  .WORD                   ;SYSTEM "" COUNT
 (2)  4418  00  00                                  .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  441A  00                                      .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  441B  07                                      .BYTE   7               ;REQUEST CODE
 (1)  441C  3A  9A  4F      13.0         LDA     ITERA           ;GET ITERATION COUNT
 (1)  441F  3D               4.0         DCR     A               ;DOWNCOUNT
 (1)  4420  32  9A  4F      13.0         STA     ITERA           ;SAVE COUNT
 (1)  4423  F2  E6  43      10.0         JP      TEST04          ;DO TEST UNTIL TILL = 0
1589
```

```
                                .SBTTL  TEST 5 - LOOP WRITE/READ - MTA - PE - +10% W/R CLOCKS

                                ST
                                ; ***************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*PE - LOOP WRITE/READ - MTA +10% W/R CLOCKS
                                SD
                                ; ***************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*PE FORMAT WITH +10% WRITE AND READ CLOCKS
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
                                SP
                                ; ***************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT (MTA)
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO +10%
                                ;*  :  SET THE READ PATH CLOCK TO +10%
                                ;*  :  CALL SUBROUTINE PESUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE=640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL PESUB SUBROUTINE
                                ;*ENDTST
                                SE
                                ; ***************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA4 MICRO TEST 05
                                ;*MTA4 MICRO ERROR 02
                                ;*MTA4-LOOP WRITE/READ - MTA - PE - +10% W/R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
                                S
                                ; ***************************************************************************

                                TEST05: TESTX   @5              ;INITIALIZE THE TEST
 (1)  4426  3E  05          7.0                 MVI     A,@5            ;DEFINE THE TEST NUMBER
 (1)  4428  CD  03  28     18.0                 CALL    TSET            ;SETUP THE TEST
                                ;%MTA4-LOOP WRITE/READ - MTA - PE - +10% W/R CLOCKS
```

```
1630                                    ;&ALL MODULES
1631  442B  21  80  00      10.0               LXI     H,128           ;SETUP THE BYTE COUNT
1632  442E  22  40  46      16.0               SHLD    SIZE            ;
1633                                           ;
1634  4431  3E  0A           7.0       TST05X: MVI     A,@012          ;SET W/R CLOCKS TO +10%
1635  4433  32  06  43      13.0               STA     SPEED           ;
1636  4436  CD  65  45      18.0               CALL    PESUB           ;CALL SUBROUTINE PESUB
1637  4439  2A  40  46      16.0               LHLD    SIZE            ;GET THE BYTE COUNT
1638  443C  01  80  00      10.0               LXI     B,128           ;SET UP INCREMENT VALUE
1639  443F  09              10.0               DAD     B               ;INCREMENT THE BYTE COUNT
1640  4440  22  40  46      16.0               SHLD    SIZE            ;SAVE THE NEW BYTE COUNT
1641  4443  7C               4.0               MOV     A,H             ;DO UNTIL THE BYTE COUNT=1
1642  4444  FE  03           7.0               CPI     $3
1643  4446  C2  31  44      10.0               JNZ     TST05X
1644  4449  21  58  04      10.0               LXI     H,1112.                 ;SET UP BYTE COUNT
1645  444C  22  40  46      16.0               SHLD    SIZE                    ;
1646  444F  CD  65  45      18.0               CALL    PESUB                   ;CALL SUBROUTINE PESUB
1647  4452                                     ENDTST  TEST05
 (1)                                     ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  4452                                     REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
 (2)  4452  CD  06  28      18.0                                CALL    REQST
 (2)  4455  00                                               .BYTE                   ;DATA PATTERN NUMBER
 (2)  4456  00  00                                           .WORD                   ;SYSTEM "" COUNT
 (2)  4458  00  00                                           .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  445A  00                                               .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  445B  07                                               .BYTE   7               ;REQUEST CODE
 (1)  445C  3A  9A  4F      13.0                LDA     ITERA           ;GET ITERATION COUNT
 (1)  445F  3D               4.0                DCR     A               ;DOWNCOUNT
 (1)  4460  32  9A  4F      13.0                STA     ITERA           ;SAVE COUNT
 (1)  4463  F2  26  44      10.0                JP      TEST05          ;DO TEST UNTIL TILL = 0
1648
```

```
                                .SBTTL  TEST 6 - LOOP WRITE/READ - MTA - PE - -20% W -30% R CLOCKS

                                ST
                                ; ************************************************************************
                                ;*TEST TITLE
                                ;*---- -----
                                ;*PE - LOOP WRITE/READ - MTA - -20%W -30%R CLOCKS
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS TEST PERFORMS A LOOP WRITE/READ FUNCTION IN
                                ;*PE FORMAT WITH -20% WRITE CLOCK AND -30% READ CLOCK
                                ;*AND RECORD SIZES OF 128, 256, 384, AND 512 BYTES.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNTST
                                ;*  GET USER SELECTED TU PORT (MTA)
                                ;*  SET THE RECORD SIZE TO 128
                                ;*  BGNDO
                                ;*  :  SET THE WRITE PATH CLOCK TO -20%
                                ;*  :  SET THE READ PATH CLOCK TO -30%
                                ;*  :  CALL SUBROUTINE PESUB
                                ;*  :  ADD 128 TO THE RECORD SIZE
                                ;*  :  DO UNTIL THE RECORD SIZE=640
                                ;*  ENDDO
                                ;*  SET RECORD SIZE TO 1112.
                                ;*  CALL PESUB SUBROUTINE
                                ;*ENDTST
                                SE
                                ; ************************************************************************
                                ;*ERRORS
                                ;*------
                                ;*MTA4 MICRO TEST 06
                                ;*MTA4 MICRO ERROR 02
                                ;*MTA4-LOOP WRITE/READ - MTA - PE - -20% W -30% R CLOCKS
                                ;*ALL MODULES
                                ;*LOOP WRITE/READ ERROR
                                ;*ACTUAL = AAAA
                                ;*EXPECTED = EEEE
                                ;*BYTE-SCLK COUNT = LLL
                                ;*DATA FORMAT = MM
                                ;*SKIP COUNT = NN
                                ;*ACTUAL DATA IS THE FAILURE CODE
                                ;*EXPECTED DATA IS THE INTERRUPT CODE
                                S
                                ; ************************************************************************

