DUV/LSI-11

COMB TIMING & INTER TEST

MD-11-DZDUU-A

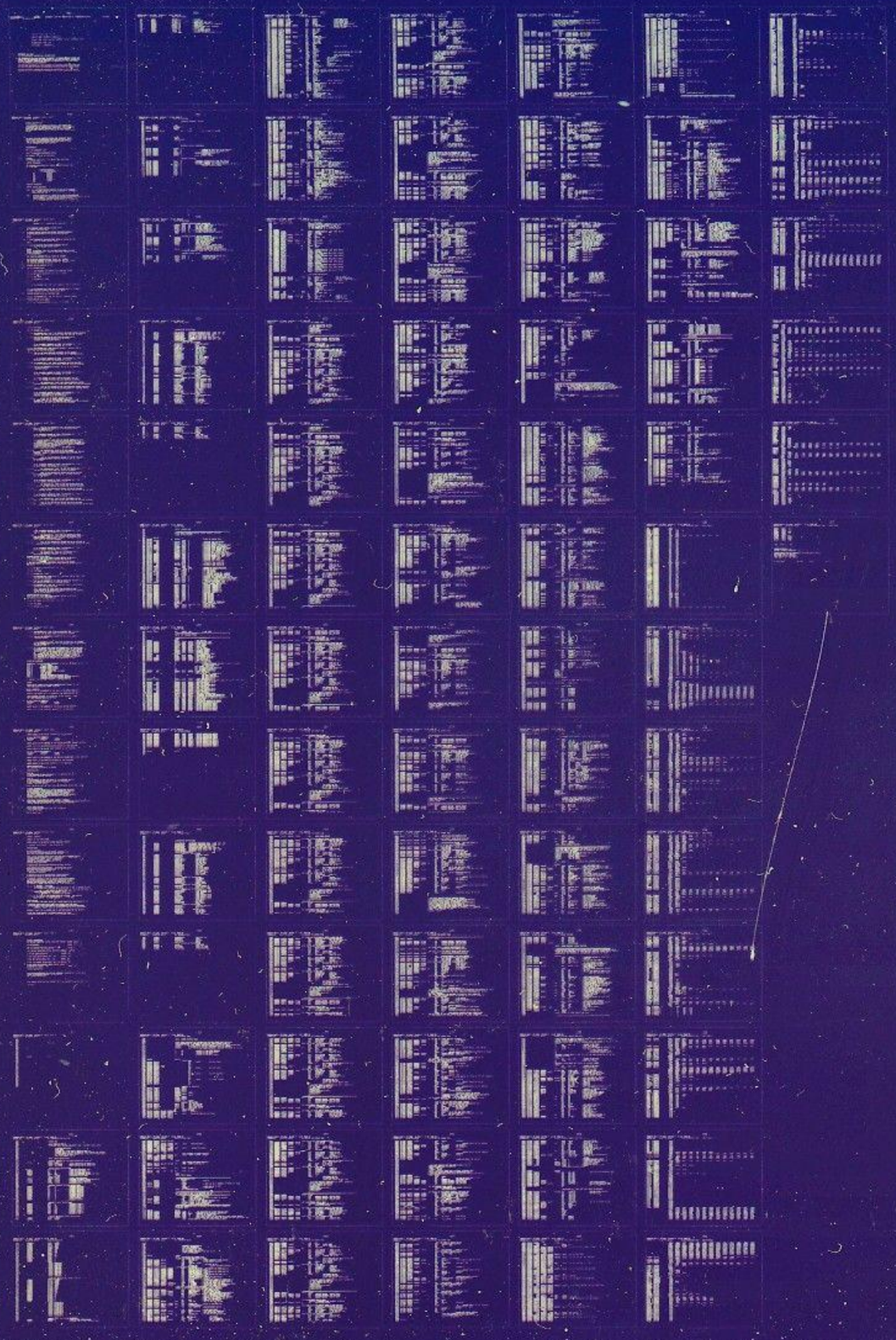
EP-DZDUU-A-DL-A

FICHE 1 OF 1

APR 1977

digital

MADE IN USA



EOF1DZD[illegible]411 [illegible] MA[illegible]327(1006) 03-F[illegible]074[illegible]8:00 PAG[illegible]DR1DZDUUASEQ 00010000 770323
DZDUU1.M11 02-FEB-77 08:18

.REM *

I D E N T I F I C A T I O N

PRODUCT CODE: MAINDEC-11-DZDUU-A-D

PRODUCT NAME: DUV11 OFFLINE COMBINED TIMING & INTERRUPT TESTS

RELEASE DATE: FEB. 1977

MAINTAINER : DIAGNOSTICS

*
.REM *

*

.REM *

GENERAL DESCRIPTION

THIS DIAGNOSTIC CAN CHAIN 16 DUV11'S. THIS MEANS THAT
16 DEVICES CAN BE SEQUENTIALLY EXERCISED. THE DIAGNOSTIC
MAKES ONE PASS BEFORE PROCEEDING TO THE NEXT DEVICE,
AND CONTINUES EXERCISING ALL DEVICES IN THIS FASHION UNTIL
HALTED.

*
.REM *

1. THE DUV11 OFFLINE COMBINED TIMING AND INTERRUPT TESTS
VERIFY THAT THE TRANSMITTER AND RECEIVER CAN CORRECTLY
TALK TO EACH OTHER.INTERRUPT LOGIC AND PRIORITY LEVEL
/VECTOR ADDRESSES ARE ALSO VERIFIED

*
.REM *

2. REQUIREMENTS

PDP-11/03 COMPUTER (LSI)

DUV11 SYNCRONOUS/ISOCRONOUS OPTION

ONE CONSOLE TELETYPE OR EQUIVALENT

2.2 STORAGE
THE PROGRAM LOADS INTO 4K OF MEMORY WITH BOOTSTRAP

3. LOADING PROCEDURE

THE STANDARD PROCEDURE FOR LOADING ABSOLUTE BINARY TAPES
IS TO BE USED.

	STARTING ADDRESS FOR ABSOLUTE LOADER
4K	017500
8K	037500
12K	057500
16K	077500
20K	117500
24K	137500
28K	157500

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

NOTE: ALL SWITCHES RESIDE INTERNAL TO THE CPU AT ADDRESS
176. THESE MAY BE SET VIA THE CONSOLE TTY BY DIRECTLY
MODIFYING LOC. 176.

NOTE: RUNNING UNDER APT-11, THERE IS A USER SWITCH REGISTER
CALLED "SUSWR". IN ORDER TO BE FLEXIBLE ON THE AVAILIBILITY OF THE
H315 CONNECTOR, ONE BIT PASSES STATUS TO APT-11.
BIT 0 IN SUSWR REFLECTS THIS STATUS, A 0 = CONNECTOR

PRESENT, A 1 = CONNECTOR NOT AVAILIBLE.

4.1.1 AFTER PROGRAM LOAD (INITIAL PROGRAM START)
ALL CONSOLE SWITCHES DOWN

4.1.2 TO MODIFY DEVICE VECTOR AND CONTROL REGISTER ADDRESSES
AFTER PROGRAM RESTART OR TO RUN MULTIPLE DEVICES

SW00=1

4.1.3 TO START PROGRAM AT SELECTED TEST AFTER A PROGRAM RESTART
(ONLY IN SINGLE DEVICE TESTS)
SW01=1

4.1.4 TO LOCK ON SELECTED TEST AFTER A PROGRAM RESTART
(ONLY IN SINGLE DEVICE TESTS)

SW02=1
NOTE1: IN GENERAL SW01 WILL BE USED WHEN SW02=1 IS USED
NOTE2: WITHOUT SW01=1 "LOCK ON TEST" WILL DEFAULT TO TEST 1

4.2 STARTING ADDRESS

THE STARTING ADDRESS FOR ALL TESTS IS 000200

THE RETARTING ADDRESS FOR ALL TESTS IS 000200
THE STARTING ADDRESS TO ENTER A SELECTED TEST IS 000200
THE STARTING ADDRESS TO LOCK ON TEST IS 000200

4.3 PROGRAM AND/OR OPERATOR ACTION

4.3.1 INITIAL PROGRAM START

4.3.1.1 LOAD PROGRAM INTO MEMORY WITH ABSOLUTE LOADER

4.3.1.2 SET SWITCH REGISTER (LOC. 176) TO ZERO.

4.3.1.3 TYPE 200G.

4.3.1.4 PROGRAM WILL START.

*
.REM *

4.3.1.5 THE PROGRAM WILL TYPE "DUV11 DZDUU-A TAPE E" (ONCE ONLY)

*
.REM *

4.3.1.6 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT IS ABOUT
TO START TESTING ,AND THEN TESTING WILL BEGIN

4.3.2 PROGRAM RESTART WITH ALL SWITCHES DOWN

4.3.2.1 THE PROGRAM WILL TYPE "R" AND WILL COMMENCE TESTING

4.3.3 PROGRAM RESTART WITH SW00=1

4.3.3.1 SET SWITCH REGISTER (LOC. 176) TO A 000001.

4.3.3.2 TYPE 200G.

4.3.3.3 PROGRAM WILL START.

4.3.3.4 THE PROGRAM WILL TYPE " 1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.5 TYPE IN THE ADDRESS OF THE FIRST RECEIVER CONTROL REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ADDRESS IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.4

4.3.3.6 THE PROGRAM WILL TYPE "VECTOR ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.7 TYPE IN THE BASE RECEIVER INTERRUPT VECTOR ADDRESS FOR THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ADDRESS IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.6

4.3.3.8 THE PROGRAM WILL TYPE "ARE YOU RUNNING MULTIPLE DEVICES ?" (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.9 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS GIVEN, THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.8

IF A "NO" ANSWER IS GIVEN: JUMP TO SECTION 4.3.3.12
IF A "YES" ANSWER IS GIVEN:THE NEXT QUESTION IS ASKED

4.3.3.10 THE PROGRAM WILL TYPE "LAST DEVICE:RECEIVER CONTROL REGISTER ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.11 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.10
NOTE:ALL ADDRESSES SHALL BE CONTIGUOUS

4.3.3.11.1 IF AN "OUT OF RANGE" ADDRESS IS TYPED IE. MORE THAN 16 (10) DEVICES AWAY (UPWARDS).....THE PROGRAM WILL TYPE "OUT OF RANGE:RETYPE LAST DEVICE RXCSR ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.11.2 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE MESSAGE OF 4.3.3.11.1

IF A DEVICE ADDRESS LOWER THAN 1ST DEVICE ADDRESS IS TYPED.....
....SCHOOLS OUT..... ..THERE IS NO PROTECTION FOR THIS.
THE PROGRAM WILL DEFAULT TO TWO DEVICES ACTIVE (UPWARDS FROM 1ST DEVICE ADDRESS).THE SAME APPLIES TO IDENTICAL ADDRESSES TYPED FOR FIRST AND LAST DEVICE.
OBSERVE LOCATION @ ACTREG: SEE SECTION 7.2

4.3.3.12 THE PROGRAM WILL TYPE "# OF SYNC CHARS SELECTED (1 OR 2)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD. REFER TO MANUAL FOR PROPER SWITCH SETTINGS OF SWITCH E55-4.

4.3.3.13 TYPE IN THE APPROPRIATE ANSWER "1" OR "2" FOLLOWED BY A <CARRIAGE RETURN>.(NOTE:ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE MESSAGE OF 4.3.3.12

4.3.3.14 THE PROGRAM WILL TYPE " IS SEC XMIT SWITCH E55-2 ON? (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.15 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE RETURN>.(NOTE THAT ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE MESSAGE OF 4.3.3.14

4.3.3.16 THE PROGRAM WILL TYPE "IS SEC REC SWITCH E55-3 ON? (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.17 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE MESSAGE OF 4.3.3.16

4.3.3.18 THE PROGRAM WILL TYPE "IS OPT CLR ENABLE SWITCH E55-1 ON? (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.19 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"
AND WILL REPEAT THE MESSAGE OF 4.3.3.18

4.3.3.20 THE PROGRAM WILL TYPE "ARE YOU RUNNING IN MAINT.
MODE EXTERNAL ? AND .....DO YOU HAVE THE EXTERNAL MODEM
BYPASS JUMPER CONNECTOR ON ? (Y OR N)-" AND WAIT FOR AN
INPUT FROM THE TELETYPE KEYBOARD

4.3.3.21 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY
A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"
AND WILL REPEAT THE MESSAGE OF 4.3.3.20

4.3.3.22 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT
HAS STARTED AND WILL COMMENCE TESTING AT TEST 1

4.3.4 PROGRAM RESTART WITH SW01=1
NOTE: THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED
,,,IT WILL NOT WORK IF MULTIPLE DEVICES ARE SELECTED

IF MULTIPLE DEVICES WERE PREVIOUSLY SELECTED,LOAD 000200,
AND SELECT SW00=1 AND ANSWER "NO" TO THE MULTIPLE DEVICE QUESTION
SEE 4.3.3

4.3.4.1 SET SW01=1 IN SWITCH REG (LOC. 176)

4.3.4.2 TYPE 200G.

4.3.4.3 PROGRAM WILL START.

4.3.4.4 THE PROGRAM WILL TYPE "TEST PC-" AND WAIT FOR AN INPUT FROM
THE TELETYPE KEYBOARD

4.3.4.5 TYPE IN THE ADDRESS OF THE TEST AT WHICH THE PROGRAM IS TO
BE STARTED FOLLOWED BY A <CARRIAGE RETURN>

4.3.4.6 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT HAS STARTED
TESTING AT THE SELECTED TEST

NOTE: CARE MUST BE TAKEN WHEN THIS FEATURE IS USED
,SINCE THERE IS NO PROTECTION AGAINST SELECTING AN ADDRESS
THAT IS IN THE MIDDLE OF A TEST

4.3.5 PROGRAM RESTART WITH SW02 =1
NOTE: THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED
SEE NOTE IN 4.3.4 FOR MORE DETAILS

4.3.5.1 SET SW02=1 IN SWITCH REG. (LOC. 176)

4.3.5.2 TYPE 200G.

4.3.5.3 PROGRAM WILL START.

4.3.5.4 THE PROGRAM WILL TYPE "LOCK ON SELECTED TEST ? (Y OR N)-"
AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.5.5 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A
<CARRIAGE RETURN>

IF A NO ANSWER IS GIVEN: THIS LOCK ON TEST WILL BE IGNORED
AND THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT HAS STARTED
TESTING AT TEST 1

4.3.5.6 IF A YES ANSWER WAS GIVEN:THE PROGRAM WILL ACT AS FOLLOWS...
THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT HAS STARTED
TESTING AT TEST 1 AND WILL REMAIN IN TEST 1 UNTIL HALTED
OR IF ANY KEY IS STRUCK ON THE TELETYPE ,THE PROGRAM
WILL FREEZE ON THE NEXT TEST UNTIL A KEY IS STRUCK ON
THE TELETYPE AND SO FORTH THRU THE PROGRAM. IF SW01 =1 IT
WILL PERFORM AS IN SECTION 4.3.4 ALLOWING ONE TO FREEZE
ON A SELECTED TEST RATHER THAN DEFAULTING TO TEST 1

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS (INTERNAL TO THE CPU, ACCESSED VIA LOC. 176).

SW15 =1	HALT ON ERROR
SW14 =1	LOOP ON CURRENT TEST
SW13 =1	INHIBIT ERROR TYPEOUT
SW11 =1	INHIBIT ITERATIONS
SW10 =1	ESCAPE TO NEXT TEST ON ERROR
SW09 =1	LOOP ON ERROR
SW02 =1	LOCK ON TEST
SW01 =1	RESTART PROGRAM AT SELECTED TEST
SW00 =1	RESELECT VECTOR AND CONTROL REGISTER ADRESSES &PARAMETERS AFTER A PROGRAM RESTART

TO INHIBIT "END OF PASS" TYPEOUT - TURN TELETYPE OFF

6. ERRORS

6.1 ERROR HALTS (UNDER LSI ALL HALT ERRORS RETURN CONTROL TO O.D.T.)
THERE ARE FOUR DISTINCT ERROR TYPEOUTS

6.1.1 PC+2 = ERROR PC
WHERE PC +2 IS THE ADDRESS OF THE CALL TO THE ERROR HANDLER +2

REFER TO THE ABOVE "HLT" IN DIAGNOSTIC FOR ERROR DESCRIPTION

CHECK ADDRESS @ RXCSR: TO LOCATE THE DEVICE PRESENTLY UNDER
TEST WHEN RUNNING MULTIPLE DEVICES

6.1.2 PC +2 = REGISTER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING DEVICE REGISTER

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.3 PC +2 = RECEIVER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING RECEIVER (RXDBUF) REGISTER

WHERE YYYYYY IS THE EXPECTED DATA CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL DATA CONTENTS OF THAT REGISTER

6.1.4 PC +2 = TRANSMITTER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING TRANSMITTER (TXCSR) REGISTER

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.5 ERROR DESCRIPTIONS
SEE LISTINGS FOR DETAILS OF ERRORS

6.2 ERROR RECOVERY

6.2.1 SW15 =0
IF THE PROGRAM IS RUN WITH SW15 =0 ,NO OPERATOR ACTION IS
REQUIRED TO CONTINUE TESTING

6.2.2 SW15 =1
IF THE PROGRAM IS RUN WITH SW15 =1 ,TO CONTINUE TESTING
AFTER THE PROGRAM HAS HALTED ,PRESS THE PROCESSOR
CONSOLE "CONTINUE SWITCH"

NOTE: THE PC + 2 OF THE "HLT" WILL BE DISPLAYED IN THE DATA LIGHTS

6.2.3 ILLEGAL INTERRUPTS
IF AN INTERRUPT OCCURS TO A VECTOR ADDRESS NOT SELECTED
DURING PROGRAM INITIALIZATION, THE PROGRAM WILL HALT IN
THE TRAPCATCHER. THE ADDRESS AT WHICH THE PROGRAM
HALTS IS 2 GREATER THAN THE ADDRESS TO WHICH THE INTERRUPT
OCCURED. THE PROGRAM MUST BE RESTARTED AT 000200 TO
RECOVER FROM THIS ERROR.

6.2.4 ADDITIONAL TROUBLESHOOTING AIDS ERRCNT: & PASCNT:
CHECK THESE TWO TAG LOCATIONS FOR TOTAL # OF ERRORS AND PASSES RESPECTIVELY.
LOADING 000200 AND RESTARTING WILL CLEAR THESE LOCATIONS.

6.3 END OF PASS ROUTINE
THIS TYPEOUT IS MENTIONED HERE FOR CONVENIENCE
IT IS IN THE FORM:

END OF PASS TAPE Y

16XXXX = DEVICE

WHERE Y IS THE TAPE LOADED

WHERE 16XXXX IS THE DEVICE'S BASE REGISTER ADDRESS

TO INHIBIT THIS TYPEOUT - TURN TELETYPE OFF

7. RESTRICTIONS

7.1 MULTIPLE DEVICES

UP TO 16(10) DEVICES MAY BE TESTED. HOWEVER, THEY
MUST HAVE CONTIGUOUS ADDRESSES AND VECTORS

NOTE: IF ALL DEVICES UNDER TEST HAVE THE SAME INTERRUPT VECTOR
YOU CAN CHANGE "ZERO: ADD #10,BASEIV ;NEXT BLOCK
(VECTORS)" TO "ZERO: ADD #0,BASEIV";
THEREBY THE VECTOR ADDRESSES WILL NOT BE
UPDATED AFTER EACH PASS.