4466                            TEST06: TESTX   @6              ;INITIALIZE THE TEST
4466  3E  06          7.0                       MVI     A,@6            ;DEFINE THE TEST NUMBER
4468  CD  03  28      18.0                      CALL    TSET            ;SETUP THE TEST
                                ;%MTA4-LOOP WRITE/READ - MTA - PE - -20% W -30% R CLOCKS
```

```
1689                                    ;&ALL MODULES
1690  446B  21  80  00      10.0               LXI     H,128           ;SET UP THE BYTE COUNT
1691  446E  22  40  46      16.0               SHLD    SIZE            ;
1692                                            ;
1693  4471  3E  1B            7.0  TST06X:     MVI     A,@033          ;SET -20% W AND -30% R CLOCKS
1694  4473  32  06  43      13.0               STA     SPEED           ;
1695  4476  CD  65  45      18.0               CALL    PESUB           ;CALL SUBROUTINE PESUB
1696  4479  2A  40  46      16.0               LHLD    SIZE            ;GET THE BYTE COUNT
1697  447C  01  80  00      10.0               LXI     B,128           ;SET UP THE INCREMENT VALUE
1698  447F  09              10.0               DAD     B               ;INCREMENT THE BYTE COUNT
1699  4480  22  40  46      16.0               SHLD    SIZE            ;SAVE THE NEW BYTE COUNT
1700  4483  7C               4.0               MOV     A,H             ;DO UNTIL THE BYTE COUNT=
1701  4484  FE  03            7.0               CPI     $3
1702  4486  C2  71  44      10.0               JNZ     TST06X
1703  4489  21  58  04      10.0               LXI     H,1112.                 ;SET UP BYTE COUNT
1704  448C  22  40  46      16.0               SHLD    SIZE                    ;
1705  448F  CD  65  45      18.0               CALL    PESUB                   ;CALL SUBROUTINE PESUB
1706  4492                                    ENDTST  TEST06
  (1)                                   ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
  (2)  4492                                 REQ     7                       ;FAKE CALL TO KEEP TEST ALIVE
  (2)  4492  CD  06  28      18.0                                   CALL    REQST
  (2)  4495  00                                                     .BYTE                   ;DATA PATTERN NUMBER
  (2)  4496  00  00                                                 .WORD                   ;SYSTEM "" COUNT
  (2)  4498  00  00                                                 .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
  (2)  449A  00                                                     .BYTE                   ;DATA COMPARE FLAG IF =1
  (2)  449B  07                                                     .BYTE   7               ;REQUEST CODE
  (1)  449C  3A  9A  4F      13.0               LDA     ITERA           ;GET ITERATION COUNT
  (1)  449F  3D               4.0               DCR     A               ;DOWNCOUNT
  (1)  44A0  32  9A  4F      13.0               STA     ITERA           ;SAVE COUNT
  (1)  44A3  F2  66  44      10.0               JP      TEST06          ;DO TEST UNTIL TILL = 0
```

```
                                .SBTTL  TEST7 - MTA VELOCITY TESTS

44A6                            ST
 (1)                            ; ************************************************************************
 (1)                            ;*TEST TITLE
 (1)                            ;*---- -----
                                ;*MTA - VELOCITY TESTS
44A6                            SD
 (1)                            ; ************************************************************************
 (1)                            ;*DESCRIPTION
 (1)                            ;*-----------
                                ;*THIS TEST PERFORMS THE TAPE SPEED, START RAMP, STOP RAMP FORWARD,
                                ;*AND STOP RAMP REVERSE TESTS ON THE SELECTED TU78 WHICH MUST BE IN
                                ;*THE MAINTENANCE MODE.
44A6                            SP
 (1)                            ; ************************************************************************
 (1)                            ;*PROCEDURE
 (1)                            ;*---------
                                ;*BGNTST
                                ;*    GET THE USER SELECTED TU PORT (MTA)
                                ;*    CALL SUBROUTINE CLEAR
                                ;*    IF MTA IS IN THE MAINTENANCE POSITION
                                ;*    :    THEN - CONTINUE
                                ;*    :    ELSE - ERROR
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE VELTST
                                ;*    IF CARRY SET
                                ;*    :    THEN - ERROR
                                ;*    :    ELSE - CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE RUPTST
                                ;*    IF CARRY SET
                                ;*    :    THEN - ERROR
                                ;*    :    ELSE - CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE FWDTST
                                ;*    IF CARRY SET
                                ;*    :    THEN - ERROR
                                ;*    :    ELSE - CONTINUE
                                ;*    ENDIF
                                ;*    CALL SUBROUTINE REVTST
                                ;*    IF CARRY SET
                                ;*    :    THEN ERROR
                                ;*    :    ELSE - CONTINUE
                                ;*    ENDIF
                                ;*ENDTST
44A6                            SE
 (1)                            ; ************************************************************************
 (1)                            ;*ERRORS
 (1)                            ;*------
                                ;*
44A6                            S
 (1)                            ; ************************************************************************
```