7.2 DISQUALIFYING DEVICES WHEN RUNNING MULTIPLE DEVICES

WHEN RUNNING MULTIPLE DEVICES AN ACTIVE BIT IS SET
FOR EACH DEVICE RUNNING UNDER TEST IE. BIT 0 FOR
DEVICE 0 ,BIT 15 FOR DEVICE 15
TO DISQUALIFY DEVICES:

7.2.1 IF DEVICE 0 IS TO BE DISQUALIFIED ,SIMPLY RESTART

PROGRAM WITH SW00 =1 AND OMIT THE FIRST DEVICE.

7.2.2 IF HOWEVER, DEVICES 1 THRU 15 OR ANY COMBINATION THEREOF

ARE TO BE DISQUALIFIED....LOAD THE LOCATION OF ACTREG:
OBSERVE THE ACTIVE BITS (ACTIVE =1, NONACTIVE = 0)
AND DEPOSIT 0 WHERE THOSE DEVICES ARE TO BE DISQUALIFIED

7.2.2.1 TO RESTART...TYPE 200G...
THE PROGRAM WILL CONTINUE WITH THE DEVICE IT WAS IN BEFORE HALTING.

7.2.2.2 .....OR ....SET SW00=1 IN SWITCH REG (LOC. 176) AND TYPE 200G....
ANSWER THE QUESTION :1ST DEVICE : ETC.....
....THE PROGRAM WILL CONTINUE WITH DEVICE 0

7.2.2.3 IF ALL DEVICES ARE DISQUALIFIED BY MISTAKE THE PROGRAM
WILL TYPEOUT AN ERROR MESSAGE.....TYPE 200G.

7.3 CABLE DELAYS

NOTE: EXTERNAL LOOP BACK TESTS ONLY (MODEM CABLE WITH H315 CONNECTOR ON)

7.3.1 TO PROVIDE SUFFICIENT DELAY FOR CLOCK SIGNAL OVER THE CABLE,

LOCATION "HOLD:" MUST BE MODIFIED TO ACCOMODATE FOR FASTER MACHINES.
PRESENTLY "HOLD:" =20 IS SUFFICIENT TIME ON AN 11/03 MACHINE.

BASICALLY DON'T TRY TO EXCEED 10K TO 12K RATE USING THE EIA DRIVERS

7.4 TO USE THE "XOR" TESTER THE BRANCH AROUND THE "XOR"

CODE MUST BE PATCHED TO A "NOP". (SEE LISTINGS FOR DETAILS)

8. DEFAULT PARAMETERS:
1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- RXCSR: 160010

VECTOR ADDRESS- DURIV: 770

ARE YOU RUNNING MULTIPLE DEVICES ?- NO MULTD: 0

LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- LASTADD: 0

OF SYNC CHARS SELECTED - 2 SYNCNO: 377

IS SEC XMIT SWITCH E55-2 ON?- YES SEXMIT: 377

IS SEC REC SWITCH E55-3 ON?- YES SEREC: 377

IS OPT CLR ENABLE SWITCH E55-1 ON?- YES OPTCLR: 377

DO YOU HAVE THE EXTERNAL MODEM BYPASS JUMPER
CONNECTOR ON (H315)- YES JMRBY: 377

9. PROGRAM DESCRIPTION

```
                                                                    *
                                                                .REM    *
```

9.1 THIS PROGRAM PERFORMS THE OFFLINE COMBINED (TRANSMITTER & RECEIVER)
TIMING & INTERRUPT TESTING OF THE DEVICE
SEE LISTING FOR DETAILS

```
                                                                    *
                                                                .REM    *
```

10. FLOW CHARTS: RECEIVER FLOW,TRANSMITTER FLOW,TRANSMITTER & RECEIVER FLOW

11. LISTINGS

```
                                                                    *
```

```
                000001                  STN=1
```

```
DZDUU-A MACY11 27(1006)  03-FEB-77  08:00  PAGE 14
DZDUUA.M11    13-OCT-76 08:28          APT COMMUNICATIONS ROUTINE

                                       .ENABLE ABS

                                       ;DUV11 DZDUU-A TAPE E
                                       ;COPYRIGHT 1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

                                               ;STARTING PROCEDURE
                                               ;TYPE 200G
                                               ;PROGRAM WILL TYPE "DUV11 DZDUU-A TAPE E "
                                               ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
                                               ;AT THE END OF A PASS, PROGRAM WILL TYPE "END OF PASS TAPE E"
                                               ;AND THEN RESUME TESTING

                                       .SBTTL  BASIC DEFINITIONS

                                       ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                                 STACK=  1100
                                       .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                       .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                                       ;*MISCELLANEOUS DEFINITIONS
000011                                 HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                                 LF=     12              ;;CODE FOR LINE FEED
000015                                 CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                                 CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                                 PS=     177776          ;;PROCESSOR STATUS WORD
                                       .EQUIV  PS,PSW
177774                                 STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                                 PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                                 DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                                 DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                                       ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                                 R0=     %0              ;;GENERAL REGISTER
000001                                 R1=     %1              ;;GENERAL REGISTER
000002                                 R2=     %2              ;;GENERAL REGISTER
000003                                 R3=     %3              ;;GENERAL REGISTER
000004                                 R4=     %4              ;;GENERAL REGISTER
000005                                 R5=     %5              ;;GENERAL REGISTER
000006                                 R6=     %6              ;;GENERAL REGISTER
000007                                 R7=     %7              ;;GENERAL REGISTER
000006                                 SP=     %6              ;;STACK POINTER
000007                                 PC=     %7              ;;PROGRAM COUNTER

                                       ;*PRIORITY LEVEL DEFINITIONS
000000                                 PR0=    0               ;;PRIORITY LEVEL 0
000040                                 PR1=    40              ;;PRIORITY LEVEL 1
000100                                 PR2=    100             ;;PRIORITY LEVEL 2
000140                                 PR3=    140             ;;PRIORITY LEVEL 3
000200                                 PR4=    200             ;;PRIORITY LEVEL 4
000240                                 PR5=    240             ;;PRIORITY LEVEL 5
000300                                 PR6=    300             ;;PRIORITY LEVEL 6
000340                                 PR7=    340             ;;PRIORITY LEVEL 7

                                       ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                                 SW15=   100000
040000                                 SW14=   40000
```

```
DZDUU-A MACY11 27(1006)  03-FEB-77  08:00  PAGE 15
DZDUUA.M11    13-OCT-76 08:28          BASIC DEFINITIONS

                        020000          SW13=   20000
                        010000          SW12=   10000
                        004000          SW11=   4000
                        002000          SW10=   2000
                        001000          SW09=   1000
                        000400          SW08=   400
                        000200          SW07=   200
                        000100          SW06=   100
                        000040          SW05=   40
                        000020          SW04=   20
                        000010          SW03=   10
                        000004          SW02=   4
                        000002          SW01=   2
                        000001          SW00=   1
                                        .EQUIV  SW09,SW9
                                        .EQUIV  SW08,SW8
                                        .EQUIV  SW07,SW7
                                        .EQUIV  SW06,SW6
                                        .EQUIV  SW05,SW5
                                        .EQUIV  SW04,SW4
                                        .EQUIV  SW03,SW3
                                        .EQUIV  SW02,SW2
                                        .EQUIV  SW01,SW1
                                        .EQUIV  SW00,SW0

                                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                        100000          BIT15=  100000
                        040000          BIT14=  40000
                        020000          BIT13=  20000
                        010000          BIT12=  10000
                        004000          BIT11=  4000
                        002000          BIT10=  2000
                        001000          BIT09=  1000
                        000400          BIT08=  400
                        000200          BIT07=  200
                        000100          BIT06=  100
                        000040          BIT05=  40
                        000020          BIT04=  20
                        000010          BIT03=  10
                        000004          BIT02=  4
                        000002          BIT01=  2
                        000001          BIT00=  1
                                        .EQUIV  BIT09,BIT9
                                        .EQUIV  BIT08,BIT8
                                        .EQUIV  BIT07,BIT7
                                        .EQUIV  BIT06,BIT6
                                        .EQUIV  BIT05,BIT5
                                        .EQUIV  BIT04,BIT4
                                        .EQUIV  BIT03,BIT3
                                        .EQUIV  BIT02,BIT2
                                        .EQUIV  BIT01,BIT1
                                        .EQUIV  BIT00,BIT0

                                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
                        000004          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
                        000010          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
```

```
   669          000014                  TBITVEC=14              ;;"T" BIT
   670          000014                  TRTVEC= 14              ;;TRACE TRAP
   671          000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
   672          000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
   673          000024                  PWRVEC= 24              ;;POWER FAIL
   674          000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
   675          000034                  TRAPVEC=34              ;;"TRAP" TRAP
   676          000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
   677          000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
   678          000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
```

```
                                        ;STANDARD INTERRUPT VECTORS


                000174                  .=174
000174  000000                 DISPREG:0
000176  000000                 SWREG:0
                000200          .=200
000200  000167  001746                  JMP     .START          ;GO TO START OF PROGRAM


                001100          .=1100
001100  000000                 .WORD 0
001102  177570                 LIGHTS:177570


                                        ;PROGRAM CONTROL PARAMETERS

001104  000000                 RETURN: 0
001106  000000                 NEXT:   0                       ;ADDRESS OF NEXT TEST TO BE EXECUTED
001110  000000                 LOCK:   0                       ;ADDRESS FOR LOCK ON CURRENT DATA
001112  000000                 PASCNT: 0               ;ADDRESS CONTAINING PASS COUNT
001114  000000                 ERRCNT: 0                       ;ERROR COUNT
001116  000000                 SAVSP:  0               ;STACK POINTER STORAGE

                                        ;PROGRAM VARIABLES

001120  000020                 HOLD:   20              ;TEMPORARY STORAGE=DELAY TIME FOR CABLES
001122  000000                 SHIFT:  0               ;TEMPORARY STORAGE= # OF SHIFTS PER CHAR
001124  000000                 COUNT:  0               ;TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
001126  000000                 SAVPC:  0                       ;PROGRAM COUNTER STORAGE
001130  000000                 HLD0:   0
001132  000000                 HLD1:   0
001134  000000                 HLD2:   0
001136  000000                 HLD3:   0
001140  000000                 HLD4:   0
001142  000000                 HLD5:   0
001144  000000                 HLD6:   0

```

```
DZDUU-A MACY11 27(1006)  03-FEB-77  08:00  PAGE 18
DZDUUA.M11    13-OCT-76 08:28          BASIC DEFINITIONS

                                       ;PROGRAM CONVERSATIONAL PARAMETERS
001146     377                 SYNCNO: .BYTE   377     ;# OF SYNC CHARS REQ'D FOR SYNC'ZATION
001147     377                 SEXMIT: .BYTE   377     ;SEC XMIT JUMPER "IN"
001150     377                 SEREC:  .BYTE   377     ;SEC REC JUMPER "IN"
001151     377                 OPTCLR: .BYTE   377     ;OPTION JUMPER CLR "IN"
001152     000                 MULTD:  .BYTE   0       ;NO MULTIPLE DEVICE FLAG
001153     377                 JMRBY:  .BYTE   377     ;EXTERNAL MODEM BYPASS JUMPER "IN"
                               .EVEN

                                       ;PROGRAM MULTIPLE DEVICE PARAMETERS
001154  000000                 BASEADD:        0       ;PROG CONTROLLED 1ST DEVICE ADDR
001156  000000                 KEEPADD:        0       ;SAVED 1ST DEVICE ADDR
001160  000000                 LASTADD:        0       ;LAST DEVICE RXCSR ADDR
001162  000000                 BASEIV:         0       ;PROG CONTROLLED IV
001164  000000                 KEEPIV:         0       ;SAVED INTR VECTOR
001166  000000                 ACTREG:         0       ;ACTIVE REGISTER   MODIFY THIS
                                                       ;LOCATION TO DISQUALIFY OR QUALIFY
                                                       ;DEVICES (1= RUN,  0= DON'T RUN)
001170  000000                 ROTADD:         0       ;ROTATING POINTER FOR ACTREG..POINTS
                               ;TO DEVICE PRESENTLY UNDER TEST WHEN RUNNING MULTIPLE DEVICES

                                       ;PROGRAM CONTROL FLAGS

001172     000                 INIFLG: .BYTE   0               ;PROGRAM INITIALIZATION FLAG
001173     000                 STFLG:  .BYTE   0               ;TEST START FLAG
001174     000                 LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
           001176              .EVEN
           001400              .=1400
```

```
                                ;INSTRUCTION DEFINITIONS

005746                          PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
005726                          POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
010046                          PUSHR0=10046    ;SAVE R0 ON STACK =MOV R0,-(SP)
012600                          POPR0=12600     ;RESTORE R0 FROM STACK =MOV (SP)+,R0
024646                          PUSH2SP=24646   ;DECREMENT STACK TWICE =CMP -(SP),-(SP)
022626                          POP2SP=22626    ;INCREMENT STACK TWICE =CMP (SP)+,(SP)+
                                ;REGISTER DEFINITIONS
                                        ;RXCSR BIT DEFINITIONS
100000                          DSC=BIT15               ;DATA SET CHANGE
040000                          RING=BIT14              ;RING
020000                          CTS=BIT13               ;CLR TO SEND
010000                          CARDET=BIT12            ;CARRIER DETECT
004000                          RECACT=BIT11            ;REC ACTIVE
002000                          SRD=BIT10               ;SEC REC DATA
001000                          DSR=BIT9                ;DATA SET RDY
000400                          STPSYN=BIT8             ;STRIP SYNC
000200                          RXDONE=BIT7             ;REC DONE
000100                          RINTEN=BIT6             ;REC INTR ENABLE
000040                          DSINTE=BIT5             ;DSC INTR ENABLE
000020                          SYNSCH=BIT4             ;SYNC SEARCH
000010                          STD=BIT3                ;SEC XMIT DATA
000004                          RTS=BIT2                ;REQ TO SEND
000002                          DTR=BIT1                ;DATA TERM RDY
000001                          VOID=BIT0
                                        ;RXDBUF BIT DEFINITIONS
100000                          RXERR=BIT15             ;REC ERROR
040000                          OVRRUN=BIT14            ;OVERRUN
020000                          FRMERR=BIT13            ;FRAME ERROR
010000                          PARER=BIT12             ;PARITY ERROR
                                        ;PARCSR BIT DEFINITIONS
001000                          PAREN=BIT9              ;PARITY ENABLE
000400                          EVPAR=BIT8              ;EVEN PARITY SENSE
                                        ;PARCSR WRD DEFINITIONS
030000                          SYNINT=30000            ;SYNC EXTERNAL MODE
020000                          SYNEXT=20000            ;SYNC INTERNAL MODE
000000                          ISYMOD=0                ;ISOC MODE
000000                          FIVE=0                  ;WORD LENGTH 5 BITS
002000                          SIX=2000                ;WORD LENGTH 6 BITS
004000                          SEVEN=4000              ;WORD LENGTH 7 BITS
006000                          EIGHT=6000              ;WORD LENGTH 8 BITS
000000                          NOPAR=0                 ;NO PARITY
001000                          ODDPAR=1000             ;ODD PARITY
001400                          EVEPAR=1400             ;EVEN PARITY
                                        ;TXCSR BIT DEFINITIONS
100000                          DNA=BIT15               ;DATA NOT AVAILABLE
040000                          MTDATA=BIT14            ;MAINT DATA
020000                          CLK=BIT13               ;CLK
002000                          BITW=BIT10              ;BIT WINDOW
000400                          MRESET=BIT8             ;MASTER RESET
000200                          TXDONE=BIT7             ;XMIT DONE
000100                          TXINTE=BIT6             ;XMIT INTR ENABLE
```

BASIC DEFINITIONS

```
805          000040                  DNAINTE=BIT5                ;DNA INTR ENAB
806          000020                  SEND=BIT4                   ;SEND
807          000010                  HDXEN=BIT3                  ;HDX/FDX
808          000001                  BREAK=BIT0                  ;BREAK
809                                          ;TXCSR WRD DEFINITIONS
810          000000                  USER=0                      ;USER MODE
811          004000                  MINT=4000                   ;MAINT INT MODE
812          010000                  MEXT=10000                  ;MAINT EXT MODE
813          014000                  SYSTST=14000                ;SYSTEM TEST MODE
```

```
                                .SBTTL  COMMON TAGS

                                ;;*****************************************************************
                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                ;*USED IN THE PROGRAM.

          001400                .=.
001400                  $CMTAG:                         ;;START OF COMMON TAGS
001400  000000                  .WORD   0
001402     000          $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001403     000          $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001404  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001406  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001410  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001412  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001414     000          $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001415     001          $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001416  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001420  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001422  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001424  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001426  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001430  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001432  000000                  .WORD   0
001434     000          $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001435     000          $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001436  000000                  .WORD   0
001440  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001442  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001444  177560          $TKS:   177560                  ;;TTY KBD STATUS
001446  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001450  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001452  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001454     000          $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001455     002          $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001456     012          $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001457     000          $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001460  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001462  000000          $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001464  000000          $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001466  000000          $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001470  000000          $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001472  000000          $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001474  000000          $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001476  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001500  000000          $TMP1:  .WORD   0               ;;USER DEFINED
001502  000000          $TMP2:  .WORD   0               ;;USER DEFINED
001504  000000          $TMP3:  .WORD   0               ;;USER DEFINED
001506  000000          $TMP4:  .WORD   0               ;;USER DEFINED
001510  000000          $TMP5:  .WORD   0               ;;USER DEFINED
001512  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001514  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001516  177607  000377    $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001522     077          $QUES:  .ASCII  /?/             ;;QUESTION MARK
001523     015          $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
```

```
DZDUU-A MACY11 27(1006)  03-FEB-77  08:00  PAGE 22
DZDUUA.M11    13-OCT-76 08:28          COMMON TAGS

   870  001524  000012                 $LF:    .ASCIZ  <12>              ;;LINE FEED
   871                                 ;;*****************************************************************
   872                                 .SBTTL  APT MAILBOX-ETABLE
   873
   874                                 ;;*****************************************************************
   875                                 .EVEN
   876  001526                         $MAIL:                            ;;APT MAILBOX
   877  001526  000000                 $MSGTY: .WORD   AMSGTY             ;;MESSAGE TYPE CODE
   878  001530  000000                 $FATAL: .WORD   AFATAL             ;;FATAL ERROR NUMBER
   879  001532  000000                 $TESTN: .WORD   ATESTN             ;;TEST NUMBER
   880  001534  000000                 $PASS:  .WORD   APASS              ;;PASS COUNT
   881  001536  000000                 $DEVCT: .WORD   ADEVCT             ;;DEVICE COUNT
   882  001540  000000                 $UNIT:  .WORD   AUNIT              ;;I/O UNIT NUMBER
   883  001542  000000                 $MSGAD: .WORD   AMSGAD             ;;MESSAGE ADDRESS
   884  001544  000000                 $MSGLG: .WORD   AMSGLG             ;;MESSAGE LENGTH
   885  001546                         $ETABLE:                          ;;APT ENVIRONMENT TABLE
   886  001546     000                 $ENV:   .BYTE   AENV               ;;ENVIRONMENT BYTE
   887  001547     000                 $ENVM:  .BYTE   AENVM              ;;ENVIRONMENT MODE BITS
   888  001550  000000                 $SWREG: .WORD   ASWREG             ;;APT SWITCH REGISTER
   889  001552  000000                 $USWR:  .WORD   AUSWR              ;;USER SWITCHES
   890  001554  000000                 $CPUOP: .WORD   ACPUOP             ;;CPU TYPE,OPTIONS
   891                                 ;*                                BITS 15-11=CPU TYPE
   892                                 ;*                                        11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
   893                                 ;*                                        11/70=06,PDQ=07,Q=10
   894                                 ;*                                BIT 10=REAL TIME CLOCK
   895                                 ;*                                BIT  9=FLOATING POINT PROCESSOR
   896                                 ;*                                BIT  8=MEMORY MANAGEMENT
   897  001556     000                 $MAMS1: .BYTE   AMAMS1             ;;HIGH ADDRESS,M.S. BYTE
   898  001557     000                 $MTYP1: .BYTE   AMTYP1             ;;MEM. TYPE,BLK#1
   899                                 ;*                                MEM.TYPE BYTE   --  (HIGH BYTE)
   900                                 ;*                                        900 NSEC CORE=001
   901                                 ;*                                        300 NSEC BIPOLAR=002
   902                                 ;*                                        500 NSEC MOS=003
   903  001560  000000                 $MADR1: .WORD   AMADR1             ;;HIGH ADDRESS,BLK#1
   904                                 ;*                                 MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
   905  001562     000                 $MAMS2: .BYTE   AMAMS2             ;;HIGH ADDRESS,M.S. BYTE
   906  001563     000                 $MTYP2: .BYTE   AMTYP2             ;;MEM.TYPE,BLK#2
   907  001564  000000                 $MADR2: .WORD   AMADR2             ;;MEM.LAST ADDRESS,BLK#2
   908  001566     000                 $MAMS3: .BYTE   AMAMS3             ;;HIGH ADDRESS,M.S.BYTE
   909  001567     000                 $MTYP3: .BYTE   AMTYP3             ;;MEM.TYPE,BLK#3
   910  001570  000000                 $MADR3: .WORD   AMADR3             ;;MEM.LAST ADDRESS,BLK#3
   911  001572     000                 $MAMS4: .BYTE   AMAMS4             ;;HIGH ADDRESS,M.S.BYTE
   912  001573     000                 $MTYP4: .BYTE   AMTYP4             ;;MEM.TYPE,BLK#4
   913  001574  000000                 $MADR4: .WORD   AMADR4             ;;MEM.LAST ADDRESS,BLK#4
   914  001576  000000                 $VECT1: .WORD   AVECT1             ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
   915  001600  000000                 $VECT2: .WORD   AVECT2             ;;INTERRUPT VECTOR#2BUS PRIORITY#2
   916  001602  000000                 $BASE:  .WORD   ABASE              ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
   917  001604  000000                 $DEVM:  .WORD   ADEVM              ;;DEVICE MAP
   918  001606  000000                 $CDW1:  .WORD   ACDW1              ;;CONTROLLER DESCRIPTION WORD#1
   919  001610  000000                 $CDW2:  .WORD   ACDW2              ;;CONTROLLER DESCRIPTION WORD#2
   920  001612  000000                 $DDW0:  .WORD   ADDW0              ;;DEVICE DESCRIPTOR WORD#0
   921  001614  000000                 $DDW1:  .WORD   ADDW1              ;;DEVICE DESCRIPTOR WORD#1
   922  001616  000000                 $DDW2:  .WORD   ADDW2              ;;DEVICE DESCRIPTOR WORD#2
   923  001620  000000                 $DDW3:  .WORD   ADDW3              ;;DEVICE DESCRIPTOR WORD#3
   924  001622  000000                 $DDW4:  .WORD   ADDW4              ;;DEVICE DESCRIPTOR WORD#4
   925  001624  000000                 $DDW5:  .WORD   ADDW5              ;;DEVICE DESCRIPTOR WORD#5
```

APT MAILBOX-ETABLE

```
926  001626  000000                  $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
927  001630  000000                  $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
928  001632  000000                  $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
929  001634  000000                  $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
930  001636  000000                  $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
931  001640  000000                  $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
932  001642  000000                  $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
933  001644  000000                  $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
934  001646  000000                  $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
935  001650  000000                  $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
936
937
938  001652                          $ETEND:
939
940
941
```

```
                                ;INSTRUCTION DEFINITIONS

005746                          PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
005726                          POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
010046                          PUSHR0=10046    ;SAVE R0 ON STACK =MOV R0,-(SP)
012600                          POPR0=12600     ;RESTORE R0 FROM STACK =MOV (SP)+,R0
024646                          PUSH2SP=24646   ;DECREMENT STACK TWICE =CMP -(SP),-(SP)
022626                          POP2SP=22626    ;INCREMENT STACK TWICE =CMP (SP)+,(SP)+
                                ;REGISTER DEFINITIONS
                                        ;RXCSR BIT DEFINITIONS
100000                          DSC=BIT15                       ;DATA SET CHANGE
040000                          RING=BIT14                      ;RING
020000                          CTS=BIT13                       ;CLR TO SEND
010000                          CARDET=BIT12                    ;CARRIER DETECT
004000                          RECACT=BIT11                    ;REC ACTIVE
002000                          SRO=BIT10                       ;SEC REC DATA
001000                          DSR=BIT9                        ;DATA SET RDY
000400                          STPSYN=BIT8                     ;STRIP SYNC
000200                          RXDONE=BIT7                     ;REC DONE
000100                          RINTEN=BIT6                     ;REC INTR ENABLE
000040                          DSINTE=BIT5                     ;DSC INTR ENABLE
000020                          SYNSCH=BIT4                     ;SYNC SEARCH
000010                          STD=BIT3                        ;SEC XMIT DATA
000004                          RTS=BIT2                        ;REQ TO SEND
000002                          DTR=BIT1                        ;DATA TERM RDY
000001                          VOID=BIT0
                                        ;RXDBUF BIT DEFINITIONS
100000                          RXERR=BIT15                     ;REC ERROR
040000                          OVRRUN=BIT14                    ;OVERRUN
020000                          FRMERR=BIT13                    ;FRAME ERROR
010000                          PARER=BIT12                     ;PARITY ERROR
                                        ;PARCSR BIT DEFINITIONS
001000                          PAREN=BIT9                      ;PARITY ENABLE
000400                          EVPAR=BIT8                      ;EVEN PARITY SENSE
                                        ;PARCSR WRD DEFINITIONS
030000                          SYNINT=30000                    ;SYNC EXTERNAL MODE
020000                          SYNEXT=20000                    ;SYNC INTERNAL MODE
000000                          ISYMOD=0                        ;ISOC MODE
000000                          FIVE=0                          ;WORD LENGTH 5 BITS
002000                          SIX=2000                        ;WORD LENGTH 6 BITS
004000                          SEVEN=4000                      ;WORD LENGTH 7 BITS
006000                          EIGHT=6000                      ;WORD LENGTH 8 BITS
000000                          NOPAR=0                         ;NO PARITY
001000                          ODDPAR=1000                     ;ODD PARITY
001400                          EVEPAR=1400                     ;EVEN PARITY
                                        ;TXCSR BIT DEFINITIONS
100000                          DNA=BIT15                       ;DATA NOT AVAILABLE
040000                          MTDATA=BIT14                    ;MAINT DATA
020000                          CLK=BIT13                       ;CLK
002000                          BITW=BIT10                      ;BIT WINDOW
000400                          MRESET=BIT8                     ;MASTER RESET
000200                          TXDONE=BIT7                     ;XMIT DONE
000100                          TXINTE=BIT6                     ;XMIT INTR ENABLE
```

```
                        APT MAILBOX-ETABLE

   998          000040              DNAINTE=BIT5            ;DNA INTR ENAb
   999          000020              SEND=BIT4               ;SEND
  1000          000010              HDXEN=BIT3              ;HDX/FDX
  1001          000001              BREAK=BIT0              ;BREAK
  1002                                      ;TXCSR WRD DEFINITIONS
  1003          000000              USER=0                  ;USER MODE
  1004          004000              MINT=4000               ;MAINT INT MODE
  1005          010000              MEXT=10000              ;MAINT EXT MODE
  1006          014000              SYSTST=14000            ;SYSTEM TEST MODE
```

```
1007                                    .SBTTL  ERROR POINTER TABLE
1008
1009                                    ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
1010                                    ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
1011                                    ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
1012                                    ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
1013                                    ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
1014
1015                                    ;*      EM              ;;POINTS TO THE ERROR MESSAGE
1016                                    ;*      DH              ;;POINTS TO THE DATA HEADER
1017                                    ;*      DT              ;;POINTS TO THE DATA
1018                                    ;*      DF              ;;POINTS TO THE DATA FORMAT
1019
1020
1021  001652                            $ERRTB:
1022                                            ;ERROR TABLE
1023  001652  001762                            EM1     ;ERROR 1        REGISTER ERROR
1024  001654  002067                            DH1
1025  001656  002116                            DT1
1026  001660  002132                            DF1
1027  001662  002022                            EM2     ;ERROR 2        RECEIVER ERROR
1028  001664  002067                            DH1
1029  001666  002116                            DT1
1030  001670  002132                            DF1
1031  001672  002043                            EM3     ;ERROR 3        TRANSMITTER ERROR
1032  001674  002067                            DH1
1033  001676  002116                            DT1
1034  001700  002132                            DF1
1035  001702  001746                            EM4     ;ERROR 4        BIT ERROR (GENERAL)
1036  001704  000000                            0
1037  001706  002126                            DT4
1038  001710  002132                            DF1
1039
1040                                            ;DEFAULT DU ADDRESSES
1041  001712  160010                    RXCSR:  160010
1042  001714  160011                    HRXCSR: 160011
1043  001716  160012                    RXDBUF: 160012
1044  001720  160013                    HRXDBUF:        160013
1045  001722  160012                    PARCSR: 160012
1046  001724  160013                    HPARCSR:        160013
1047  001726  160014                    TXCSR:  160014
1048  001730  160015                    HTXCSR: 160015
1049  001732  160016                    TXDBUF: 160016
1050  001734  160017                    HTXDBUF:        160017
1051                                            ;DEFAULT DU VECTORS
1052  001736  000770                    DURIV:  770     ;REC INTR VECTOR
1053  001740  000772                    DURIS:  772     ;REC INTR STATUS
1054  001742  000774                    DUTIV:  774     ;XMIT INTR VECTOR
1055  001744  000776                    DUTIS:  776     ;XMIT INTR STATUS
1056                                            ;ERROR MESSAGES
1057  001746  020040  051105  047522    EM4:    .ASCIZ  /  ERROR PC /
1058  001754  020122  041520  000040
1059  001762  020040  047503  050115    EM1:    .ASCIZ /  COMPARISON ERROR ON REGISTERS/
1060  001770  051101  051511  047117
1061  001776  042440  051122  051117
1062  002004  047440  020116  042522
```

```
                                                ERROR POINTER TABLE

  1063  002012  044507  052123  051105
  1064  002020  000123
  1065  002022  020040  042522  042503  EM2:    .ASCIZ  /  RECEIVER ERROR/
  1066  002030  053111  051105  042440
  1067  002036  051122  051117     000
  1068  002043     040  052040  040522  EM3:    .ASCIZ  /  TRANSMITTER ERROR/
  1069  002050  051516  044515  052124
  1070  002056  051105  042440  051122
  1071  002064  051117     000
  1072                                          ;DATA HEADERS FOR ERROR MESSAGES
  1073  002067     105  051122  041520  DH1:    .ASCIZ  /ERRPC  WANTED  ACTUAL/
  1074  002074  020040  040527  052116
  1075  002102  042105  020040  041501
  1076  002110  052524  046101     000
  1077          002116                  .EVEN
  1078                                          ;DATA TABLES FOR ERROR MESSAGES
  1079  002116  001416  001130  001132  DT1:    .WORD   $ERRPC,HLD0,HLD1,0
  1080  002124  000000
  1081
  1082  002126  001416  000000          DT4:    .WORD   $ERRPC,0
  1083
  1084  002132     000     000     000  DF1:    .BYTE   0,0,0,0
  1085  002135     000
  1086                                  .EVEN
  1087                                  .SBTTL  ACT11 HOOKS
  1088
  1089                                  ;;*****************************************************************
  1090                                  ;HOOKS REQUIRED BY ACT11
  1091          002136                          $SVPC=.                 ;SAVE PC
  1092          000046                          .=46
  1093  000046  012702                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
  1094          000052                          .=52
  1095  000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
  1096          002136                          .=$SVPC                 ;; RESTORE PC
  1097                                  .SBTTL  APT PARAMETER BLOCK
  1098
  1099                                  ;;*****************************************************************
  1100                                  ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
  1101                                  ;;*****************************************************************
  1102          002136                          .$X=.   ;;SAVE CURRENT LOCATION
  1103          000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
  1104  000024  000200                          200     ;;FOR APT START UP
  1105          000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
  1106  000044  002136                          $APTHDR ;;POINT TO APT HEADER BLOCK
  1107          002136                          .=.$X   ;;RESET LOCATION COUNTER
  1108                                  ;;*****************************************************************
  1109                                  ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
  1110                                  ;INTERFACE SPEC.
  1111
  1112  002136                          $APTHD:
  1113  002136  000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
  1114  002140  001526                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
  1115  002142  000010                  $TSTM:  .WORD   10      ;;RUN TIM OF LONGEST TEST
  1116  002144  000010                  $PASTM: .WORD   10      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
  1117  002146  000000                  $UNITM: .WORD           ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
  1118  002150  000052                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                        ;PROGRAM INITIALIZATION
                                        ;LOCK OUT INTERRUPTS
                                        ;SET UP PROCESSOR STACK
                                        ;SET UP POWER FAIL VECTOR
                                        ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
                                        ;TYPE TITLE MESSAGE

002152                          .START:
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
002152  012706  001400                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
002156  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
002160  022706  001440                  CMP     #$WR,R6 ;;DONE?
002164  001374                          BNE     .-6             ;;LOOP BACK IF NO
002166  012706  001100                  MOV     #STACK,SP               ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
002172  012737  016332  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
002200  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
002206  012737  014222  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
002214  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
002222  012737  016650  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
002230  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
002236  012737  015024  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
002244  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
002252  005067  177234                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
002256  005067  177232                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
002262  112767  000001  177125          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
002270  012767  002270  177110          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
002276  012767  002276  177104          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
002304  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
002310  012737  002344  000004          MOV     #  4$,@#ERRVEC  ;;SET UP ERROR VECTOR
002316  012767  177570  177114          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
002324  012767  177570  177110          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
002332  022777  177777  177100          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
002340  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
002342  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
002344  012716  002352          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
002350  000002                          RTI
002352  012767  000176  177060  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
002360  012767  000174  177054          MOV     #DISPREG,DISPLAY
002366  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

002372  005067  177136                  CLR     $PASS           ;;CLEAR PASS COUNT
002376  132767  000200  177143          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
002404  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
002406  012767  001550  177024          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
002414                          67$:
002414  012706  001100                  MOV     #STACK,SP               ;SET STACK
002420  106427  000340                  MTPS    #340                    ;LOCK INTERRUPTS
002424  012737  015024  000024          MOV     #.PFAIL,@#24            ;SET UP POWER FAIL VECTOR
002432  105067  176535                  CLRB    STFLG                   ;CLEAR START FLAG
```