```
1749  44A6                          TEST07: TESTX   @7              ;INITIALIZE THE TEST
 (1)  44A6  3E  07          7.0                        MVI     A,@7            ;DEFINE THE TEST NUMBER
 (1)  44A8  CD  03  28     18.0                        CALL    TSET            ;SETUP THE TEST
1750                                ;%MTA - VELOCITY TESTS
1751                                ;&VELOCITY ADJUSTMENTS
1752  44AB                                  RIN     R11L            ;GET THE MANUAL INTERVENTION BYTE
 (1)  44AB  DB  92          10.0                       IN      R11L            ;READ R11L INTO AC
 (1)  44AD  7F               4.0                       MOV     A,A             ;RETRY LINK
1753  44AE  E6  20           7.0            ANI     BIT5            ;CHECK IF MANUAL INTERVENTION ALLOWED
1754  44B0  CA  62  45       10.0            JZ      EXIT            ;IF NOT - QUIT
1755  44B3  06  00           7.0            MVI     B,@0            ;ELSE DO VELOCITY TESTS
1756
1757  44B5  3A  3F  46       13.0            LDA     UNITMP          ;DID THE USER SPECIFY TU
1758  44B8  E6  01           7.0            ANI     @01             ;PORT 0?
1759  44BA  C2  D0  44       10.0            JNZ     FOUND1          ;YES - GO USE PORT #0
1760  44BD  04               4.0            INR     B               ;NO - UPDATE POINTER TO PORT #1
1761  44BE  3A  3F  46       13.0            LDA     UNITMP          ;DID THE USER SPECIFY TU
1762  44C1  E6  02           7.0            ANI     @02             ;PORT 1 ?
1763  44C3  C2  D0  44       10.0            JNZ     FOUND1          ;YES - GO USE PORT #1
1764  44C6  04               4.0            INR     B               ;NO - UPDATE THE POINTER TO PORT #2
1765  44C7  3A  3F  46       13.0            LDA     UNITMP          ;DID THE USER SPECIFY TU
1766  44CA  E6  04           7.0            ANI     @04             ;PORT 2 ?
1767  44CC  C2  D0  44       10.0            JNZ     FOUND1          ;YES - GO USE PORT #2
1768  44CF  04               4.0            INR     B               ;NO - ASSUME PORT #3
1769  44D0  CD  13  46       18.0  FOUND1:   CALL    CLEAR
1770  44D3  DB  E0           10.0            IN      INTSTA
1771  44D5  E6  80           7.0            ANI     BIT7
1772  44D7  B0               4.0            ORA     B
1773  44D8  D3  E0           10.0            OUT     MBSEL           ;SELECT DESIRED TU PORT AND MASSBUS
1774
1775  44DA  3E  82           7.0          MVI     A,$82
1776  44DC  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER 2
1777  44DE  DB  40          10.0          IN      TSTS            ;GET THE TU PORT STATUS
1778  44E0  E6  07           7.0          ANI     $07             ;SAVE THE PORT SELECT BITS
1779  44E2  FE  07           7.0          CPI     BIT0!BIT1.BIT2  ;COMPARE TO MAINTENANCE POSITION
1780  44E4  CA  EC  44       10.0          JZ      T2L01           ;YES - CONTINUE TESTING
1781  44E7                                ERR     EXIT,T2L01
 (1)                                ;FLAG AN ERROR - NO EXPECTED OR ACTUAL
 (1)  44E7  CD  09  28     18.0                        CALL    ERLP            ;PROCESS ERROR - DO 2.3
 (1)        0002                                     MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  44EA  02                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  44EB  00                                       .BYTE
 (1)  44EC  CD  15  28     18.0                T2L01:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.3
 (1)  44EF  DA  62  45     10.0                        JC      EXIT            ;LOOP ADDRESS IF LOOP SPECIFIED
1782                          ;>TU78 NOT IN MAINTENANCE POSITION - VELOCITY TESTS ABORTED
1783                          ;>FATAL ERROR - TEST ABORTED
1784
```

```
1786  44F2  CD  5B  00      18.0  VELCOM: CALL    VELTST
1787  44F5  F5                12.0          PUSH    PSW             ;SAVE THE FLAGS
1788                              ;         ANA     A               ;MANIPULATE THE FLAGS
1789                              ;         JP      1$              ;SKIP AROUND IF POSITIVE
1790                              ;         CMA
1791                              ;         INR     A               ;COMPLEMENT THE VALUE
1792  44F6                        1$:       ROUT    ADATA           ;SAVE IT FOR ERROR REPORT
 (1)  44F6  D3  94          10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  44F8  7F               4.0                    MOV     A,A             ;RETRY LINK
1793  44F9  F1              10.0          POP     PSW             ;GET THE FLAGS
1794  44FA  D2  02  45       10.0          JNC     T2L02           ;
1795  44FD                          ERRA    VELCOM,T2L02
 (1)                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  44FD  CD  0F  28       18.0                    CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)      0003                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4500  03                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4501  00                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4502  CD  15  28       18.0                T2L02:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4505  DA  F2  44       10.0                    JC      VELCOM          ;LOOP ADDRESS IF LOOP SPECIFIED
1796                          ;>TU78 - TAPE SPEED FAILURE
1797
1798  4508  A7                 4.0          ANA     A
1799  4509  CD  5E  00       18.0          CALL    RUPTST
1800  450C  F5                12.0          PUSH    PSW             ;SAVE THE FLAGS
1801                              ;         ANA     A               ;MANIPULATE THE FLAGS
1802                              ;         JP      1$              ;SKIP AROUND IF POSITIVE
1803                              ;         CMA
1804                              ;         INR     A               ;COMPLEMENT THE VALUE
1805  450D                        1$:       ROUT    ADATA           ;SAVE IT FOR ERRORR REPORT
 (1)  450D  D3  94          10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  450F  7F               4.0                    MOV     A,A             ;RETRY LINK
1806  4510  F1              10.0          POP     PSW             ;GET THE FLAGS
1807  4511  D2  19  45       10.0          JNC     T2L03
1808  4514                          ERRA    T2L02,T2L03
 (1)                              ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4514  CD  0F  28       18.0                    CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)      0004                                       MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4517  04                                       .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4518  00                                       .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4519  CD  15  28       18.0                T2L03:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  451C  DA  02  45       10.0                    JC      T2L02           ;LOOP ADDRESS IF LOOP SPECIFIED
1809                          ;>TU78 - START RAMP FAILURE
1810
1811  451F  A7                 4.0          ANA     A
1812  4520  CD  61  00       18.0          CALL    RWDTST
1813  4523  F5                12.0          PUSH    PSW             ;SAVE THE FLAGS
1814                              ;         ANA     A               ;MANIPULATE THE FLAGS
1815                              ;         JP      1$              ;SKIP AROUND IF POSITIVE
1816                              ;         CMA
1817                              ;         INR     A               ;COMPLEMENT THE VALUE
1818  4524                        1$:       ROUT    ADATA           ;SAVE IT FOR ERROR REPORT
 (1)  4524  D3  94          10.0                    OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  4526  7F               4.0                    MOV     A,A             ;RETRY LINK
1819  4527  F1              10.0          POP     PSW             ;GET THE FLAGS
```

```
1820  4528  D2  30  45  10.0          JNC     T2L04
1821  452B                            ERRA    T2L03,T2L04
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  452B  CD  0F  28  18.0                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0005                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  452E  05                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  452F  00                                .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4530  CD  15  28  18.0          T2L04:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  4533  DA  19  45  10.0                  JC      T2L03           ;LOOP ADDRESS IF LOOP SPECIFIED
1822                            ;>TU78 - FORWARD STOP RAMP FAILURE
1823
1824  4536  A7                    4.0       ANA     A
1825  4537  CD  64  00  18.0              CALL    REVTST
1826  453A  F5                    12.0      PUSH    PSW             ;SAVE THE FLAGS
1827                                ;       ANA     A               ;MANIPULATE THE FLAGS
1828                                ;       JP      1$              ;SKIP AROUND IF POSITIVE
1829                                ;       CMA
1830  453B  3C                     4.0      INR     A               ;COMPLEMENT THE VALUE
1831  453C                          1$:     ROUT    ADATA           ;SAVE IT FOR ERROR REPORT
 (1)  453C  D3  94                  10.0              OUT     ADATA           ;WRITE AC INTO ADATA
 (1)  453E  7F                       4.0              MOV     A,A             ;RETRY LINK
1832  453F  F1                    10.0      POP     PSW             ;GET THE FLAGS
1833  4540  D2  48  45  10 0       JNC     T2L05
1834  4543                            ERRA    T2L04,T2L05
 (1)                                  ;FLAG ERROR - WITH ACTUAL DATA "ADATA" VALID
 (1)  4543  CD  0F  28  18.0                  CALL    ERLPA           ;PROCESS ERROR - DO 2.3
 (1)        0006                              MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  4546  06                                .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  4547  00                                .BYTE                   ;PRINT ROUTINE NUMBER
 (1)  4548  CD  15  28  18.0          T2L05:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  454B  DA  30  45  10.0                  JC      T2L04           ;LOOP ADDRESS IF LOOP SPECIFIED
1835                            ;>TU78 - REVERSE STOP RAMP FAILURE
1836
1837  454E                              ENDTST  TEST07
 (1)                            ;TEST ITERATION CONTROL - ONCE FOR QUICK VERIFY
 (2)  454E                              REQ     7               ;FAKE CALL TO KEEP TEST ALIVE
 (2)  454E  CD  06  28  18.0                          CALL    REQST
 (2)  4551  00                                        .BYTE                   ;DATA PATTERN NUMBER
 (2)  4552  00  00                                    .WORD                   ;SYSTEM "" COUNT
 (2)  4554  00  00                                    .WORD                   ;REGISTER 02 RECORD COUNT/FMT TYPE/SKIP
 (2)  4556  00                                        .BYTE                   ;DATA COMPARE FLAG IF =1
 (2)  4557  07                                        .BYTE   7               ;REQUEST CODE
 (1)  4558  3A  9A  4F  13.0          LDA     ITERA           ;GET ITERATION COUNT
 (1)  4558  3D                 .      4.0      DCR     A               ;DOWNCOUNT
 (1)  455C  32  9A  4F  13.0          STA     ITERA           ;SAVE COUNT
 (1)  455F  F2  A6  44  10.0          JP      TEST07          ;DO TEST UNTIL TILL = 0
1838  4562  C3  18  28  10.0  EXIT:   JMP     TSTEND
```

```
1840                                        .SBTTL  SUBROUTINE PESUB
1841
1842  4565  AF                      4.0   PESUB:  XRA     A               ;CLEAR THE ACCUMULATOR
1843  4566  32  3E  46              13.0           STA     PGCTL           ;SET THE PE FLAG
1844  4569  C3  71  45              10.0           JMP     PEGCOM
1845
1846  456C  3E  FF                  7.0   GCRSUB: MVI     A,@377          ;SET THE GCR FLAG
1847  456E  32  3E  46              13.0           STA     PGCTL
1848
1849  4571  06  00                  7.0   PEGCOM: MVI     B,@0
1850  4573  3A  3F  46              13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1851  4576  E6  01                  7.0           ANI     @01             ;PORT 0?
1852  4578  C2  8E  45              10.0           JNZ     FOUND           ;YES-GO USE PORT #0
1853  457B  04                      4.0           INR     B               ;NO-UPDATE POINTER TO PORT #1
1854  457C  3A  3F  46              13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1855  457F  E6  02                  7.0           ANI     @02             ;PORT 1?
1856  4581  C2  8E  45              10.0           JNZ     FOUND           ;YES-GO USE PORT #1
1857  4584  04                      4.0           INR     B               ;NO-UPDATE POINTER TO PORT #2
1858  4585  3A  3F  46              13.0           LDA     UNITMP          ;DID THE USER SPECIFY TU
1859  4588  E6  04                  7.0           ANI     @04             ;PORT 2?
1860  458A  C2  8E  45              10.0           JNZ     FOUND           ;YES-GO USE PORT #2
1861  458D  04                      4.0           INR     B               ;NO-ASSUME PORT #3
1862  458E  CD  13  46              18.0   FOUND:  CALL    CLEAR
1863  4591  DB  E0                  10.0           IN      INTSTA
1864  4593  E6  80                  7.0           ANI     BIT7
1865  4595  B0                      4.0           ORA     B
1866  4596  D3  E0                  10.0           OUT     MBSEL           ;SELECT THE DESIRED TU-PORT AND MASSBUS
1867
1868  4598  AF                      4.0   T1L01:  XRA     A               ;CLEAR THE DATA BUS CONTROL
1869  4599  32  28  4F              13.0           STA     DBUS            ;MASK TO KEEP THE MASSBUS CLEAN
1870                                              ;
1871  459C  2A  40  46              16.0           LHLD    SIZE            ;GET THE RECORD SIZE
1872  459F  22  23  4F              16.0           SHLD    BYTEL           ;TO 128 BYTES
1873  45A2  7D                      4.0           MOV     A,L             ;GET THE BYTE COUNT LOW
1874  45A3                                        ROUT    RO5L            ;SAVE IN BYTE COUNT LOW
 (1)  45A3  D3  8A                  10.0                   OUT     RO5L            ;WRITE AC INTO RO5L
 (1)  45A5  7F                      4.0                    MOV     A,A             ;RETRY LINK
1875  45A6  7C                      4.0           MOV     A,H             ;GET THE BYTE COUNT HIGH
1876  45A7                                        ROUT    RO5H            ;SAVE IN BYTE COUNT HIGH
 (1)  45A7  D3  8B                  10.0                   OUT     RO5H            ;WRITE AC INTO RO5H
 (1)  45A9  7F                      4.0                    MOV     A,A             ;RETRY LINK
1877                                              ;
1878  45AA  3A  42  46              13.0           LDA     FORMT           ;GET THE FORMAT
1879  45AD  32  25  4F              13.0           STA     FORMAT          ;NO SKIP COUNT
1880  45B0                                        ROUT    RO2H            ;SAVE IN THE CAS
 (1)  45B0  D3  85                  10.0                   OUT     RO2H            ;WRITE AC INTO RO2H
 (1)  45B2  7F                      4.0                    MOV     A,A             ;RETRY LINK
1881                                              ;
1882  45B3  3E  88                  7.0           MVI     A,DDRCO         ;CLEAR DDR IN, SO DDR
1883  45B5  D3  DB                  10.0           OUT     DDRCTL          ;WRITES TO TAPE
1884                                              ;
1885  45B7  2A  43  46              16.0           LHLD    KWDAT           ;GET THE WRITE DATA
1886  45BA  29                      10.0           DAD     H               ;SHIFT IT LEFT 1 BIT POSITION
1887  45BB  17                      4.0           RAL                     ;SAVE THE CARRY
```