INITIALIZE THE COMMON TAGS

```
1175  002436  005067  176450                  CLR     PASCNT                  ;CLEAR PASS COUNT
1176  002442  105067  176735                  CLRB    $ERFLG                  ;CLEAR ERROR FLAG
1177  002446  005067  176740                  CLR     $ERTTL                  ;CLEAR ERROR COUNT
1178  002452  005067  176740                  CLR     $ERRPC                  ;CLEAR LAST ERROR POINTER
1179  002456  012767  000001  176716          MOV     #1,$TSTNM               ;SET UP FOR TEST 1
1180  002464  012767  002152  176412          MOV     #.START,RETURN          ;SET UP FOR POWER FAIL BEFORE
1181                                                                          ;TESTING STARTS
1182  002472  013746  000006                  MOV     @#6,-(SP)
1183  002476  013746  000004                  MOV     @#4,-(SP)
1184  002502  012737  002516  000004          MOV     #1$,@#4
1185  002510  005777  176724                  TST     @SWR
1186  002514  000407                          BR      2$
1187  002516  012767  000176  176714  1$.     MOV     #SWREG,SWR
1188  002524  012767  000174  176710          MOV     #DISPREG,DISPLAY
1189  002532  022626                          CMP     (SP)+,(SP)+
1190  002534  012637  000004          2$:     MOV     (SP)+,@#4
1191  002540  012637  000006                  MOV     (SP)+,@#6
1192  002544  022767  000176  176666          CMP     #SWREG,SWR
1193  002552  001007                          BNE     3$
1194  002554  005737  000042                  TST     @#42                    ;CHECK FOR CHAIN
1195  002560  001402                          BEQ     33$
1196  002562  000167  000522                  JMP     .BEGIN
1197  002566  004767  010216          33$:    JSR     PC,CNTLU
1198  002572  105767  176374          3$:     TSTB    INIFLG                  ;HAS INITIALIZATION BEEN PERFORMED
1199  002576  001004                          BNE     ONCE
1200  002600  104401  015164                  TYPE    ,MTITLE                 ;TYPE TITLE MESSAGE
1201  002604  105167  176362                  COMB    INIFLG                  ;IF NOT SET FLAG AND DO
1202  002610  105767  176732          ONCE:   TSTB    $ENV            ;APT CONTROL?
1203  002614  001410                          BEQ     11$             ;BR IF NO
1204  002616  032767  000001  176726          BIT     #1,$USWR        ;EXTENAL JUMPER ON?
1205  002624  001002                          BNE     12$             ;NO
1206  002626  105067  176321                  CLRB    JMPBY           ;CLEAR FLAG
1207  002632  000167  000452          12$:    JMP     .BEGIN          ;GO DO IT
1208  002636  032777  000001  176574  11$:    BIT     #SW00,@SWR      ;RESELECT VECTOR & CONTROL REG?
1209  002644  001002                          BNE     1$
1210  002646  000167  000436                  JMP     .BEGIN
1211  002652  012700  000300          1$:     MOV     #300,R0 ;RESTORE VECTOR AREA TO TRAPCATCHER
1212  002656  012701  000302                  MOV     #302,R1 ;START AT LOCATION 300
1213  002662  012702  000004                  MOV     #4,R2
1214  002666  010110          2$:     MOV     R1,(R0)
1215  002670  005011                          CLR     (R1)
1216  002672  060200                          ADD     R2,R0
1217  002674  060201                          ADD     R2,R1
1218  002676  022701  001000                  CMP     #1000,R1        ;END AT LOCATION 776
1219  002702  002771                          BLT     2$
1220  002704  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1221  002706  015232                          MREGAD  ;MESSAGE
1222  002710  104410                          PARAM   ;CONVERT STRING
1223  002712  160000                          160000  ;LOW LIMIT
1224  002714  167776                          167776  ;HIGH LIMIT
1225  002716  017144                          DUBASE  ;STORE AT THIS LOCATION
1226  002720     001                  .BYTE   1       ;MASK
1227  002721     001                  .BYTE   1       ;HOW MANY TIMES + 2
1228  002722  016767  014216  176226          MOV     DUBASE,KEEPADD  ;SAVE
1229  002730  004767  014056                  JSR     PC,DUADDR
1230  002734  016767  176216  176212          MOV     KEEPADD,BASEADD ;RESTORE FOR ROTATION
```

```
1231   002742  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1232   002744  015217                          MVECTO  ;MESSAGE
1233   002746  104410                          PARAM   ;CONVERT STRING
1234   002750  000300                          300     ;LOW LIMIT
1235   002752  000776                          776     ;HIGH LIMIT
1236   002754  001736                          DURIV   ;STORE AT THIS LOCATION
1237   002756     001                  .BYTE   1       ;MASK
1238   002757     004                  .BYTE   4       ;HOW MANY TIMES + 2
1239   002760  016767  176752  176176          MOV     DURIV,KEEPIV    ;SAVE
1240   002766  016767  176744  176166          MOV     DURIV,BASEIV    ;SET UP FOR ROTATION
1241   002774  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1242   002776  015262                          MMULT   ;MESSAGE
1243   003000  104414                          SETFLG  ;SET FLAG BASED UPON INPUT STRING
1244   003002  001152                          MULTD   ;THIS FLAG
1245   003004  105767  176142                  TSTB    MULTD   ;ARE THERE MULTIPLE DEVICES
1246                                                           ;ON THE SYSTEM ?
1247   003010  100406                          BMI     BBB     ;YES,ASK NEXT QUESTION
1248   003012  005067  176150                  CLR     ACTREG
1249   003016  005067  176146                  CLR     ROTADD
1250   003022  000167  000140                  JMP     OUTMUL  ;JUMP AROUND NEXT QUESTION
1251   003026                          BBB:
1252   003026  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1253   003030  015311                          MLASTD  ;MESSAGE
1254   003032  104410                          PARAM   ;CONVERT STRING
1255   003034  160000                          160000  ;LOW LIMIT
1256   003036  167776                          167776  ;HIGH LIMIT
1257   003040  001160                          LASTADD ;STORE AT THIS LOCATION
1258   003042     001                  .BYTE   1       ;MASK
1259   003043     001                  .BYTE   1       ;HOW MANY TIMES + 2
1260                          ;THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
1261   003044  012767  000001  176116  1$:     MOV     #1,ROTADD       ;SET UP POINTER
1262   003052  005067  176110                  CLR     ACTREG  ;CLR ACTIVE REGISTER
1263   003056  056767  176106  176102  2$:     BIS     ROTADD,ACTREG   ;MAKE THIS DEVICE ACTIVE
1264   003064  000241                          CLC
1265   003066  006167  176076                  ROL     ROTADD  ;SET UP POINTER
1266   003072  103421                          BCS     3$      ;ARE YOU OUT OF RANGE ?
1267   003074  062767  000010  176052          ADD     #10,BASEADD     ;SET UP BASE ADDRESS
1268   003102  026767  176052  176044          CMP     LASTADD,BASEADD ;IS THIS THE LAST DEVICE ?
1269   003110  101362                          BHI     2$      ;NO DO IT AGAIN
1270   003112  056767  176052  176046          BIS     ROTADD,ACTREG   ;THIS ASSUMES THAT THERE ARE AT
1271                                                           ;LEAST TWO DEVICES WHEN YOU ANSWER YES TO
1272                                                           ;MULTIPLE DEVICE QUESTION
1273   003120  012767  000001  176042  4$:     MOV     #1,ROTADD       ;SET UP FOR LATER USE IN END OF PASS ROUTINE
1274   003126  016767  176024  176020          MOV     KEEPADD,BASEADD ;DITTO
1275   003134  000414                          BR      OUTMUL  ;CONTINUE QUESTIONS
1276   003136  016767  176014  176010  3$:     MOV     KEEPADD,BASEADD ;RESTORE
1277   003144  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1278   003146  015405                          MRANGE  ;MESSAGE
1279   003150  104410                          PARAM   ;CONVERT STRING
1280   003152  160000                          160000  ;LOW LIMIT
1281   003154  167776                          167776  ;HIGH LIMIT
1282   003156  001160                          LASTADD ;STORE AT THIS LOCATION
1283   003160     001                  .BYTE   1       ;MASK
1284   003161     001                  .BYTE   1       ;HOW MANY TIMES + 2
1285   003162  000167  177656                  JMP     1$      ;DO IT AGAIN
1286   003166  012767  000340  013612  OUTMUL: MOV     #340,DUPRT
```

INITIALIZE THE COMMON TAGS

```
1287  003174  004767  013536                        JSR     PC,DULEV
1288                                                ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1289                                                ;BUFFER TO THE CHARACTERS "1" AND "2".
1290                                                ;IF THE CHARACTER IS "1" CLEAR THE FLAG
1291                                                ;IF THE CHARACTER IS "2" SET THE FLAG
1292  003200                              AAA:
1293  003200  104406                             INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1294  003202  015623                             MSYNC   ;MESSAGE
1295  003204  122767  000061  012752     3$:     CMPB    #'1,INBUF       ;IS IT "1" ?
1296  003212  001003                             BNE     1$
1297  003214  105067  175726                    CLRB    SYNCNO  ;000
1298  003220  000412                             BR      4$
1299  003222  122767  000062  012734     1$:     CMPB    #'2,INBUF       ;IS IT "2" ?
1300  003230  001004                             BNE     2$
1301  003232  112767  177777  175706             MOVB    #-1,SYNCNO      ;377
1302  003240  000402                             BR      4$
1303  003242  104407                     2$:     INSTER  ;RETRY
1304  003244  000757                             BR      3$
1305  003246  000240                     4$:     NOP
1306  003250  104406                             INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1307  003252  015671                             MWIRE6  ;MESSAGE
1308  003254  104414                             SETFLG  ;SET FLAG BASED UPON INPUT STRING
1309  003256  001147                             SEXMIT  ;THIS FLAG
1310  003260  104406                             INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1311  003262  015742                             MWIRE5  ;MESSAGE
1312  003264  104414                             SETFLG  ;SET FLAG BASED UPON INPUT STRING
1313  003266  001150                             SEREC   ;THIS FLAG
1314  003270  104406                             INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1315  003272  016012                             MWIRE4  ;MESSAGE
1316  003274  104414                             SETFLG  ;SET FLAG BASED UPON INPUT STRING
1317  003276  001151                             OPTCLR  ;THIS FLAG
1318  003300  104406                             INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1319  003302  016071                             MEXTJ   ;MESSAGE
1320  003304  104414                             SETFLG  ;SET FLAG BASED UPON INPUT STRING
1321  003306  001153                             JMRBY   ;THIS FLAG
1322
1323                                              ;TEST START AND RESTART
1324
1325  003310  012706  001100          .BEGIN:    MOV     #STACK,SP               ;SET UP STACK
1326  003314  106427  000340                     MTPS    #340                    ;LOCK OUT INTERRUPTS
1327  003320  032777  000002  176112             BIT     #SW01,@SWR              ;IF SW01=1, GET STARTING PC
1328  003326  001406                             BEQ     3$
1329  003330  104406                             INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1330  003332  015555                             MTSTPC  ;MESSAGE
1331  003334  104410                             PARAM   ;CONVERT STRING
1332  003336  003362                             TST1    ;LOW LIMIT
1333                                                    ;HIGH LIMIT
1334                                                    ;STORE AT THIS LOCATION
1335  003340     001                  .BYTE   1       ;MASK
1336  003341     001                  .BYTE   1       ;HOW MANY TIMES + 2
1337  003342  000403                             BR      4$
1338  003344  012767  003362  175532     3$:     MOV     #TST1,RETURN            ;START AT TEST 1
1339  003352  104401  015551             4$:     TYPE    ,MR                     ;TYPE R
1340  003356  000177  175522                     JMP     @RETURN                 ;START TESTING
1341
1342                                              ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
```

```
                                        INITIALIZE THE COMMON TAGS

1343                                                    ;;OF THE TRANSMITTER SECTION.
1344                                                    ;;IT ALSO CHECKS DNA TIMING
1345                                                    ;;MODE:ISYMOD
1346                                                    ;;LENGTH:SEVEN PLUS PARITY
1347                                                    ;;PARITY:EVEPAR
1348                                                    ;;CHARACTER:12S
1349                                                    ;;
1350                                    ;;**************************************************************************
1351  003362  000004                    TST1:   SCOPE
1352  003364  052777  000400  176334            BIS     #MRESET,@TXCSR  ;MASTER RESET
1353  003372  012777  000000  176322            MOV     #ISYMOD,@PARCSR ;SET THE MODE
1354  003400  052777  000400  176320            BIS     #MRESET,@TXCSR  ;MASTER RESET
1355
1356                                    ;SET MAINTENANCE MODE & SEND
1357                                            ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1358  003406  012777  004020  176312            MOV     #MINT!SEND,@TXCSR
1359
1360                                    ;SET MODE, # OF BITS,PARITY SENSE,& LOAD SYNC REG
1361  003414  012777  005426  176300            MOV     #ISYMOD!SEVEN!EVEPAR!26,@PARCSR
1362  003422  016703  176300                   MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
1363  003426  112777  000125  176276            MOVB    #12S,@TXDBUF    ;LOAD DATA CHAR
1364  003434  012767  001252  176036            MOV     #1252,$TMP1     ;TO BE SHIFTED CHAR
1365  003442  012767  000012  175452            MOV     #10.,SHIFT      ;# OF SHIFTS
1366                                    ;POKE CLK TO GET INTO SYNCRONIZATION
1367  003450  052777  020000  176250            BIS     #CLK,@TXCSR     ;POKE CLK UP
1368  003456  042777  020000  176242            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1369  003464  005000                    1$:     CLR     R0
1370  003466  006067  176006                   ROR     $TMP1   ;FORCE CARRY
1371  003472  103002                           BCC     2$      ;BR IF CARRY CLR
1372  003474  052700  002000                   BIS     #BITW,R0        ;EQUIV OF BITW
1373  003500                            2$:
1374  003500  052777  020000  176220            BIS     #CLK,@TXCSR     ;POKE CLK UP
1375  003506  042777  020000  176212            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1376  003514  017701  176206                   MOV     @TXCSR,R1       ;ACTUAL
1377  003520  042701  075777                   BIC     #075777,R1      ;SAVE BITW & DNA
1378  003524  020001                           CMP     R0,R1   ;COMPARE EXP VS ACT
1379  003526  001401                           BEQ     .+4
1380  003530  104003                           ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1381                                                            ;BIT....ALSO CHECK DNA
1382  003532  005367  175364                   DEC     SHIFT   ;# OF SHIFTS
1383  003536  001352                           BNE     1$      ;DO IT AGAIN ?
1384                                    ;NOW POKE CLK TO SEE DNA
1385  003540  052777  020000  176160            BIS     #CLK,@TXCSR     ;POKE CLK
1386  003546  012700  000000                   MOV     #0,R0   ;EXPECTED
1387  003552  017701  176150                   MOV     @TXCSR,R1       ;ACTUAL
1388  003556  042701  077777                   BIC     #77777,R1       ;SAVE DNA ONLY
1389  003562  020001                           CMP     R0,R1           ;COMPARE EXP VS ACT
1390  003564  001401                           BEQ     .+4
1391  003566  104003                           ERROR   3       ;DNA SHOULD BE SET
1392                                                            ;IF DNA DID NOT SET
1393                                                            ;CHECK WORD LENGTH SELECT LOGIC
1394
1395                                            ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1396                                            ;;OF THE TRANSMITTER SECTION.
1397                                            ;;IT ALSO CHECKS DNA TIMING
1398                                            ;;MODE:ISYMOD
```

```
                                        INITIALIZE THE COMMON TAGS

1399                                                    ;;LENGTH:SEVEN PLUS PARITY
1400                                                    ;;PARITY:ODDPAR
1401                                                    ;;CHARACTER:125
1402                                                    ;;
1403                                    ;;**************************************************************************
1404  003570  000004                    TST2:   SCOPE
1405  003572  052777  000400  176126            BIS     #MRESET,@TXCSR  ;MASTER RESET
1406  003600  012777  000000  176114            MOV     #ISYMOD,@PARCSR ;SET THE MODE
1407  003606  052777  000400  176112            BIS     #MRESET,@TXCSR  ;MASTER RESET
1408
1409                                    ;SET MAINTENANCE MODE & SEND
1410                                            ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1411  003614  012777  004020  176104            MOV     #MINT!SEND,@TXCSR
1412
1413                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1414  003622  012777  005026  176072            MOV     #ISYMOD!SEVEN!ODDPAR!26,@PARCSR
1415  003630  016703  176072                   MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
1416  003634  112777  000125  176070            MOVB    #125,@TXDBUF    ;LOAD DATA CHAR
1417  003642  012767  001652  175630            MOV     #1652,$TMP1     ;TO BE SHIFTED CHAR
1418  003650  012767  000012  175244            MOV     #10.,SHIFT      ;# OF SHIFTS
1419                                    ;POKE CLK TO GET INTO SYNCRONIZATION
1420  003656  052777  020000  176042            BIS     #CLK,@TXCSR     ;POKE CLK UP
1421  003664  042777  020000  176034            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1422  003672  005000                    1$:     CLR     R0
1423  003674  006067  175600                   ROR     $TMP1   ;FORCE CARRY
1424  003700  103002                           BCC     2$      ;BR IF CARRY CLR
1425  003702  052700  002000                   BIS     #BITW,R0        ;EQUIV OF BITW
1426  003706                            2$:
1427  003706  052777  020000  176012            BIS     #CLK,@TXCSR     ;POKE CLK UP
1428  003714  042777  020000  176004            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1429  003722  017701  176000                   MOV     @TXCSR,R1       ;ACTUAL
1430  003726  042701  075777                   BIC     #075777,R1      ;SAVE BITW & DNA
1431  003732  020001                           CMP     R0,R1   ;COMPARE EXP VS ACT
1432  003734  001401                           BEQ     .+4
1433  003736  104003                           ERROR   3               ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1434                                                                    ;BIT...ALSO CHECK DNA
1435  003740  005367  175156                   DEC     SHIFT   ;# OF SHIFTS
1436  003744  001352                           BNE     1$      ;DO IT AGAIN ?
1437                                    ;NOW POKE CLK TO SEE DNA
1438  003746  052777  020000  175752            BIS     #CLK,@TXCSR     ;POKE CLK
1439  003754  012700  000000                   MOV     #0,R0   ;EXPECTED
1440  003760  017701  175742                   MOV     @TXCSR,R1       ;ACTUAL
1441  003764  042701  077777                   BIC     #77777,R1       ;SAVE DNA ONLY
1442  003770  020001                           CMP     R0,R1           ;COMPARE EXP VS ACT
1443  003772  001401                           BEQ     .+4
1444  003774  104003                           ERROR   3               ;DNA SHOULD BE SET
1445                                                                    ;IF DNA DID NOT SET
1446                                                                    ;CHECK WORD LENGTH SELECT LOGIC
1447
1448                                                    ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1449                                                    ;;OF THE TRANSMITTER SECTION.
1450                                                    ;;IT ALSO CHECKS DNA TIMING
1451                                                    ;;MODE:SYNINT
1452                                                    ;;LENGTH:EIGHT PLUS PARITY
1453                                                    ;;PARITY:EVEPAR
1454                                                    ;;CHARACTER:125
```

```
                                        INITIALIZE THE COMMON TAGS

1455                                                    ;;
1456                                    ;;**************************************************************************
1457  003776  000004                    TST3:   SCOPE
1458  004000  052777  000400  175720              BIS     #MRESET,@TXCSR  ;MASTER RESET
1459  004006  012777  030000  175706              MOV     #SYNINT,@PARCSR ;SET THE MODE
1460  004014  052777  000400  175704              BIS     #MRESET,@TXCSR  ;MASTER RESET
1461
1462                                    ;SET MAINTENANCE MODE & SEND
1463                                            ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1464  004022  012777  004020  175676              MOV     #MINT!SEND,@TXCSR
1465
1466                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1467  004030  012777  037426  175664              MOV     #SYNINT!EIGHT!EVEPAR!26,@PARCSR
1468  004036  016703  175664                    MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
1469  004042  112777  000125  175662              MOVB    #125,@TXDBUF    ;LOAD DATA CHAR
1470  004050  012767  000125  175422              MOV     #125,$TMP1      ;TO BE SHIFTED CHAR
1471  004056  012767  000011  175036              MOV     #9.,SHIFT       ;# OF SHIFTS
1472                                    ;POKE CLK TO GET INTO SYNCRONIZATION
1473  004064  052777  020000  175634              BIS     #CLK,@TXCSR     ;POKE CLK UP
1474  004072  042777  020000  175626              BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1475  004100  005000                    1$:     CLR     R0
1476  004102  006067  175372                    ROR     $TMP1   ;FORCE CARRY
1477  004106  103002                            BCC     2$      ;BR IF CARRY CLR
1478  004110  052700  002000                    BIS     #BITW,R0        ;EQUIV OF BITW
1479  004114                            2$:
1480  004114  052777  020000  175604              BIS     #CLK,@TXCSR     ;POKE CLK UP
1481  004122  042777  020000  175576              BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1482  004130  017701  175572                    MOV     @TXCSR,R1       ;ACTUAL
1483  004134  042701  075777                    BIC     #075777,R1      ;SAVE BITW & DNA
1484  004140  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
1485  004142  001401                            BEQ     .+4
1486  004144  104003                            ERROR   3               ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1487                                                                    ;BIT...ALSO CHECK DNA
1488  004146  005367  174750                    DEC     SHIFT   ;# OF SHIFTS
1489  004152  001352                            BNE     1$      ;DO IT AGAIN ?
1490                                    ;NOW POKE CLK TO SEE DNA
1491  004154  052777  020000  175544              BIS     #CLK,@TXCSR     ;POKE CLK
1492  004162  012700  100000                    MOV     #100000,R0      ;EXPECTED
1493  004166  017701  175534                    MOV     @TXCSR,R1       ;ACTUAL
1494  004172  042701  077777                    BIC     #77777,R1       ;SAVE DNA ONLY
1495  004176  020001                            CMP     R0,R1           ;COMPARE EXP VS ACT
1496  004200  001401                            BEQ     .+4
1497  004202  104003                            ERROR   3               ;DNA SHOULD BE SET
1498                                                                    ;IF DNA DID NOT SET
1499                                                                    ;CHECK WORD LENGTH SELECT LOGIC
1500
1501                                                    ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1502                                                    ;;OF THE TRANSMITTER SECTION.
1503                                                    ;;IT ALSO CHECKS DNA TIMING
1504                                                    ;;MODE:SYNINT
1505                                                    ;;LENGTH:EIGHT PLUS PARITY
1506                                                    ;;PARITY:ODDPAR
1507                                                    ;;CHARACTER:125
1508                                                    ;;
1509                                    ;;**************************************************************************
1510  004204  000004                    TST4:   SCOPE
```

```
DZDUU-A MACY11 27(1006)  03-FEB-77  08:00  PAGE 35
DZDUUA.M11    13-OCT-76 08:28          INITIALIZE THE COMMON TAGS

  1511  004206  052777  000400  175512          BIS     #MRESET,@TXCSR  ;MASTER RESET
  1512  004214  012777  030000  175500          MOV     #SYNINT,@PARCSR ;SET THE MODE
  1513  004222  052777  000400  175476          BIS     #MRESET,@TXCSR  ;MASTER RESET
  1514
  1515                                  ;SET MAINTENANCE MODE & SEND
  1516                                          ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
  1517  004230  012777  004020  175470          MOV     #MINT!SEND,@TXCSR
  1518
  1519                                  ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
  1520  004236  012777  037026  175456          MOV     #SYNINT!EIGHT!ODDPAR!26,@PARCSR
  1521  004244  016703  175456                  MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
  1522  004250  112777  000125  175454          MOVB    #125,@TXDBUF    ;LOAD DATA CHAR
  1523  004256  012767  000525  175214          MOV     #525,$TMP1      ;TO BE SHIFTED CHAR
  1524  004264  012767  000011  174630          MOV     #9.,SHIFT       ;# OF SHIFTS
  1525                                  ;POKE CLK TO GET INTO SYNCRONIZATION
  1526  004272  052777  020000  175426          BIS     #CLK,@TXCSR     ;POKE CLK UP
  1527  004300  042777  020000  175420          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
  1528  004306  005000                  1$:     CLR     R0
  1529  004310  006067  175164                  ROR     $TMP1   ;FORCE CARRY
  1530  004314  103002                          BCC     2$      ;BR IF CARRY CLR
  1531  004316  052700  002000                  BIS     #BITW,R0        ;EQUIV OF BITW
  1532  004322                          2$:
  1533  004322  052777  020000  175376          BIS     #CLK,@TXCSR     ;POKE CLK UP
  1534  004330  042777  020000  175370          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
  1535  004336  017701  175364                  MOV     @TXCSR,R1       ;ACTUAL
  1536  004342  042701  075777                  BIC     #075777,R1      ;SAVE BITW & DNA
  1537  004346  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
  1538  004350  001401                          BEQ     .+4
  1539  004352  104003                          ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
  1540                                                          ;BIT....ALSO CHECK DNA
  1541  004354  005367  174542                  DEC     SHIFT   ;# OF SHIFTS
  1542  004360  001352                          BNE     1$      ;DO IT AGAIN ?
  1543                                  ;NOW POKE CLK TO SEE DNA
  1544  004362  052777  020000  175336          BIS     #CLK,@TXCSR     ;POKE CLK
  1545  004370  012700  100000                  MOV     #100000,R0      ;EXPECTED
  1546  004374  017701  175326                  MOV     @TXCSR,R1       ;ACTUAL
  1547  004400  042701  077777                  BIC     #77777,R1       ;SAVE DNA ONLY
  1548  004404  020001                          CMP     R0,R1           ;COMPARE EXP VS ACT
  1549  004406  001401                          BEQ     .+4
  1550  004410  104003                          ERROR   3       ;DNA SHOULD BE SET
  1551                                                          ;IF DNA DID NOT SET
  1552                                                          ;CHECK WORD LENGTH SELECT LOGIC
  1553
  1554                                          ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
  1555                                          ;;OF THE TRANSMITTER SECTION.
  1556                                          ;;IT ALSO CHECKS DNA TIMING
  1557                                          ;;MODE:ISYMOD
  1558                                          ;;LENGTH:EIGHT PLUS PARITY
  1559                                          ;;PARITY:EVEPAR
  1560                                          ;;CHARACTER:125
  1561                                          ;;
  1562                                  ;;**************************************************************
  1563  004412  000004                  TST5:   SCOPE
  1564  004414  052777  000400  175304          BIS     #MRESET,@TXCSR  ;MASTER RESET
  1565  004422  012777  000000  175272          MOV     #ISYMOD,@PARCSR ;SET THE MODE
  1566  004430  052777  000400  175270          BIS     #MRESET,@TXCSR  ;MASTER RESET
```