```
1888  45BC  29              10.0          DAD     H               ;SHIFT IT LEFT ANOTHER BIT POSITION
1889  45BD  17               4.0          RAL                     ;SAVE THAT CARRY TOO.
1890  45BE  E6  03           7.0          ANI     $3              ;SAVE ONLY THE CARRY BITS
1891  45C0  47               4.0          MOV     B,A             ;SAVE IN REGISTER B
1892  45C1  7C               4.0          MOV     A,H             ;
1893  45C2  D3  D9          10.0          OUT     DDRB            ;LOAD <13:6> IN DDR B
1894  45C4  3A  45  46      13.0          LDA     KWDAT+2         ;GET BITS <17:16>
1895  45C7  85               4.0          ADD     L               ;MERGE WITH <5:0>
1896  45C8  D3  D8          10.0          OUT     DDPA            ;LOAD INTO DDR A
1897  45CA  AC               4.0          XRA     H               ;COMPUTE PARITY ON ALL BITS
1898  45CB  A8               4.0          XRA     B               ;
1899  45CC  78               4.0          MOV     A,B             ;GET BITS <15:14> AGAIN
1900  45CD  E2  D2  45      10.0          JPO     1$              ;JUMP IF ODD PARITY
1901  45D0  F6  04           7.0          ORI     BIT2            ;SET THE PARITY BIT TO MAKE IT ODD
1902  45D2  D3  DA          10.0  1$:     OUT     DDRC            ;LOAD <P:14> IN DDRC
1903                                                             ;
1904  45D4  21  03  43      10.0          LXI     H,SPEED-3       ;POINT TO THE CLOCK
1905  45D7  22  26  4F      16.0          SHLD    CLOCK           ;SPEED TABLE
1906  45DA  3E  00           7.0          MVI     A,0
1907  45DC  D3  D2          10.0          OUT     TRKENA
1908  45DE  D3  D2          10.0          OUT     TRKENA
1909  45E0  3E  FF           7.0          MVI     A,@377
1910  45E2  D3  D2          10.0          OUT     TRKENA
1911  45E4  D3  D2          10.0          OUT     TRKENA
1912  45E6  3A  3E  46      13.0          LDA     PGCTL           ;GET THE PE/GCR FLAG
1913  45E9  A7               4.0          ANA     A               ;SET CONDITION BITS
1914  45EA  C2  F3  45      10.0          JNZ     2$              ;GO DO GCR
1915  45ED  CD  55  00      18.0          CALL    LKLWMP          ;ELSE-DO PE
1916  45F0  C3  F6  45      10.0          JMP     3$              ;
1917  45F3  CD  58  00      18.0  2$:     CALL    LKLWMG          ;CALL LOOP WRITE/READ - MTA - GCR
1918                                                             ;
1919  45F6  D2  0C  46      10.0  3$:     JNC     T1CN1           ;CONTINUE IF NO ERROR
1920  45F9  78               4.0          MOV     A,B             ;GET THE FAILURE CODE
1921  45FA  E6  FC           7.0          ANI     $FC             ;MASK OUT THE MEANINGFUL BITS
1922  45FC  0F               4.0          RRC                     ;POSITION FOR OUTPUT
1923  45FD  0F               4.0          RRC                     ;
1924  45FE                                ROUT    ADATA           ;STORE IN THE CAS
 (1)  45FE  D3  94          10.0                  OUT     ADATA   ;WRITE AC INTO ADATA
 (1)  4600  7F               4.0                  MOV     A,A     ;RETRY LINK
1925  4601  79               4.0          MOV     A,C             ;GET THE INTERRUPT CODE
1926  4602  E6  3F           7.0          ANI     $3F             ;MASK OUT THE MEANINGFUL BITS
1927  4604                                ROUT    EDATA           ;STORE IN THE CAS
 (1)  4604  D3  95          10.0                  OUT     EDATA   ;WRITE AC INTO EDATA
 (1)  4606  7F               4.0                  MOV     A,A     ;RETRY LINK
```

```
1929  4607                                          ERRB    TEST07,T1CN1,3
 (1)                                                ;FLAG ERROR - WITH ACTUAL DATA "ADATA" AND EXPECTED DATA "EDATA" VALID
 (1)  4607  CD   12   28        18.0                          CALL    ERLPB           ;PROCESS ERROR - DO 2.3
 (1)        0007                                          MSGN    =       MSGN+1          ;UPDATE MESSAGE NUMBER FOR THIS
 (1)  460A  07                                            .BYTE   MSGN            ;MESSAGE NUMBER ID
 (1)  460B  03                                            .BYTE   3               ;PRINT ROUTINE NUMBER
 (1)  460C  CD   15   28        18.0                T1CN1:: CALL    CKLOP           ;CHECK LOOP FUNCTION - DO 2.2
 (1)  460F  DA   A6   44        10.0                        JC      TEST07          ;LOOP ADDRESS IF LOOP SPECIFIED
1930                                            ;>LOOP WRITE/READ ERROR
1931                                            ;<ACTUAL DATA IS THE FAILURE CODE
1932                                            ;<EXPECTED DATA IS THE INTERRUPT CODE
1933  4612  C9                 10.0                RET
```