INITIALIZE THE COMMON TAGS

```
1567
1568                                        ;SET MAINTENANCE MODE & SEND
1569                                                ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1570  004436  012777  004020  175262                MOV     #MINT!SEND,@TXCSR
1571
1572                                        ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1573  004444  012777  007426  175250                MOV     #ISYMOD!EIGHT!EVEPAR!26,@PARCSR
1574  004452  016703  175250                      MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
1575  004456  112777  000125  175246              MOVB    #125,@TXDBUF    ;LOAD DATA CHAR
1576  004464  012767  002252  175006              MOV     #2252,$TMP1     ;TO BE SHIFTED CHAR
1577  004472  012767  000013  174422              MOV     #11.,SHIFT      ;# OF SHIFTS
1578                                        ;POKE CLK TO GET INTO SYNCRONIZATION
1579  004500  052777  020000  175220              BIS     #CLK,@TXCSR     ;POKE CLK UP
1580  004506  042777  020000  175212              BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1581  004514  005000                        1$:   CLR     R0
1582  004516  006067  174756                      ROR     $TMP1   ;FORCE CARRY
1583  004522  103002                              BCC     2$      ;BR IF CARRY CLR
1584  004524  052700  002000                      BIS     #BITW,R0        ;EQUIV OF BITW
1585  004530                                2$:
1586  004530  052777  020000  175170              BIS     #CLK,@TXCSR     ;POKE CLK UP
1587  004536  042777  020000  175162              BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1588  004544  017701  175156                      MOV     @TXCSR,R1       ;ACTUAL
1589  004550  042701  075777                      BIC     #075777,R1      ;SAVE BITW & DNA
1590  004554  020001                              CMP     R0,R1   ;COMPARE EXP VS ACT
1591  004556  001401                              BEQ     .+4
1592  004560  104003                              ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1593                                                        ;BIT....ALSO CHECK DNA
1594  004562  005367  174334                      DEC     SHIFT   ;# OF SHIFTS
1595  004566  001352                              BNE     1$      ;DO IT AGAIN ?
1596                                        ;NOW POKE CLK TO SEE DNA
1597  004570  052777  020000  175130              BIS     #CLK,@TXCSR     ;POKE CLK
1598  004576  012700  000000                      MOV     #0,R0   ;EXPECTED
1599  004602  017701  175120                      MOV     @TXCSR,R1       ;ACTUAL
1600  004606  042701  077777                      BIC     #77777,R1       ;SAVE DNA ONLY
1601  004612  020001                              CMP     R0,R1           ;COMPARE EXP VS ACT
1602  004614  001401                              BEQ     .+4
1603  004616  104003                              ERROR   3       ;DNA SHOULD BE SET
1604                                                        ;IF DNA DID NOT SET
1605                                                        ;CHECK WORD LENGTH SELECT LOGIC
1606
1607                                                ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1608                                                ;;OF THE TRANSMITTER SECTION.
1609                                                ;;IT ALSO CHECKS DNA TIMING
1610                                                ;;MODE:ISYMOD
1611                                                ;;LENGTH:EIGHT PLUS PARITY
1612                                                ;;PARITY:ODDPAR
1613                                                ;;CHARACTER:125
1614                                                ;;
1615                                        ;;**************************************************************
1616  004620  000004                        TST6:   SCOPE
1617  004622  052777  000400  175076              BIS     #MRESET,@TXCSR  ;MASTER RESET
1618  004630  012777  000000  175064              MOV     #ISYMOD,@PARCSR ;SET THE MODE
1619  004636  052777  000400  175062              BIS     #MRESET,@TXCSR  ;MASTER RESET
1620
1621                                        ;SET MAINTENANCE MODE & SEND
1622                                                ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
```

```
                                        INITIALIZE THE COMMON TAGS

1623  004644  012777  004020  175054            MOV     #MINT!SEND,@TXCSR
1624
1625                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1626  004652  012777  007026  175042            MOV     #ISYMOD!EIGHT!ODDPAR!26,@PARCSR
1627  004660  016703  175042                    MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
1628  004664  112777  000125  175040            MOVB    #125,@TXDBUF    ;LOAD DATA CHAR
1629  004672  012767  003252  174600            MOV     #3252,$TMP1     ;TO BE SHIFTED CHAR
1630  004700  012767  000013  174214            MOV     #11.,SHIFT      ;# OF SHIFTS
1631                                    ;POKE CLK TO GET INTO SYNCRONIZATION
1632  004706  052777  020000  175012            BIS     #CLK,@TXCSR     ;POKE CLK UP
1633  004714  042777  020000  175004            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1634  004722  005000                    1$:     CLR     R0
1635  004724  006067  174550                    ROR     $TMP1   ;FORCE CARRY
1636  004730  103002                            BCC     2$      ;BR IF CARRY CLR
1637  004732  052700  002000                    BIS     #BITW,R0        ;EQUIV OF BITW
1638  004736                            2$:
1639  004736  052777  020000  174762            BIS     #CLK,@TXCSR     ;POKE CLK UP
1640  004744  042777  020000  174754            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1641  004752  017701  174750                    MOV     @TXCSR,R1       ;ACTUAL
1642  004756  042701  075777                    BIC     #075777,R1      ;SAVE BITW & DNA
1643  004762  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
1644  004764  001401                            BEQ     .+4
1645  004766  104003                            ERROR   3               ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1646                                                                    ;BIT...ALSO CHECK DNA
1647  004770  005367  174126                    DEC     SHIFT   ;# OF SHIFTS
1648  004774  001352                            BNE     1$      ;DO IT AGAIN ?
1649                                    ;NOW POKE CLK TO SEE DNA
1650  004776  052777  020000  174722            BIS     #CLK,@TXCSR     ;POKE CLK
1651  005004  012700  000000                    MOV     #0,R0   ;EXPECTED
1652  005010  017701  174712                    MOV     @TXCSR,R1       ;ACTUAL
1653  005014  042701  077777                    BIC     #77777,R1       ;SAVE DNA ONLY
1654  005020  020001                            CMP     R0,R1           ;COMPARE EXP VS ACT
1655  005022  001401                            BEQ     .+4
1656  005024  104003                            ERROR   3       ;DNA SHOULD BE SET
                                                                ;IF DNA DID NOT SET
                                                                ;CHECK WORD LENGTH SELECT LOGIC
1657
1658
1659
1660
1661                                                    ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1662                                                    ;;OF THE TRANSMITTER SECTION.
1663                                                    ;;IT ALSO CHECKS DNA TIMING
1664                                                    ;;MODE:SYNINT
1665                                                    ;;LENGTH:EIGHT PLUS PARITY
1666                                                    ;;PARITY:EVEPAR
1667                                                    ;;CHARACTER:252
1668                                                    ;;
1669                                    ;;**************************************************************
1670  005026  000004                    TST7:   SCOPE
1671  005030  052777  000400  174670            BIS     #MRESET,@TXCSR  ;MASTER RESET
1672  005036  012777  030000  174656            MOV     #SYNINT,@PARCSR ;SET THE MODE
1673  005044  052777  000400  174654            BIS     #MRESET,@TXCSR  ;MASTER RESET
1674
1675                                    ;SET MAINTENANCE MODE & SEND
1676                                            ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1677  005052  012777  004020  174646            MOV     #MINT!SEND,@TXCSR
1678
```

```
                                        INITIALIZE THE COMMON TAGS

1679
1680  005060  012777  037426  174634            ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
                        MOV     #SYNINT!EIGHT!EVEPAR!26,@PARCSR
1681  005066  016703  174634                    MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
1682  005072  112777  000252  174632            MOVB    #252,@TXDBUF    ;LOAD DATA CHAR
1683  005100  012767  000252  174372            MOV     #252,$TMP1      ;TO BE SHIFTED CHAR
1684  005106  012767  000011  174006            MOV     #9.,SHIFT       ;# OF SHIFTS
1685                                    ;POKE CLK TO GET INTO SYNCRONIZATION
1686  005114  052777  020000  174604            BIS     #CLK,@TXCSR     ;POKE CLK UP
1687  005122  042777  020000  174576            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1688  005130  005000                    1$:     CLR     R0
1689  005132  006067  174342                    ROR     $TMP1   ;FORCE CARRY
1690  005136  103002                            BCC     2$      ;BR IF CARRY CLR
1691  005140  052700  002000                    BIS     #BITW,R0        ;EQUIV OF BITW
1692  005144                            2$:
1693  005144  052777  020000  174554            BIS     #CLK,@TXCSR     ;POKE CLK UP
1694  005152  042777  020000  174546            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1695  005160  017701  174542                    MOV     @TXCSR,R1       ;ACTUAL
1696  005164  042701  075777                    BIC     #075777,R1      ;SAVE BITW & DNA
1697  005170  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
1698  005172  001401                            BEQ     .+4
1699  005174  104003                            ERROR   3               ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1700                                                                    ;BIT....ALSO CHECK DNA
1701  005176  005367  173720                    DEC     SHIFT   ;# OF SHIFTS
1702  005202  001352                            BNE     1$      ;DO IT AGAIN ?
1703                                    ;NOW POKE CLK TO SEE DNA
1704  005204  052777  020000  174514            BIS     #CLK,@TXCSR     ;POKE CLK
1705  005212  012700  100000                    MOV     #100000,R0      ;EXPECTED
1706  005216  017701  174504                    MOV     @TXCSR,R1       ;ACTUAL
1707  005222  042701  077777                    BIC     #77777,R1       ;SAVE DNA ONLY
1708  005226  020001                            CMP     R0,R1           ;COMPARE EXP VS ACT
1709  005230  001401                            BEQ     .+4
1710  005232  104003                            ERROR   3       ;DNA SHOULD BE SET
1711                                                            ;IF DNA DID NOT SET
1712                                                            ;CHECK WORD LENGTH SELECT LOGIC
1713
1714                                            ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1715                                            ;;OF THE TRANSMITTER SECTION.
1716                                            ;;IT ALSO CHECKS DNA TIMING
1717                                            ;;MODE:SYNINT
1718                                            ;;LENGTH:EIGHT PLUS PARITY
1719                                            ;;PARITY:ODDPAR
1720                                            ;;CHARACTER:252
1721                                            ;;
1722                                    ;;**************************************************************
1723  005234  000004                    TST10:  SCOPE
1724  005236  052777  000400  174462            BIS     #MRESET,@TXCSR  ;MASTER RESET
1725  005244  012777  030000  174450            MOV     #SYNINT,@PARCSR ;SET THE MODE
1726  005252  052777  000400  174446            BIS     #MRESET,@TXCSR  ;MASTER RESET
1727
1728                                    ;SET MAINTENANCE MODE & SEND
1729                                            ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1730  005260  012777  004020  174440            MOV     #MINT!SEND,@TXCSR
1731
1732                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1733  005266  012777  037026  174426            MOV     #SYNINT!EIGHT!ODDPAR!26,@PARCSR
1734  005274  016703  174426                    MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
```

```
                                        INITIALIZE THE COMMON TAGS

1735  005300  112777  000252  174424            MOVB    #252,@TXDBUF    ;LOAD DATA CHAR
1736  005306  012767  000652  174164            MOV     #652,$TMP1      ;TO BE SHIFTED CHAR
1737  005314  012767  000011  173600            MOV     #9.,SHIFT       ;# OF SHIFTS
1738                                    ;POKE CLK TO GET INTO SYNCRONIZATION
1739  005322  052777  020000  174376            BIS     #CLK,@TXCSR     ;POKE CLK UP
1740  005330  042777  020000  174370            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1741  005336  005000                    1$:     CLR     R0
1742  005340  006067  174134                    ROR     $TMP1   ;FORCE CARRY
1743  005344  103002                            BCC     2$      ;BR IF CARRY CLR
1744  005346  052700  002000                    BIS     #BITW,R0        ;EQUIV OF BITW
1745  005352                            2$:
1746  005352  052777  020000  174346            BIS     #CLK,@TXCSR     ;POKE CLK UP
1747  005360  042777  020000  174340            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1748  005366  017701  174334                    MOV     @TXCSR,R1       ;ACTUAL
1749  005372  042701  075777                    BIC     #075777,R1      ;SAVE BITW & DNA
1750  005376  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
1751  005400  001401                            BEQ     .+4
1752  005402  104003                            ERROR   3               ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1753                                                                    ;BIT...ALSO CHECK DNA
1754  005404  005367  173512                    DEC     SHIFT   ;# OF SHIFTS
1755  005410  001352                            BNE     1$      ;DO IT AGAIN ?
1756                                    ;NOW POKE CLK TO SEE DNA
1757  005412  052777  020000  174306            BIS     #CLK,@TXCSR     ;POKE CLK
1758  005420  012700  100000                    MOV     #100000,R0      ;EXPECTED
1759  005424  017701  174276                    MOV     @TXCSR,R1       ;ACTUAL
1760  005430  042701  077777                    BIC     #77777,R1       ;SAVE DNA ONLY
1761  005434  020001                            CMP     R0,R1           ;COMPARE EXP VS ACT
1762  005436  001401                            BEQ     .+4
1763  005440  104003                            ERROR   3       ;DNA SHOULD BE SET
1764                                                                    ;IF DNA DID NOT SET
1765                                                                    ;CHECK WORD LENGTH SELECT LOGIC
1766
1767                                                            ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1768                                                            ;;OF THE TRANSMITTER SECTION.
1769                                                            ;;IT ALSO CHECKS DNA TIMING
1770                                                            ;;MODE:SYNINT
1771                                                            ;;LENGTH:EIGHT PLUS PARITY
1772                                                            ;;PARITY:EVEPAR
1773                                                            ;;CHARACTER:0
1774                                                            ;;
1775                                    ;;**************************************************************
1776  005442  000004                    TST11:  SCOPE
1777  005444  052777  000400  174254            BIS     #MRESET,@TXCSR  ;MASTER RESET
1778  005452  012777  030000  174242            MOV     #SYNINT,@PARCSR ;SET THE MODE
1779  005460  052777  000400  174240            BIS     #MRESET,@TXCSR  ;MASTER RESET
1780
1781                                    ;SET MAINTENANCE MODE & SEND
1782                                            ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1783  005466  012777  004020  174232            MOV     #MINT!SEND,@TXCSR
1784
1785                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1786  005474  012777  037426  174220            MOV     #SYNINT!EIGHT!EVEPAR!26,@PARCSR
1787  005502  016703  174220                    MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
1788  005506  112777  000000  174216            MOVB    #0,@TXDBUF      ;LOAD DATA CHAR
1789  005514  012767  000000  173756            MOV     #0,$TMP1        ;TO BE SHIFTED CHAR
1790  005522  012767  000011  173372            MOV     #9.,SHIFT       ;# OF SHIFTS
```

```
                                        INITIALIZE THE COMMON TAGS

1791                                    ;POKE CLK TO GET INTO SYNCRONIZATION
1792  005530  052777  020000  174170            BIS     #CLK,@TXCSR     ;POKE CLK UP
1793  005536  042777  020000  174162            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1794  005544  005000                    1$:     CLR     R0
1795  005546  006067  173726                    ROR     $TMP1   ;FORCE CARRY
1796  005552  103002                            BCC     2$      ;BR IF CARRY CLR
1797  005554  052700  002000                    BIS     #BITW,R0        ;EQUIV OF BITW
1798  005560                            2$:
1799  005560  052777  020000  174140            BIS     #CLK,@TXCSR     ;POKE CLK UP
1800  005566  042777  020000  174132            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1801  005574  017701  174126                    MOV     @TXCSR,R1       ;ACTUAL
1802  005600  042701  075777                    BIC     #075777,R1      ;SAVE BITW & DNA
1803  005604  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
1804  005606  001401                            BEQ     .+4
1805  005610  104003                            ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1806                                                            ;BIT....ALSO CHECK DNA
1807  005612  005367  173304                    DEC     SHIFT   ;# OF SHIFTS
1808  005616  001352                            BNE     1$      ;DO IT AGAIN ?
1809                                    ;NOW POKE CLK TO SEE DNA
1810  005620  052777  020000  174100            BIS     #CLK,@TXCSR     ;POKE CLK
1811  005626  012700  100000                    MOV     #100000,R0      ;EXPECTED
1812  005632  017701  174070                    MOV     @TXCSR,R1       ;ACTUAL
1813  005636  042701  077777                    BIC     #77777,R1       ;SAVE DNA ONLY
1814  005642  020001                            CMP     R0,R1           ;COMPARE EXP VS ACT
1815  005644  001401                            BEQ     .+4
1816  005646  104003                            ERROR   3       ;DNA SHOULD BE SET
1817                                                            ;IF DNA DID NOT SET
1818                                                            ;CHECK WORD LENGTH SELECT LOGIC
1819
1820                                            ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1821                                            ;;OF THE TRANSMITTER SECTION.
1822                                            ;;IT ALSO CHECKS DNA TIMING
1823                                            ;;MODE:SYNINT
1824                                            ;;LENGTH:EIGHT PLUS PARITY
1825                                            ;;PARITY:ODDPAR
1826                                            ;;CHARACTER:0
1827                                            ;;
1828                                    ;;*****************************************************************
1829  005650  000004                    TST12:  SCOPE
1830  005652  052777  000400  174046            BIS     #MRESET,@TXCSR  ;MASTER RESET
1831  005660  012777  030000  174034            MOV     #SYNINT,@PARCSR ;SET THE MODE
1832  005666  052777  000400  174032            BIS     #MRESET,@TXCSR  ;MASTER RESET
1833
1834                                    ;SET MAINTENANCE MODE & SEND
1835                                            ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1836  005674  012777  004020  174024            MOV     #MINT!SEND,@TXCSR
1837
1838                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1839  005702  012777  037026  174012            MOV     #SYNINT!EIGHT!ODDPAR!26,@PARCSR
1840  005710  016703  174012                    MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
1841  005714  112777  000000  174010            MOVB    #0,@TXDBUF      ;LOAD DATA CHAR
1842  005722  012767  000400  173550            MOV     #400,$TMP1      ;TO BE SHIFTED CHAR
1843  005730  012767  000011  173164            MOV     #9.,SHIFT       ;# OF SHIFTS
1844                                    ;POKE CLK TO GET INTO SYNCRONIZATION
1845  005736  052777  020000  173762            BIS     #CLK,@TXCSR     ;POKE CLK UP
1846  005744  042777  020000  173754            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
```

```
DZDUU-A MACY11 27(1006)  03-FEB-77  08:00  PAGE 41
DZDUUA.M11    13-OCT-76 08:28          INITIALIZE THE COMMON TAGS

 1847  005752  005000                  1$:     CLR     R0
 1848  005754  006067  173520                  ROR     $TMP1   ;FORCE CARRY
 1849  005760  103002                          BCC     2$      ;BR IF CARRY CLR
 1850  005762  052700  002000                  BIS     #BITW,R0        ;EQUIV OF BITW
 1851  005766                          2$:
 1852  005766  052777  020000  173732          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1853  005774  042777  020000  173724          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1854  006002  017701  173720                  MOV     @TXCSR,R1       ;ACTUAL
 1855  006006  042701  075777                  BIC     #075777,R1      ;SAVE BITW & DNA
 1856  006012  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
 1857  006014  001401                          BEQ     .+4
 1858  006016  104003                          ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
 1859                                                          ;BIT....ALSO CHECK DNA
 1860  006020  006367  173076                  DEC     SHIFT   ;# OF SHIFTS
 1861  006024  001352                          BNE     1$      ;DO IT AGAIN ?
 1862                                  ;NOW POKE CLK TO SEE DNA
 1863  006026  052777  020000  173672          BIS     #CLK,@TXCSR     ;POKE CLK
 1864  006034  012700  100000                  MOV     #100000,R0      ;EXPECTED
 1865  006040  017701  173662                  MOV     @TXCSR,R1       ;ACTUAL
 1866  006044  042701  077777                  BIC     #77777,R1       ;SAVE DNA ONLY
 1867  006050  020001                          CMP     R0,R1           ;COMPARE EXP VS ACT
 1868  006052  001401                          BEQ     .+4
 1869  006054  104003                          ERROR   3       ;DNA SHOULD BE SET
 1870                                                          ;IF DNA DID NOT SET
 1871                                                          ;CHECK WORD LENGTH SELECT LOGIC
 1872
 1873                                          ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
 1874                                          ;;OF THE TRANSMITTER SECTION.
 1875                                          ;;IT ALSO CHECKS DNA TIMING
 1876                                          ;;MODE:SYNINT
 1877                                          ;;LENGTH:EIGHT PLUS PARITY
 1878                                          ;;PARITY:EVEPAR
 1879                                          ;;CHARACTER:377
 1880                                          ;;
 1881                                  ;;**************************************************************
 1882  006056  000004                  TST13:  SCOPE
 1883  006060  052777  000400  173640          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1884  006066  012777  030000  173626          MOV     #SYNINT,@PARCSR ;SET THE MODE
 1885  006074  052777  000400  173624          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1886
 1887                                  ;SET MAINTENANCE MODE & SEND
 1888                                          ;NOTE:BIT WINDOWS CLK ARE CLEARED (MTDATA=0)
 1889  006102  012777  004020  173616          MOV     #MINT!SEND,@TXCSR
 1890
 1891                                  ;SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
 1892  006110  012777  037426  173604          MOV     #SYNINT!EIGHT!EVEPAR!26,@PARCSR
 1893  006116  016703  173604                  MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
 1894  006122  112777  000377  173602          MOVB    #377,@TXDBUF    ;LOAD DATA CHAR
 1895  006130  012767  000377  173342          MOV     #377,$TMP1      ;TO BE SHIFTED CHAR
 1896  006136  012767  000011  172756          MOV     #9.,SHIFT       ;# OF SHIFTS
 1897                                  ;POKE CLK TO GET INTO SYNCRONIZATION
 1898  006144  052777  020000  173554          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1899  006152  042777  020000  173546          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1900  006160  005000                  1$:     CLR     R0
 1901  006162  006067  173312                  ROR     $TMP1   ;FORCE CARRY
 1902  006166  103002                          BCC     2$      ;BR IF CARRY CLR
```