```
                                .SBTTL  SUBROUTINE CLEAR ALL MTAS
                                SSUB
                                ; ************************************************************************
                                ;*SUBROUTINE TITLE
                                ;*---------- -----
                                ;*CLEAR ALL MTAS
                                SD
                                ; ************************************************************************
                                ;*DESCRIPTION
                                ;*-----------
                                ;*THIS SUBROUTINE IS USED TO DESELECT ALL TRANSPORTS ATTACHED TO THE
                                ;*SUBSYSTEM AND TO CLEAR ALL MTA PENDING INTERRUPTS, AMTIE LINES
                                ;*AND LOOP MODES.
                                SP
                                ; ************************************************************************
                                ;*PROCEDURE
                                ;*---------
                                ;*BGNSUB
                                ;*    SELECT A PORT TO CLEAR (START AT 0 - STOP AT 3)
                                ;*    CLEAR SELECTED TRANSPORT COMMAND REGISTER A
                                ;*    CLEAR PORT SELECT FOR TRANSPORT
                                ;*    CLEAR PORT PARITY ERRORS & ENABLE WORD
                                ;*    CLEAR PORT DIAGNOSTIC CONTROL
                                ;*    CLEAR PORT AMTIE WORD
                                ;*ENDSUB
                                S
                                ; ************************************************************************

4613  F5              12.0  CLEAR:  PUSH    PSW             ;SAVE THE SELECTED PORT #
4614  C5              12.0          PUSH    B
4615  06  00           7.0          MVI     B,0             ;START TO CLEAR AT PORT #0
4617  DB  E0          10.0  CLRLP:  IN      INTSTA          ;GET MB SELECT INFO
4619  E6  80           7.0          ANI     BIT7            ;SAVE ONLY THE MASSBUS SELECT BIT
461B  B0               4.0          ORA     B               ;ADD IN THE SELECTED PORT #
461C  D3  E0          10.0          OUT     MBSEL           ;RESET TO THIS PORT
461E  3E  80           7.0          MVI     A,@200          ;LOAD MTA REGISTER #0 SELECT CODE
4620  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #0 FOR TRANSPORT #0
4622  AF               4.0          XRA     A               ;CLEAR TU COMMAND A
4623  D3  40          10.0          OUT     TCMD            ;
4625  3E  81           7.0          MVI     A,@201          ;LOAD MTA REGISTER #1 SELECT CODE
4627  D3  40          10.0          OUT     TCMD            ;SELECT MTA REGISTER #1 FOR TRANSPORT #1
4629  3E  00           7.0          MVI     A,SELCLR        ;LOAD TU "CLEAR SELECT" COMMAND
462B  D3  40          10.0          OUT     TCMD            ;ISSUE TU "CLEAR SELECT" FOR TRANSPORT #0
462D  AF               4.0          XRA     A
462E  D3  44          10.0          OUT     TAMT            ;CLEAR AMTIE WORD
4630  D3  48          10.0          OUT     PDIAG           ;CLEAR DIAG CONTROL WORD
4632  D3  4C          10.0          OUT     PENAB           ;CLEAR PORT ENABLE WORD
4634  04               4.0          INR     B               ;POINT TO THE NEXT PORT TO CLEAR
4635  78               4.0          MOV     A,B
4636  FE  04           7.0          CPI     4               ;DONE?
4638  C2  17  46      10.0          JNZ     CLRLP           ;NO - CLEAR THIS PORT ALSO
463B  C1              10.0          POP     B               ;RESET B & C
463C  F1              10.0          POP     PSW             ;ALL DONE
463D  C9              10.0          RET                     ;EXIT
```

```
1980                                    .SBTTL  PROGRAM VARIABLES
1981
1982  463E   00                         PGCTL:  .BYTE   0               ;PE/GCR CONTROL FLAG
1983  463F   00                         UNITMP: .BYTE   0               ;UNIT MAP
1984  4640   00    00                   SIZE:   .WORD   0               ;RECORD SIZE
1985  4642   00                         FORMT:  .BYTE   0               ;DATA FORMAT
1986  4643   00                         KWDAT:  .BYTE   0               ;18 BITS OF INPUT DATA
1987  4644   00                                 .BYTE   0
1988  4645   00                                 .BYTE   0
1989        0000                                .END
```

```
A      =%0007          ADATA  = 0094          AMTIEP= 0001          AMTIE7= 0002
ARAIDF= 0098           ASAVE    4F9B          ATTCD   4F97          AXNUM   4F91
B      =%0000          BADST  = 0090          BIT0   = 0001          BIT1   = 0002
BIT15  = 8000          BIT2   = 0004          BIT3   = 0008          BIT4   = 0010
BIT5   = 0020          BIT6   = 0040          BIT7   = 0080          BIT8   = 0100
BIT9   = 0200          BRKPBC= 4F0A           BRKRAM= 4F10          BRKSTR= 4E60
BRKXCT= 4F00           BSAVE    4F9C          BYTCNT= 00D4          BYTEH   4F24
BYTEL    4F23          C      =%0001          CASCT   4F21          CASCTL= 00A0
CASSTA= 00A0           CATTH  = 0089          CATTL  = 0088          CBUSST= 00A1
CBYTH  = 008B          CBYTL  = 008A          CDG1H  = 0087          CDG1L  = 0086
CDG2H  = 0093          CDG2L  = 0092          CDG3H  = 0095          CDG3L  = 0094
CDVTH  = 008D          CDVTL  = 008C          CHPTIE= 0028          CHOTIE= 0020
CH1TIE= 0021           CH2TIE= 0022           CH3TIE= 0023          CH4TIE= 0024
CH5TIE= 0025           CH6TIE= 0026           CH7TIE= 0027          CKLOP  = 2815
CLEAR    4613          CLKCTL= 00F0           CLOCK   4F26          CLRLP   4617
CMCOH  = 0099          CMCOL  = 0098          CMC1H  = 009B          CMC1L  = 009A
CMC2H  = 009D          CMC2L  = 009C          CMC3H  = 009F          CMC3L  = 009E
CMINH  = 0097          CMINL  = 0096          CNTCTL= 00D7          CRCWRD= 0018
CSAVE    4F9D          CSRLH  = 0091          CSRLL  = 0090          CTCH   = 0085
CTCL   = 0084          CTSTH  = 008F          CTSTL  = 008E          CXCTH  = 0081
CXCTL  = 0080          CXINH  = 0083          CXINL  = 0082          C.     = 0001
C.AVAI= 0080           C.DP   = 0008          C.DSE  = 0010          C.DTU  = 0003
C.DVA  = 0008          C.FAIL= 00FC           C.FMT  = 0070          C.FNCT= 003E
C.GO   = 0001          C.INTC= 00FE           C.MAIN= 0020          C.NSA  = 0080
C.RCT  = 00FC          C.SER  = 0080          C.SHR  = 0040          C.SKPC= 000F
C.TAPE= 0040           C.WCS  = 0002          D      =%0002          DATACT= 00D0
DBUS     4F28          DBUSCT= 00C0           DBUSST= 00C0          DDRA   = 00D8
DDRAIN= 0010           DDRB   = 00D9          DDRBIN= 0002          DDRC   = 00DA
DDRCIN= 0001           DDRCO  = 0088          DDRCTL= 00DB          DIAFLG  4F22
DIAGPG= 4300           DIAGRM= 4F90           DIARD  = 000B          DONE1  = 0045
DONINT= 0010           DSAVE    4F9E          DSE    = 0006          DUMMY   4325  G
D.ATH0= 0001           D.ATH1= 0002           D.EOTD= 0010          D.LAGC= 0020
D.NOIW= 0040           D.NTHR= 0004           D.TACH= 0008          D.WR4  = 0080
E      =%0003          ECCBAD= 0042           ECCCOR= 0019          ECCOK  = 0041
ECCSTA= 001A           ECCTST= 000E           EDATA  = 0095          EOTCLR= 0003
ERFLG    4F93          ERLP   = 2809          ERLPA  = 280F          ERLPB  = 2812
ERLPE  = 280C          ERNUM    4F90          ERRCNT= 00D6          ESAVE    4F9F
EXIT     4562          E.ACRC= 0010           E.AMT  = 0020          E.CDP  = 0080
E.CRC  = 0080          E.PNTR= 0008           E.RPE  = 0040          E.STEC= 0001
E.TTEC= 0002           E.UNC  = 0004          FIFORD= 006A          FORMAT  4F25
FORMT    4642          FOUND    458E          FOUND1  44D0          FWDTST= 0061
GCRID  = 0089          GCRSET= 0002           GCRSUB  456C          GOODTM= 0092
H      =%0004          HLSAVE   4FA0          IE     = 0008          INTSTA= 00E0
ITERA    4F9A          I.PWR  = 0020          I.RMPE= 0040          I5.5   = 0010
I6.5   = 0020          I7.5   = 0040          KCALL  = 005F          KCLR   = 007B
KDEP   = 003F          KENAB  = 0078          KEXAM  = 003E          KEYBRD= 00C8
KEY1   = 0078          KEY10  = 006D          KEY11  = 006E          KEY12  = 006F
KEY13  = 005C          KEY14  = 005D          KEY15  = 005E          KEY16  = 005F
KEY17  = 003C          KEY18  = 003D          KEY19  = 003E          KEY2   = 0079
KEY20  = 003F          KEY3   = 007A          KEY4   = 007B          KEY5   = 0074
KEY6   = 0075          KEY7   = 0076          KEY8   = 0077          KEY9   = 006C
KINTA  = 006F          KLDAD  = 003D          KN0    = 003C          KN1    = 005C
KN2    = 005D          KN3    = 005E          KN4    = 006C          KN5    = 006D
KN6    = 006E          KN7    = 0074          KN8    = 0075          KN9    = 0076
```

```
KU2    = 0079          KU3    = 007A          KU8    = 0077          KWDAT   4643
L      =%0005          LBLANK= 000F           LCE    = 000B          LCH    = 000C
LCL    = 000D          LCP    = 000E          LC0    = 0000          LC1    = 0001
LC2    = 0002          LC3    = 0003          LC4    = 0004          LC5    = 0005
LC6    = 0006          LC7    = 0007          LC8    = 0008          LC9    = 0009
LDLEDA= 00CA           LDLEDB= 00CB           LDLEDC= 00CC           LDLEDD= 00CD
LDLEDE= 00CE           LDLEDF= 00CF           LKDIAG= 2800           LKKBD  = 004C
LKKEY  = 0049          LKLWMG= 0058           LKLWMP= 0055           LKLWPG= 0052
LKLWPP= 004F           LKMOD7= 0046           LKOPR  = 0046          LPFLG   4F94
LPNUM   4F92           M      =%0006          MBSEL  = 00E0          MB.A   = 0008
MB.B   = 0004          MEMTOP= 4FFF           MINUS  = 000A          MM     = 8000
MSE    = 0008          MSGN   = 0007          MTACLR= 0000           MT.ARA= 0020
MT.CPE= 0080           MT.DSE= 0001           MT.FWD= 0040           MT.INH= 0008
MT.LWR= 0004           MT.MOT= 0002           MT.NWT= 0080           MT.PEC= 0040
MT.PSB= 0004           MT.PS0= 0001           MT.PS1= 0002           MT.REV= 0020
MT.WRT= 0010           MT.Z   = 0008          M.ATA  = 0080          M.CAPE= 0020
M.CONT= 0080           M.DEM  = 0020          M.EBL  = 0010          M.EXC  = 0008
M.FAIL= 0008           M.ILR  = 0010          M.INIT= 0010           M.OCC  = 0020
M.ONLI= 0001           M.PE   = 0040          M.PORT= 0080           M.RDEN= 0002
M.RDPE= 0008           M.RUN  = 0004          M.SCLK= 0001           M.TRA  = 0040
M.UNIT= 0007           M.WCLK= 0040           M.WCLN= 0080           M.WREN= 0080
M5.5   = 0001          M6.5   = 0002          M7.5   = 0004          NOTCAP= 0088
OKAY   = 00FF          OPRRAM= 4300           OPSTRT= 0058           OPVER  = 0040
PADCNT= 00D5           PADCRC= 0080           PDIAG  = 0048          PEGCOM  4571
PEID   = 008A          PENAB  = 004C          PESET  = 0001          PESUB   4565
PGCTL   463E           PL     = 00B1          PRDD   = 004C          PRENF  = 009C
PS     = 00B2          PSTAT  = 0048          PSW    =%0009          P.AMTP= 0001
P.BCTC= 0040           P.CMDP= 0020           P.INTE= 0080           P.LCS  = 0040
P.LWR  = 0020          P.RDP  = 0002          P.RPEN= 0004           P.RPST= 0002
P.RPOE= 0020           P.RP1E= 0010           P.RP2E= 0020           P.RP3E= 0010
P.SING= 0080           P.STAT= 0002           P.STPE= 0080           P.TACH= 0008
P.TUPR= 0010           P.WCSP= 0004           P.WDS  = 0040          P.WFLP= 0001
P.WPEN= 0010           P.WPOE= 0008           P.WP1E= 0004           P.WP2E= 0008
P.WP3E= 0004           P.5VOK= 0002           QUE    = 281B          QUEM   = 281E
RAMT   = 0010          RARA   = 0006          RARAI  = 0004          RCHBD0= 0048
RCHBD1= 0047           RCHOK  = 0046          RCHTST= 000C           RCLRT  = 000D
RCMD   = 000B          RCMLP  = 0003          RCONT  = 0080          RDATA  = 0017
RDCLK  = 0010          RDON   = 0011          READG  = 0007          REND   = 0014
REQST  = 2806          RESCHR= 00D1           REVTST= 0064           REWIND= 0004
RFIFOL= 0008           RGCLK  = 0002          RGCRI  = 0003          RIBG   = 0001
RILL   = 0012          RINST  = 000C          RMCTST= 0008           RMK2   = 0013
RNOP   = 0000          RPATH  = 0001          RPBAD  = 0044          RPCHI  = 0001
RPCLK  = 0003          RPCTL  = 0009          RPEI   = 0002          RPFAIL= 0000
RPF1   = 009D          RPF2   = 009E          RPOK   = 0043          RPOSTN= 0016
RPSTA  = 0015          RRCMT  = 000A          RSTAT  = 0002          RTIEB  = 000A
RTIER  = 0030          RTM    = 0005          RUNKI  = 0009          RUPTST= 005E
RWDUNL= 0005           R.AMT  = 0001          R.BOP  = 0008          R.DATA= 0040
R.DON  = 0002          R.DRDY= 0010           R.END  = 0010          R.ILL  = 0004
R.JVOK= 0004           R.MK2  = 0008          R.PLOD= 0008           R.PLOO= 0010
R.PLO1= 0020           R.POST= 0020           R.STNM= 0002           R.STOP= 0004
R.STPC= 0001           R.TBJN= 0080           R.TSTD= 0040           R.VOK  = 0080
R00H   = 0081          R00L   = 0080          R01H   = 0083          R01L   = 0082
R02H   = 0085          R02L   = 0084          R03H   = 0087          R03L   = 0086
R04H   = 0089          R04L   = 0088          R05H   = 008B          R05L   = 008A
```