INITIALIZE THE COMMON TAGS

```
006170  052700  002000                  BIS     #BITW,R0        ;EQUIV OF BITW
006174                          2$:
006174  052777  020000  173524          BIS     #CLK,@TXCSR     ;POKE CLK UP
006202  042777  020000  173516          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
006210  017701  173512                  MOV     @TXCSR,R1       ;ACTUAL
006214  042701  075777                  BIC     #075777,R1      ;SAVE BITW & DNA
006220  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
006222  001401                          BEQ     .+4
006224  104003                          ERROR   3               ;BIT WINDOW DID NOT MATCH ACTUAL DATA
                                                                ;BIT,...ALSO CHECK DNA
006226  005367  172670                  DEC     SHIFT           ;# OF SHIFTS
006232  001352                          BNE     1$              ;DO IT AGAIN ?
                                ;NOW POKE CLK TO SEE DNA
006234  052777  020000  173464          BIS     #CLK,@TXCSR     ;POKE CLK
006242  012700  100000                  MOV     #100000,R0      ;EXPECTED
006246  017701  173454                  MOV     @TXCSR,R1       ;ACTUAL
006252  042701  077777                  BIC     #77777,R1       ;SAVE DNA ONLY
006256  020001                          CMP     R0,R1           ;COMPARE EXP VS ACT
006260  001401                          BEQ     .+4
006262  104003                          ERROR   3               ;DNA SHOULD BE SET
                                                                ;IF DNA DID NOT SET
                                                                ;CHECK WORD LENGTH SELECT LOGIC

                                ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
                                ;;OF THE TRANSMITTER SECTION.
                                ;;IT ALSO CHECKS DNA TIMING
                                ;;MODE:SYNINT
                                ;;LENGTH:EIGHT PLUS PARITY
                                ;;PARITY:ODDPAR
                                ;;CHARACTER:377
                                ;;
                        ;;**************************************************************
006264  000004          TST14:  SCOPE
006266  052777  000400  173432          BIS     #MRESET,@TXCSR  ;MASTER RESET
006274  012777  030000  173420          MOV     #SYNINT,@PARCSR ;SET THE MODE
006302  052777  000400  173416          BIS     #MRESET,@TXCSR  ;MASTER RESET

                                ;SET MAINTENANCE MODE & SEND
                                        ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
006310  012777  004020  173410          MOV     #MINT!SEND,@TXCSR

                                ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
006316  012777  037026  173376          MOV     #SYNINT!EIGHT!ODDPAR!26,@PARCSR
006324  016703  173376                  MOV     TXCSR,R3        ;SET UP FOR ERROR MSG
006330  112777  000377  173374          MOVB    #377,@TXDBUF    ;LOAD DATA CHAR
006336  012767  000777  173134          MOV     #777,$TMP1      ;TO BE SHIFTED CHAR
006344  012767  000011  172550          MOV     #9.,SHIFT       ;# OF SHIFTS
                                ;POKE CLK TO GET INTO SYNCRONIZATION
006352  052777  020000  173346          BIS     #CLK,@TXCSR     ;POKE CLK UP
006360  042777  020000  173340          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
006366  005000                  1$:     CLR     R0
006370  006067  173104                  ROR     $TMP1   ;FORCE CARRY
006374  103002                          BCC     2$      ;BR IF CARRY CLR
006376  052700  002000                  BIS     #BITW,R0        ;EQUIV OF BITW
006402                          2$:
006402  052777  020000  173316          BIS     #CLK,@TXCSR     ;POKE CLK UP
```

```
                                        INITIALIZE THE COMMON TAGS

1959  006410  042777  020000  173310          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1960  006416  017701  173304                  MOV     @TXCSR,R1       ;ACTUAL
1961  006422  042701  075777                  BIC     #075777,R1      ;SAVE BITW & DNA
1962  006426  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
1963  006430  001401                          BEQ     .+4
1964  006432  104003                          ERROR   3       ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1965                                                          ;BIT,...ALSO CHECK DNA
1966  006434  005367  172462                  DEC     SHIFT   ;# OF SHIFTS
1967  006440  001352                          BNE     1$      ;DO IT AGAIN ?
1968                                  ;NOW POKE CLK TO SEE DNA
1969  006442  052777  020000  173256          BIS     #CLK,@TXCSR     ;POKE CLK
1970  006450  012700  100000                  MOV     #100000,R0      ;EXPECTED
1971  006454  017701  173246                  MOV     @TXCSR,R1       ;ACTUAL
1972  006460  042701  077777                  BIC     #77777,R1       ;SAVE DNA ONLY
1973  006464  020001                          CMP     R0,R1           ;COMPARE EXP VS ACT
1974  006466  001401                          BEQ     .+4
1975  006470  104003                          ERROR   3       ;DNA SHOULD BE SET
1976                                                          ;IF DNA DID NOT SET
1977                                                          ;CHECK WORD LENGTH SELECT LOGIC
1978
1979
1980
1981                                    ;;THIS TEST VERIFYS THAT BY SENDING ONLY ONE SYNC
1982                                    ;;CHARACTER (TWO SELECTED BY STRAPPING ) RECACT =0
1983                                    ;;THEN SEND ONE ORDINARY CHARACTER (TO BREAK UP THE SEQUENCE)
1984                                    ;;RECACT =0.............IT WILL TAKE TWO MORE SYNC CHARS
1985                                    ;;BEFORE RECACT =1
1986                                    ;;NOTE: THIS TEST WILL ONLY WORK WHEN TWO SYNC CHARS
1987                                    ;;HAS BEEN BEEN SELECTED ...OTHERWISE JUMP AROUND THIS TEST
1988                                    ;;MODE:SYNC INTERNAL (SYNINT)
1989                                    ;;PARITY: NOPAR
1990                                    ;;LENGTH: EIGHT
1991                                    ;;
1992                                    ;;THIS TEST CHECKS ONLY THE RECEIVER SECTION
1993                                    ;;
1994                                    ;;*****************************************************************
1995  006472  000004                  TST15:  SCOPE
1996  006474  105767  172446                  TSTB    SYNCNO  ;TEST FOR  # OF SYNC CHARS REQUIRED
1997  006500  100127                          BPL     2$      ;IF NOT TWO  GET OUT OF TEST
1998  006502  052777  000400  173216          BIS     #MRESET,@TXCSR  ;MASTER RESET
1999  006510  012777  030000  173204          MOV     #SYNINT,@PARCSR ;SET THE MODE
2000  006516  052777  000400  173202          BIS     #MRESET,@TXCSR  ;MASTER RESET
2001
2002                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2003  006524  012777  064001  173174          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2004
2005                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2006  006532  012777  036026  173162          MOV     #SYNINT!EIGHT!NOPAR!26,@PARCSR
2007  006540  052777  000020  173144          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2008                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2009  006546  042777  020000  173152          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2010  006554  052777  020000  173144          BIS     #CLK,@TXCSR     ;POKE CLK UP
2011                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2012  006562  042777  020000  173136          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2013  006570  052777  020000  173130          BIS     #CLK,@TXCSR     ;POKE CLK UP
2014  006576  012767  000010  172316          MOV     #8.,SHIFT       ;# OF SHIFTS
```

```
                                        INITIALIZE THE COMMON TAGS

2015  006604  012767  000026  172666            MOV     #26,$TMP1       ;SYNC CHAR TO BE SHIFTED IN
2016  006612  004767  010330                    JSR     PC,RPOKE        ;SHIFT IN THIS SYNC CHAR
2017  006616  032777  004000  173066            BIT     #RECACT,@RXCSR  ;RECACT = 0 ?
2018  006624  001401                            BEQ     .+4
2019  006626  104004                            ERROR   4               ;RECACT SHOULD BE 0
2020  006630  012767  000010  172264            MOV     #8.,SHIFT       ;# OF SHIFTS
2021  006636  012767  000025  172634            MOV     #25,$TMP1       ;ANY CHARACTER
2022  006644  004767  010276                    JSR     PC,RPOKE        ;SHIFT IN THIS CHARACTER
2023                                  ;YOU HAVE JUST LOST SYNCRONIZATION.....................
2024                                  ;POKE THE CLK TWICE TO GET INTO SYNCRONIZATION
2025                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2026  006650  042777  020000  173050            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2027  006656  052777  020000  173042            BIS     #CLK,@TXCSR     ;POKE CLK UP
2028                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2029  006664  042777  020000  173034            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2030  006672  052777  020000  173026            BIS     #CLK,@TXCSR     ;POKE CLK UP
2031  006700  012767  000002  172216            MOV     #2,COUNT        ;# OF SYNC CHARS
2032  006706  032777  004000  172776  1$:       BIT     #RECACT,@RXCSR  ;RECACT = 0 ?
2033  006714  001401                            BEQ     .+4
2034  006716  104004                            ERROR   4               ;RECACT SHOULD BE  0
2035  006720  012767  000010  172174            MOV     #8.,SHIFT       ;# OF SHIFTS
2036  006726  012767  000026  172544            MOV     #26,$TMP1       ;SYNC CHAR
2037  006734  004767  010206                    JSR     PC,RPOKE        ;SHIFT IN THIS SYNC CHAR
2038  006740  005367  172160                    DEC     COUNT
2039  006744  001360                            BNE     1$      ;IS COUNT = 0 ? NO GO AGAIN
2040  006746  032777  004000  172736            BIT     #RECACT,@RXCSR  ;RECACT = 1 ?
2041  006754  001001                            BNE     .+4
2042  006756  104004                            ERROR   4       ;RECACT SHOULD BE ASSERTED
2043  006760                          2$:
2044                                  ;;THIS TEST VERIFYS MODE SELECT.....
2045                                  ;;SYNC EXTERNAL VS. SYNC INTERNAL
2046                                  ;;
2047                                  ;BASICALLY THE TEST CHECKS THAT THE RECEIVED
2048                                  ;DATA FREEZES  IN SYNC INTERNAL
2049                                  ;IN SYNC EXTERNAL THIS DATA IS TRANSPARENT
2050                                  ;THIS TEST ONLY APPLIES TO THE RECEIVER SECTION
2051                                  ;;LENGTH: EIGHT
2052                                  ;NOTE:SEARCH SYNC IS NOT SET
2053                                  ;;*****************************************************************
2054  006760  000004                  TST16:  SCOPE
2055  006762  052777  000400  172736            BIS     #MRESET,@TXCSR  ;MASTER RESET
2056  006770  012777  030000  172724            MOV     #SYNINT,@PARCSR ;SET THE MODE
2057  006776  052777  000400  172722            BIS     #MRESET,@TXCSR  ;MASTER RESET
2058
2059                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2060  007004  012777  064001  172714            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2061
2062                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2063  007012  012777  036026  172702            MOV     #SYNINT!EIGHT!NOPAR!26,@PARCSR
2064                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2065  007020  042777  020000  172700            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2066  007026  052777  020000  172672            BIS     #CLK,@TXCSR     ;POKE CLK UP
2067                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2068  007034  042777  020000  172664            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2069  007042  052777  020000  172656            BIS     #CLK,@TXCSR     ;POKE CLK UP
2070  007050  012767  000010  172044            MOV     #8.,SHIFT       ;# OF SHIFTS
```

```
DZDUU-A MACY11 27(1006)  03-FEB-77  08:00  PAGE 45
DZDUUA.M11    13-OCT-76 08:28          INITIALIZE THE COMMON TAGS

  2071  007056  012767  000125  172414          MOV     #125,$TMP1      ;DATA CHARACTER
  2072  007064  016703  172626                  MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
  2073  007070  042777  040000  172630  1$:     BIC     #MTDATA,@TXCSR  ;CLEAR MAINT DATA
  2074  007076  000241                          CLC
  2075  007100  006067  172374                  ROR     $TMP1   ;FORCE CARRY
  2076  007104  103003                          BCC     2$
  2077  007106  052777  040000  172612          BIS     #MTDATA,@TXCSR  ;SET MTDATA
  2078  007114  042777  020000  172604  2$:     BIC     #CLK,@TXCSR     ;POKE CLK
  2079  007122  052777  020000  172576          BIS     #CLK,@TXCSR     ;
  2080  007130  012700  000377                  MOV     #377,R0 ;EXPECTED
  2081  007134  017701  172556                  MOV     @RXDBUF,R1      ;ACTUAL
  2082  007140  020001                          CMP     R0,R1
  2083  007142  001401                          BEQ     .+4
  2084  007144  104002                          ERROR   2
  2085                                          ;DATA CHARACTER SHOULD COMPARE.....
  2086  007146  005367  171750                  DEC     $SHIFT  ;IS IT THE LAST SHIFT ?
  2087  007152  001346                          BNE     1$      ;NO ...SHIFT SOME MORE
  2088  007154  052777  000400  172544          BIS     #MRESET,@TXCSR  ;MASTER RESET
  2089  007162  012777  020000  172532          MOV     #SYNEXT,@PARCSR ;SET THE MODE
  2090  007170  052777  000400  172530          BIS     #MRESET,@TXCSR  ;MASTER RESET
  2091
  2092                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
  2093  007176  012777  064001  172522          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
  2094
  2095                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
  2096  007204  012777  026026  172510          MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
  2097                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
  2098  007212  042777  020000  172506          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
  2099  007220  052777  020000  172500          BIS     #CLK,@TXCSR     ;POKE CLK UP
  2100  007226  012767  000010  171666          MOV     #8.,SHIFT       ;# OF SHIFTS
  2101  007234  012767  000125  172236          MOV     #125,$TMP1      ;DATA CHARACTER
  2102  007242  005067  172234                  CLR     $TMP2
  2103  007246  105167  172230                  COMB    $TMP2   ;MAKE LOW BYTE ALL 1'S
  2104                                          ;TO MATCH RXDBUF'S CONTENTS AFTER A MASTER RESET
  2105  007252  042777  040000  172446  5$:     BIC     #MTDATA,@TXCSR  ;CLR MAINT DATA
  2106  007260  000241                          CLC
  2107  007262  006067  172212                  ROR     $TMP1   ;FORCE CARRY
  2108  007266  103003                          BCC     6$
  2109  007270  052777  040000  172430          BIS     #MTDATA,@TXCSR
  2110  007276  106067  172200          6$:     RORB    $TMP2   ;PICK UP CARRY BIT
  2111  007302  042777  020000  172416          BIC     #CLK,@TXCSR
  2112  007310  052777  020000  172410          BIS     #CLK,@TXCSR
  2113  007316  016700  172160                  MOV     $TMP2,R0        ;EXPECTED
  2114  007322  017701  172370                  MOV     @RXDBUF,R1      ;ACTUAL
  2115  007326  020001                          CMP     R0,R1
  2116  007330  001401                          BEQ     .+4
  2117  007332  104002                          ERROR   2       ;DATA CHARACTER SHOULD COMPARE...
  2118                                          ;THE DATA CHARACTER SHOULD BE SEEN AS IT
  2119                                          ;SHIFTS ACROSS THE RECEIVER DATA OUTPUT
  2120  007334  005367  171562                  DEC     SHIFT
  2121  007340  001344                          BNE     5$
  2122
  2123                                  ;;THIS TEST VERIFYS TX DONE FUNCTION, DONE = 1
  2124                                  ;;MODE:  SYNC INTERNAL
  2125                                  ;;PARITY:  NO PARITY (NOPAR)
  2126                                  ;;LENGTH:  EIGHT
```

```
                                        INITIALIZE THE COMMON TAGS

2127                                    ;;
2128                                    ;;**************************************************************************
2129  007342  000004                    TST17:  SCOPE
2130
2131  007344  052777  000400  172354            BIS     #MRESET,@TXCSR  ;MASTER RESET
2132  007352  012777  030000  172342            MOV     #SYNINT,@PARCSR ;SET THE MODE
2133  007360  052777  000400  172340            BIS     #MRESET,@TXCSR  ;MASTER RESET
2134
2135                                    ;SET MAINTENANCE MODE & SEND
2136                                            ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2137  007366  012777  004020  172332            MOV     #MINT!SEND,@TXCSR
2138
2139                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2140  007374  012777  036026  172320            MOV     #SYNINT!EIGHT!NOPAR!26,@PARCSR
2141
2142  007402  105777  172320                    TSTB    @TXCSR          ;TXDONE?
2143  007406  100401                            BMI     .+4
2144  007410  104004                            ERROR   4                       ;TXDONE SHOULD BE SET
2145  007412  112777  000021  172312            MOVB    #21,@TXDBUF     ;LOAD ANY CHAR
2146  007420  052777  020000  172300            BIS     #CLK,@TXCSR     ;POKE CLK UP
2147  007426  042777  020000  172272            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2148  007434  105777  172266                    TSTB    @TXCSR
2149  007440  100001                            BPL     .+4
2150  007442  104004                            ERROR   4                       ;TXDONE SHOULD BE CLR
2151
2152  007444  052777  020000  172254            BIS     #CLK,@TXCSR     ;POKE CLK UP
2153  007452  042777  020000  172246            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2154  007460  105777  172242                    TSTB    @TXCSR
2155  007464  100401                            BMI     .+4
2156  007466  104004                            ERROR   4                       ;TXDONE SHOULD BE SET
2157
2158
2159
2160                                    ;;THIS TEST PROVES THAT RXERR FREEZES THE "RECEIVER
2161                                    ;;RESET" WHILE IN STRIP SYNC MODE
2162                                    ;;THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR
2163                                    ;;WHEN STRIP SYNC IS SET AND SYNC CHARACTERS ARE SENT
2164                                    ;;BUT IF AN ERROR SHOULD OCCUR...THIS AUTOMATIC RESET
2165                                    ;;IS DISCOMBOBULATED
2166                                    ;;IE:  FORCE OVERRUN (OVERRUN) WHILE STRIP SYNC IS SET
2167                                    ;;BY TRANSMITTING A DATA CHARACTER THEN TRANSMIT A SYNC CHARACTER
2168                                    ;;AND DON'T READ THAT DATA CHARACTER.  NOTE:  NORMALLY THE LOGIC
2169                                    ;;RESETS THE RXDONE & ERROR FLAGS PROVIDING THAT ONLY SYNC CHARACTERS ARE
2170                                    ;;STRIPPED
2171                                    ;;MODE:  SYNC EXTERNAL (SYNEXT)
2172                                    ;;LENGTH:  EIGHT
2173                                    ;;NOTE:  THIS TEST USES BOTH RECEIVER AND TRANSMITTER LOGIC
2174                                    ;;
2175
2176                                    ;;**************************************************************************
2177  007470  000004                    TST20:  SCOPE
2178
2179  007472  052777  000400  172226            BIS     #MRESET,@TXCSR  ;MASTER RESET
2180  007500  012777  020000  172214            MOV     #SYNEXT,@PARCSR ;SET THE MODE
2181  007506  052777  000400  172212            BIS     #MRESET,@TXCSR  ;MASTER RESET
2182
```

INITIALIZE THE COMMON TAGS

```
                                        ;SET MAINTENANCE MODE & SEND
                                                ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
007514  012777  004020  172204                  MOV     #MINT!SEND,@TXCSR

                                        ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
007522  012777  026026  172172                  MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
007530  112777  000026  172174                  MOVB    #26,@TXDBUF     ;LOAD SYNC CHAR
007536  052777  000420  172146                  BIS     #SYNSCH!STPSYN,@RXCSR   ;SET SYNC SEARCH & STRIP SYNC
007544  016703  172146                          MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
007550  012767  000003  171346                  MOV     #3,COUNT        ;# OF TIMES SYNC WILL BE SENT
007556  052777  020000  172142                  BIS     #CLK,@TXCSR     ;POKE CLK UP
007564  042777  020000  172134                  BIC     #CLK,@TXCSR     ;POKE CLK DOWN
007572  012767  000010  171322  2$:             MOV     #8.,SHIFT       ;# OF SHIFTS
007600                        1$:
007600  052777  020000  172120                  BIS     #CLK,@TXCSR     ;POKE CLK UP
007606  042777  020000  172112                  BIC     #CLK,@TXCSR     ;POKE CLK DOWN
007614  005367  171302                          DEC     SHIFT
007620  001367                                  BNE     1$
007622  105777  172064                          TSTB    @RXCSR          ;RXDONE?
007626  100001                                  BPL     .+4
007630  104004                                  ERROR   4                       ;RXDONE SHOULD NOT ASSERT
007632  005367  171266                          DEC     COUNT
007636  001355                                  BNE     2$
007640  012700  000026                          MOV     #26,R0          ;EXPECTED
007644  017701  172046                          MOV     @RXDBUF,R1      ;ACTUAL
007650  020001                                  CMP     R0,R1
007652  001401                                  BEQ     .+4
007654  104002                                  ERROR   2               ;NOTE THAT OVERRUN SHOULD NOT OCCUR, ALSO
                                                                        ;SECOND & 3RD SYNC CHARACTER CAME FROM
                                                                        ;SYNC HOLDING REGISTER
007656  012767  000003  171240                  MOV     #3,COUNT        ;# OF TIMES
007664  112777  000025  172040                  MOVB    #25,@TXDBUF     ;LOAD ANY CHAR....HOWEVER...
                                                ;ONE MORE SYNC CHAR WILL BE SENT BEFORE
                                                ;THE "25" CHAR IS SENT (THE DNA BIT IS
                                                ;ALREADY UP)

007672  012767  000010  171222  4$:             MOV     #8.,SHIFT       ;# OF SHIFTS

007700                        3$:
007700  052777  020000  172020                  BIS     #CLK,@TXCSR     ;POKE CLK UP
007706  042777  020000  172012                  BIC     #CLK,@TXCSR     ;POKE CLK DOWN
007714  005367  171202                          DEC     SHIFT
007720  001367                                  BNE     3$
007722  005367  171176                          DEC     COUNT
007726  001361                                  BNE     4$
007730  105777  171756                          TSTB    @RXCSR  ;RXDONE = 1 ?
007734  100401                                  BMI     .+4
007736  104004                                  ERROR   4       ;RXDONE SHOULD BE SET
007740  012700  140026                          MOV     #RXERR!OVRRUN!26,R0 ;EXPECTED
007744  017701  171746                          MOV     @RXDBUF,R1      ;ACTUAL
007750  020001                                  CMP     R0,R1
007752  001401                                  BEQ     .+4
007754  104002                                  ERROR   2               ;NOTE THAT OVRRUN SHOULD OCCUR,
                                                                        ;ALSO SECOND SYNC CHARACTER CAME
                                                                        ;FROM SYNC HOLDING REGISTER
                                                                        ;SUMMARY:  THE OVRRUN STOPPED
```

INITIALIZE THE COMMON TAGS

```
                                                        ;THE AUTOMATIC RESETTING OF
                                                        ;RXDONE & ERROR FLAGS.....CHECK THIS

                                ;;THIS TEST VERIFYS THAT EITHER SEQUENCE OF
                                ;;LOADING TXDBUF AND SETTING SEND
                                ;;DOES CAUSE TRANSMISSION
                                ;;MODE:  SYNC EXT
                                ;;LENGTH:  EIGHT
                                ;;
                                ;;**********************************************************************
007756  000004                 TST21:  SCOPE

007760  052777  000400  171740          BIS     #MRESET,@TXCSR  ;MASTER RESET
007766  012777  020000  171726          MOV     #SYNEXT,@PARCSR ;SET THE MODE
007774  052777  000400  171724          BIS     #MRESET,@TXCSR  ;MASTER RESET

                                ;SET MAINTENANCE MODE & SEND
                                        ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
010002  012777  004020  171716          MOV     #MINT!SEND,@TXCSR

                                ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
010010  012777  026026  171704          MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
010016  016703  171704                  MOV     TXCSR,R3        ;SET UP FOR ERROR MESSAGE
010022  042777  000020  171676          BIC     #SEND,@TXCSR    ;DROP SEND
010030  012767  000025  171442          MOV     #25,$TMP1
010036  112777  000025  171666          MOVB    #25,@TXDBUF     ;LOAD CHARACTER
010044  052777  000020  171654          BIS     #SEND,@TXCSR    ;SET SEND
                                ;GET INTO SYNCRONIZATION
010052  052777  020000  171646          BIS     #CLK,@TXCSR     ;POKE CLK UP
010060  042777  020000  171640          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
010066  012767  000010  171026          MOV     #8.,$HIFT       ;# OF SHIFTS
010074  005000                 1$:      CLR     R0
010076  006067  171376                  ROR     $TMP1
010102  103002                          BCC     2$
010104  052700  002000                  BIS     #BITW,R0        ;EQUIV OF BIT WINDOW

010110                           2$:
010110  052777  020000  171610          BIS     #CLK,@TXCSR     ;POKE CLK UP
010116  042777  020000  171602          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
010124  017701  171576                  MOV     @TXCSR,R1       ;ACTUAL
010130  042701  075777                  BIC     #075777,R1      ;SAVE BIT WINDOW & DNA
010134  020001                          CMP     R0,R1
010136  001401                          BEQ     .+4
010140  104003                          ERROR   3               ;BIT WINDOW DID NOT MATCH ACTUAL DATA BIT
                                                                ;ALSO CHECK DNA
010142  005367  170754                  DEC     SHIFT
010146  001352                          BNE     1$


                                ;THIS TEST VERIFYS THAT DROPPING OF SEND IN THE
                                ;;MIDDLE OF TRANSMITTING A CHARACTER DOES INDEED
                                ;;FINISH TRANSMITTING THAT CHARACTER
                                ;;MODE:  SYNC EXT
                                ;;LENGTH:  EIGHT
                                ;;
```