```
R06H  = 008D          R06L  = 008C          R07H  = 008F          R07L  = 008E
R10H  = 0091          R10L  = 0090          R11H  = 0093          R11L  = 0092
R12H  = 0095          R12L  = 0094          R13H  = 0097          R13L  = 0096
R14H  = 0099          R14L  = 0098          R15H  = 009B          R15L  = 009A
R16H  = 009D          R16L  = 009C          R17H  = 009F          R17L  = 009E
R7.5  = 0010          SELCLR= 0000          SETATA= 00A1          SID   = 0080
SIZE    4640          SOD   = 0080          SOE   = 0040          SP    =%0008
SPEED   4306          SSCLK = 0040          SSTEP = 0005          STACK = 4FFF
STATRM= 4F20          STPCT   4F20          STRSP = 5000          TADR00= 0080
TADR01= 0081          TADR02= 0082          TADR03= 0083          TADR04= 0084
TADR05= 0085          TADR06= 0086          TADR07= 0087          TADR10= 0088
TADR11= 0089          TADR12= 008A          TADR13= 008B          TAMT  = 0044
TASEL = 0080          TCMD  = 0040          TC.FWD= 0040          TC.INH= 0008
TC.LWR= 0004          TC.REV= 0020          TC.WRT= 0010          TEMP    4F99
TEST02  4366          TEST03  43A6          TEST04  43E6          TEST05  4426
TEST06  4466          TEST07  44A6          TEST1   4307          TMF   = 0099
TMRDY = 0040          TRKENA= 00D2          TSET  = 2803          TSTEND= 2818
TSTS  = 0040          TST01X  432B          TST01Y  4331          TST02X  4371
TST03X  43B1          TST04X  43F1          TST05X  4431          TST06X  4471
TUSEL0= 00D1          TUSEL1= 00D2          TU78  = 0010          T.ATH0= 0001
T.ATH1= 0002          T.BOT = 0004          T.EOT = 0002          T.FPT = 0001
T.NTHR= 0004          T.ONL = 0020          T.PES = 0008          T.PSBJ= 0020
T.PSOJ= 0008          T.PS1J= 0010          T.RDY = 0080          T.RDY0= 0040
T.RWD = 0010          T.SCLK= 0002          T1CN1   460C  G       T1LO1   4598
T2LO1   44EC  G       T2LO2   4502  G       T2LO3   4519  G       T2LO4   4530  G
T2LO5   4548  G       UIBG  = 00A1          UNITMP  463F          VALFC   4F98
VALTB   4F95          VELCOM  44F2          VELTST= 005B          WDR.P = 0010
WMCCTL= 00D3          WMCERR= 00DA          WMCSTA= 00D0          WRTCLK= 0000
WRTDAT= 00D3          W.ACRC= 0004          W.CRC = 0008          W.DIAG= 0002
W.DONN= 0040          W.ECC = 0010          W.ENAB= 0080          W.ERR = 0020
W.FMT = 0070          W.GCR = 0010          W.LEFT= 0004          W.ONES= 0020
W.RES1= 0002          W.REV = 0004          W.ROME= 0010          W.RST = 0001
W.SKIP= 000F          W.WRIT= 0008          W.XFER= 0020          X     =%000A
X.DONN= 0080          X.ENAB= 0040          X.PEPE= 0002          X.ROME= 0001
X.WCLK= 0001          Y     =%000B          .     = 4646


 ERRORS DETECTED:  0

 *MTA4.A78/PTP,MTA4-NLIST,PARAM,MACRO,LIST,MTA4
 RUN-TIME: 4 5 0 SECONDS
```