INITIALIZE THE COMMON TAGS

```
2295
2296                                    ;;*****************************************************************
2297  010150  000004                   TST22:  SCOPE
2298
2299  010152  052777  000400  171546           BIS     #MRESET,@TXCSR  ;MASTER RESET
2300  010160  012777  020000  171534           MOV     #SYNEXT,@PARCSR ;SET THE MODE
2301  010166  052777  000400  171532           BIS     #MRESET,@TXCSR  ;MASTER RESET
2302
2303                                    ;SET MAINTENANCE MODE & SEND
2304                                            ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2305  010174  012777  004020  171524           MOV     #MINT!SEND,@TXCSR
2306
2307                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2308  010202  012777  026026  171512           MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
2309  010210  016703  171512                   MOV     TXCSR,R3        ;SETUP FOR ERROR MESSAGE
2310  010214  112777  000252  171510           MOVB    #252,@TXDBUF    ;LOAD DATA CHAR.
2311  010222  012767  000252  171250           MOV     #252,STMP1      ;SHIFTED CHAR
2312  010230  012767  000010  170664           MOV     #8.,SHIFT       ;# OF SHIFTS
2313                                            ;GET INTO SYNCRONIZATION
2314  010236  052777  020000  171462           BIS     #CLK,@TXCSR     ;POKE CLK UP
2315  010244  042777  020000  171454           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2316
2317  010252  005000                   1$:     CLR     R0
2318  010254  006067  171220                   ROR     STMP1           ;FORCE CARRY
2319  010260  103002                           BCC     2$
2320  010262  052700  002000                   BIS     #BITW,R0        ;EQUIV OF BIT WINDOW
2321
2322  010266                           2$:
2323  010266  052777  020000  171432           BIS     #CLK,@TXCSR     ;POKE CLK UP
2324  010274  042777  020000  171424           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2325  010302  017701  171420                   MOV     @TXCSR,R1       ;ACTUAL
2326  010306  042701  075777                   BIC     #075777,R1      ;SAVE ONLY BIT WINDOW & DNA
2327  010312  020001                           CMP     R0,R1
2328  010314  001401                           BEQ     .+4
2329  010316  104003                           ERROR   3               ;BIT WINDOW DID NOT MATCH
2330                                                                   ;ACTUAL DATA BIT
2331  010320  005367  170576                   DEC     SHIFT
2332  010324  022767  000003  170570           CMP     #3,SHIFT
2333  010332  001003                           BNE     3$
2334  010334  042777  000020  171364           BIC     #SEND,@TXCSR    ;DROP SEND
2335  010342  005767  170554           3$:     TST     SHIFT
2336  010346  001341                           BNE     1$              ;DO IT AGAIN?
2337
2338
2339
2340                                    ;;THIS TEST VERIFYS THAT RXDONE ASSERTS WHEN STRIP SYNC IS SET
2341                                    ;;MODE: SYNC INTERNAL
2342                                    ;;LENGTH: EIGHT
2343                                    ;;
2344                                    ;;*****************************************************************
2345  010350  000004                   TST23:  SCOPE
2346  010352  052777  000400  171346           BIS     #MRESET,@TXCSR  ;MASTER RESET
2347  010360  012777  030000  171334           MOV     #SYNINT,@PARCSR ;SET THE MODE
2348  010366  052777  000400  171332           BIS     #MRESET,@TXCSR  ;MASTER RESET
2349
2350                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
```

```
                                        INITIALIZE THE COMMON TAGS

2351  010374  012777  064001  171324            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2352
2353                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2354  010402  012777  036026  171312            MOV     #SYNINT!EIGHT!NOPAR!26,@PARCSR
2355
2356  010410  052777  000420  171274            BIS     #SYNSCH!STPSYN,@RXCSR   ;SET SYNC SEARCH &
2357                                                                            ;STRIP SYNC
2358                                            ;POKE CLK TO GET RECEIVER INTO SYNCRONIZATION.....
2359  010416  042777  020000  171302            BIC     #CLK,@TXCSR     ;POKE CLK
2360  010424  052777  020000  171274            BIS     #CLK,@TXCSR     ;
2361                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2362  010432  042777  020000  171266            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2363  010440  052777  020000  171260            BIS     #CLK,@TXCSR     ;POKE CLK UP
2364  010446  016703  171244                    MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
2365  010452  012767  000002  170444            MOV     #2,COUNT        ;# OF TIMES OF SYNC CHARS.
2366                                            ;TEST TO SEE HOW MANY SYNC CHARS  NEEDED
2367  010460  105767  170462                    TSTB    SYNCNO
2368  010464  100402                            BMI     3$      ;WILL IT BE ONE OR TWO ?
2369  010466  005367  170432                    DEC     COUNT   ;MAKE IT ONE LESS
2370  010472  012767  000010  170422  3$:       MOV     #8.,SHIFT       ;#OF SHIFTS
2371  010500  012767  000026  170772            MOV     #26,$TMP1       ;SYNC CHAR
2372  010506  004767  006434                    JSR     PC,RPOKE
2373  010512  005367  170406                    DEC     COUNT   ;IS IT THE LAST SYNC CHAR ?
2374  010516  001365                            BNE     3$      ;GO AGAIN AND SHIFT IN ANOTHER SYNC CHAR
2375  010520  032777  004000  171164            BIT     #RECACT,@RXCSR  ;RECACT=1?
2376  010526  001001                            BNE     .+4
2377  010530  104004                            ERROR   4       ;RECACT SHOULD BE ASSERTED
2378  010532  105777  171154                    TSTB    @RXCSR          ;RXDONE=0?
2379  010536  100001                            BPL     .+4
2380  010540  104004                            ERROR   4                       ;RXDONE SHOULD NOT BE ASSERTED
2381  010542  012767  000010  170352            MOV     #8.,SHIFT       ;#OF SHIFTS
2382  010550  012767  000025  170722            MOV     #25,$TMP1       ;ANY CHARACTER
2383  010556  004767  006364                    JSR     PC,RPOKE
2384  010562  105777  171124                    TSTB    @RXCSR          ;RXDONE=1?
2385  010566  100401                            BMI     .+4
2386  010570  104004                            ERROR   4                       ;RXDONE SHOULD NOW BE ASSERTED
2387  010572  012700  000025                    MOV     #25,R0          ;EXPECTED
2388  010576  017701  171114                    MOV     @RXDBUF,R1      ;ACTUAL
2389  010602  020001                            CMP     R0,R1
2390  010604  001401                            BEQ     .+4
2391  010606  104002                            ERROR   2               ;CHARACTERS SHOULD BE MATCHED
2392
2393
2394
2395                                    ;;THIS TEST VERIFYS THAT BY DROPPING SYNSCH
2396                                    ;;IN THE MIDDLE OF A CHARACTER, SYNC CHARACTER SEQUENCE
2397                                    ;;IS NEEDED BEFORE RECACT, RXDONE ASSERT AGAIN.
2398                                    ;;ALSO NOTE:  SINCE RECACT IS DEPENDENT ON MATCH DETECT,
2399                                    ;;AND IF SYNSCH IS DROPPED IN THE MIDDLE OF
2400                                    ;;A SYNC CHARACTER AND THEN RAISED AGAIN ;RXDONE SHOULD
2401                                    ;;NOT ASSERT UNTIL NEW SYNC CHARACTER SEQUENCE.....HOWEVER
2402                                    ;;ONE "PAD" CHARACTER MUST PRECEED THIS JUGGLING OF SEARCH SYNC (SYNSCH)
2403                                    ;;MODE:  SYNC INTERNAL (SYNINT)
2404                                    ;;LENGTH:  EIGHT
2405                                    ;;
2406
```

```
                                        INITIALIZE THE COMMON TAGS

2407                                    ;;*****************************************************************
2408  010610  000004                    TST24:  SCOPE
2409
2410  010612  052777  000400  171106            BIS     #MRESET,@TXCSR  ;MASTER RESET
2411  010620  012777  030000  171074            MOV     #SYNINT,@PARCSR ;SET THE MODE
2412  010626  052777  000400  171072            BIS     #MRESET,@TXCSR  ;MASTER RESET
2413
2414                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2415  010634  012777  064001  171064            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2416
2417                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2418  010642  012777  036026  171052            MOV     #SYNINT!EIGHT!NOPAR!26,@PARCSR
2419
2420  010650  052777  000020  171034            BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2421                                            ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2422  010656  042777  020000  171042            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2423  010664  052777  020000  171034            BIS     #CLK,@TXCSR     ;POKE CLK UP
2424                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2425  010672  042777  020000  171026            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2426  010700  052777  020000  171020            BIS     #CLK,@TXCSR     ;POKE CLK UP
2427  010706  012767  000002  170210            MOV     #2,COUNT        ;# OF TIMES
2428  010714  016703  170776                    MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
2429  010720  012767  000010  170174  1$:       MOV     #8.,SHIFT       ;# OF SHIFTS
2430  010726  012767  000026  170544            MOV     #26,$TMP1       ;SYNC CHAR.
2431  010734  004767  006206                    JSR     PC,RPOKE
2432  010740  005367  170160                    DEC     COUNT
2433  010744  001403                            BEQ     2$
2434                                    ;TEST TO SEE HOW MANY SYNC CHARACTERS NEEDED
2435  010746  105767  170174                    TSTB    SYNCNO
2436  010752  100762                            BMI     1$
2437
2438  010754  032777  004000  170730  2$:       BIT     #RECACT,@RXCSR  ;RECACT=1?
2439  010762  001001                            BNE     .+4
2440  010764  104004                            ERROR   4                       ;RECACT SHOULD BE SET
2441  010766  105777  170720                    TSTB    @RXCSR  ;RXDONE = 0?
2442  010772  100001                            BPL     .+4
2443  010774  104004                            ERROR   4       ;RXDONE SHOULD = 0
2444                                    ;THE FOLLOWING ALLOWS THE RECEIVER "CHIP" TO RECOGNIZE THE DROPPING OF SYNSCH
2445                                    ;...BASICALLY IT SIMULATES THE "PAD" CHARACTER REQUIREMENT AT THE END
2446                                    ;OF DATA TRANSFER.
2447                                    ;ONE "PAD" CHARACTER  IS REQUIRED TO FLUSH OUT SYNC CHARACTER
2448                                    ;AND ...YOU HAVE TO DROP SYNSCH ON THE NEXT OR FOLLOWING CHARACTER
2449                                    ;FOR A "CLEAN" RESTART
2450  010776  012767  000010  170116            MOV     #8.,SHIFT       ;# OF SHIFTS
2451  011004  012767  000025  170466            MOV     #25,$TMP1       ;"PAD" CHARACTER
2452  011012  004767  006130                    JSR     PC,RPOKE        ;SHIFT IN "PAD" CHARACTER
2453  011016  105777  170670                    TSTB    @RXCSR  ;RXDONE = 1 ?
2454  011022  100401                            BMI     .+4
2455  011024  104004                            ERROR   4       ;RXDONE SHOULD = 1
2456  011026  017701  170664                    MOV     @RXDBUF,R1      ;STORE ACTUAL & CLEAR RXDONE
2457  011032  012700  000025                    MOV     #25,R0  ;EXPECTED
2458  011036  020001                            CMP     R0,R1   ;COMPARE EXPECTED DATA VS ACTUAL DATA
2459  011040  001401                            BEQ     .+4
2460  011042  104002                            ERROR   2       ;DATA SHOULD COMPARE WITHOUT ERROR FLAGS
2461                                    ;THE FOLLOWING IS THE SYNC CHARACTER AFTER THE "PAD" CHARACTER
2462                                    ;IN WHICH SEARCH SYNC (SYNSCH) IS JUGGLED TO CREATE AN ABORT SITUATION
```

```
                                        INITIALIZE THE COMMON TAGS

 2463                                   ;AT THE END OF A DATA TRANSFER SEQUENCE
 2464  011044  012767  000004  170050          MOV     #4,SHIFT        ;# OF SHIFTS
 2465  011052  012767  000026  170420          MOV     #26,$TMP1       ;SYNC CHAR.
 2466  011060  004767  006062                  JSR     PC,RPOKE
 2467  011064  032777  004000  170620          BIT     #RECACT,@RXCSR  ;RECACT=1?
 2468  011072  001001                          BNE     .+4
 2469  011074  104004                          ERROR   4                       ;RECACT SHOULD  STILL BE SET
 2470  011076  105777  170610                  TSTB    @RXCSR  ;RXDONE = 0 ?
 2471  011102  100001                          BPL     .+4
 2472  011104  104004                          ERROR   4       ;RXDONE SHOULD = 0 FROM PREVIOUS READING OF RXDBUF
 2473  011106  042777  000020  170576          BIC     #SYNSCH,@RXCSR  ;DROP SEARCH SYNC
 2474  011114  032777  004000  170570          BIT     #RECACT,@RXCSR  ;RECACT=0?
 2475  011122  001401                          BEQ     .+4
 2476  011124  104004                          ERROR   4                       ;RECACT SHOULD NOT BE SET
 2477                                  ;NOW SHIFT TWO BITS TO ALLOW SEARCH SYNC =0 TO TAKE
 2478                                  ;EFFECT IN THE LOGIC(THIS ALLOWS THE RECEIVER CHIP TO SEE
 2479                                  ;THE DROPPING OF SEARCH SYNC)
 2480  011126  012767  000002  167766          MOV     #2,SHIFT        ;# OF SHIFTS
 2481  011134  004767  006006                  JSR     PC,RPOKE
 2482  011140  052777  000020  170544          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
 2483  011146  032777  004000  170536          BIT     #RECACT,@RXCSR
 2484  011154  001401                          BEQ     .+4
 2485  011156  104004                          ERROR   4       ;RECACT =0 ?
 2486  011160  105777  170526                  TSTB    @RXCSR  ;RXDONE = 0 ?
 2487  011164  100001                          BPL     .+4
 2488  011166  104004                          ERROR   4       ;RXDONE = 0 ?
 2489  011170  012767  000002  167724          MOV     #2,SHIFT        ;# OF SHIFTS TO FINISH UP THE SYNC CHARACTER
 2490  011176  004767  005744                  JSR     PC,RPOKE
 2491  011202  032777  004000  170502          BIT     #RECACT,@RXCSR  ;RECACT=0?
 2492  011210  001401                          BEQ     .+4
 2493  011212  104004                          ERROR   4                       ;RECACT SHOULD NOT BE SET
 2494  011214  105777  170472                  TSTB    @RXCSR          ;RXDONE=0?
 2495  011220  100001                          BPL     .+4
 2496  011222  104004                          ERROR   4                       ;RXDONE SHOULD NOT BE ASSERTED
 2497  011224  012700  000025                  MOV     #25,R0          ;EXPECTED
 2498  011230  017701  170462                  MOV     @RXDBUF,R1      ;ACTUAL
 2499  011234  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS ACTUAL
 2500  011236  001401                          BEQ     .+4
 2501  011240  104002                          ERROR   2                       ;CHARACTERS SHOULD BE MATCHED.....
 2502                                          ;REMBEMBER THAT THE " 25 " FROZE WHEN SYNSCH WAS DROPPED
 2503                                          ;THEREFORE ....25 WILL BE READ AND NOT 26
 2504                                  ;THE FOLLOWING VERIFYS THAT THE RE SYNCRONIZATION COMES UP "CLEAN"
 2505  011242  012767  000002  167654          MOV     #2,COUNT        ;# OF TIMES OF SYNC CHARS.
 2506                                          ;
 2507                                          ;TEST TO SEE HOW MANY SYNC CHARS  NEEDED
 2508  011250  105767  167672                  TSTB    SYNCNO  ;HOW MANY SYNCS ARE YOU STRAPPED FOR ?
 2509  011254  100402                          BMI     3$      ;WILL IT BE ONE OR TWO ?
 2510  011256  005367  167642                  DEC     COUNT   ;IT WAS ONLY ONE NEEDED
 2511  011262  012767  000010  167632  3$:     MOV     #8.,SHIFT       ;#OF SHIFTS
 2512  011270  012767  000026  170202          MOV     #26,$TMP1       ;SYNC CHAR
 2513  011276  004767  005644                  JSR     PC,RPOKE
 2514  011302  005367  167616                  DEC     COUNT   ;IS IT THE LAST SYNC CHAR ?
 2515  011306  001365                          BNE     3$      ;GO AGAIN AND SHIFT IN ANOTHER SYNC CHAR
 2516  011310  032777  004000  170374          BIT     #RECACT,@RXCSR  ;RECACT=1?
 2517  011316  001001                          BNE     .+4
 2518  011320  104004                          ERROR   4       ;RECACT SHOULD BE ASSERTED
```

```
2519  011322  105777  170364                  TSTB    @RXCSR          ;RXDONE=0?
2520  011326  100001                          BPL     .+4
2521  011330  104004                          ERROR   4                       ;RXDONE SHOULD NOT BE ASSERTED
2522  011332  012767  000010  167562          MOV     #8.,SHIFT       ;#OF SHIFTS
2523  011340  012767  000025  170132          MOV     #25,$TMP1       ;ANY CHARACTER
2524  011346  004767  005574                  JSR     PC,RPOKE
2525  011352  105777  170334                  TSTB    @RXCSR          ;RXDONE=1?
2526  011356  100401                          BMI     .+4
2527  011360  104004                          ERROR   4                       ;RXDONE SHOULD NOW BE ASSERTED
2528  011362  012700  000025                  MOV     #25,R0          ;EXPECTED
2529  011366  017701  170324                  MOV     @RXDBUF,R1      ;ACTUAL
2530  011372  020001                          CMP     R0,R1
2531  011374  001401                          BEQ     .+4
2532  011376  104002                          ERROR   2               ;CHARACTERS SHOULD BE MATCHED
2533
2534
2535
2536                                  ;;THIS TEST VERIFYS THAT HDX MODE DISQUALIFIES THE
2537                                  ;;RECEIVER WHEN SEND IS ASSERTED
2538                                  ;;MODE:  SYNC EXT
2539                                  ;;LENGTH:  EIGHT
2540                                  ;;NOTE:  THIS TEST WORKS ONLY IN MAINT. EXTERNAL MODE
2541                                  ;;THIS TEST USES BOTH RECEIVER & TRANSMITTER LOGIC
2542                                  ;;*****************************************************************
2543  011400  000004                  TST25:  SCOPE
2544  011402  105767  167545                  TSTB    JMRBY
2545  011406  100155                          BPL     1$              ;GET OUT OF THIS TEST IF "NO"
2546  011410  016703  170302                  MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
2547  011414  052777  000400  170304          BIS     #MRESET,@TXCSR  ;MASTER RESET
2548  011422  012777  020000  170272          MOV     #SYNEXT,@PARCSR ;SET THE MODE
2549  011430  052777  000400  170270          BIS     #MRESET,@TXCSR  ;MASTER RESET
2550
2551                                  ;SET MAINTENANCE MODE & SEND
2552                                          ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2553  011436  012777  010020  170262          MOV     #MEXT!SEND,@TXCSR
2554
2555                                  ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2556  011444  012777  026026  170250          MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
2557  011452  052777  000020  170232          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2558  011460  112777  000025  170244          MOVB    #25,@TXDBUF     ;ANY CHARACTER
2559                                  ;POKE CLK FOR SYNCRONIZATION
2560  011466  052777  020000  170232          BIS     #CLK,@TXCSR     ;POKE CLK UP
2561                                  ;WAIT FOR CABLE & DRIVER DELAYS
2562  011474  016702  167420                  MOV     HOLD,R2 ;WAIT THIS AMT
2563  011500  005302                          DEC     R2      ;WAIT
2564  011502  001376                          BNE     .-2
2565                                  ;EXIT...
2566  011504  042777  020000  170214          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2567                                  ;WAIT FOR CABLE & DRIVER DELAYS
2568  011512  016702  167402                  MOV     HOLD,R2 ;WAIT THIS AMT
2569  011516  005302                          DEC     R2      ;WAIT
2570  011520  001376                          BNE     .-2
2571                                  ;EXIT...
2572  011522  012767  000010  167372          MOV     #8.,SHIFT       ;# OF SHIFTS
2573  011530                          2$:
2574  011530  052777  020000  170170          BIS     #CLK,@TXCSR     ;POKE CLK UP
```

```
2575
2576  011536  016702  167356                  ;WAIT FOR CABLE & DRIVER DELAYS
                                              MOV     HOLD,R2 ;WAIT THIS AMT
2577  011542  005302                          DEC     R2      ;WAIT
2578  011544  001376                          BNE     .-2
2579                                          ;EXIT...
2580  011546  042777  020000  170152          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2581                                          ;WAIT FOR CABLE & DRIVER DELAYS
2582  011554  016702  167340                  MOV     HOLD,R2 ;WAIT THIS AMT
2583  011560  005302                          DEC     R2      ;WAIT
2584  011562  001376                          BNE     .-2
2585                                          ;EXIT...
2586  011564  005367  167332                  DEC     SHIFT   ;# OF SHIFTS
2587  011570  022767  000003  167324          CMP     #3,SHIFT        ;IS IT TIME TO LOAD NEXT CHAR ?
2588  011576  001003                          BNE     3$
2589  011600  112777  000024  170124          MOVB    #24,@TXDBUF     ;LOAD NEXT CHAR.
2590  011606  005767  167310          3$:     TST     SHIFT
2591  011612  001346                          BNE     2$
2592  011614  105777  170072                  TSTB    @RXCSR          ;RXDONE=1?
2593  011620  100401                          BMI     .+4
2594  011622  104004                          ERROR   4                       ;RXDONE SHOULD BE SET
2595  011624  012700  000025                  MOV     #25,R0          ;EXPECTED
2596  011630  017701  170062                  MOV     @RXDBUF,R1      ;ACTUAL
2597  011634  020001                          CMP     R0,R1
2598  011636  001401                          BEQ     .+4
2599  011640  104002                          ERROR   2               ;CHARACTERS SHOULD COMPARE
2600  011642  052777  000010  170056          BIS     #HDXEN,@TXCSR   ;SET HALF DUPLEX HDX
2601  011650  012767  000010  167244          MOV     #8.,SHIFT       ;# OF SHIFT
2602  011656                          4$:
2603  011656  052777  020000  170042          BIS     #CLK,@TXCSR     ;POKE CLK UP
2604                                          ;WAIT FOR CABLE & DRIVER DELAYS
2605  011664  016702  167230                  MOV     HOLD,R2 ;WAIT THIS AMT
2606  011670  005302                          DEC     R2      ;WAIT
2607  011672  001376                          BNE     .-2
2608                                          ;EXIT...
2609  011674  042777  020000  170024          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2610                                          ;WAIT FOR CABLE & DRIVER DELAYS
2611  011702  016702  167212                  MOV     HOLD,R2 ;WAIT THIS AMT
2612  011706  005302                          DEC     R2      ;WAIT
2613  011710  001376                          BNE     .-2
2614                                          ;EXIT...
2615  011712  005367  167204                  DEC     SHIFT
2616  011716  001357                          BNE     4$
2617  011720  105777  167766                  TSTB    @RXCSR          ;RXDONE=0?
2618  011724  100001                          BPL     .+4
2619  011726  104004                          ERROR   4                       ;RXDONE SHOULD NOT BE ASSERTED
2620                                                                  ;CHECK OUT HDX LOGIC
2621  011730  017701  167762                  MOV     @RXDBUF,R1      ;ACTUAL
2622  011734  020001                          CMP     R0,R1
2623  011736  001401                          BEQ     .+4
2624  011740  104002                          ERROR   2               ;CHARACTERS SHOULD COMPARE
2625                                          ;NOTE THAT CHARACTER 25 WILL BE FROZEN
2626                                          ;IN THE RXDBUF EVEN THOUGH CHARACTER 24 WAS
2627                                          ;SENT TO THE RECEIVER
2628
2629  011742                          1$:
2630
```

INITIALIZE THE COMMON TAGS

```
2631
2632                                    ;;THIS TEST VERIFYS THAT BREAK FORCES A SPACE CONDITION
2633                                    ;;ON THE LINE WHILE TRANSMITTING
2634                                    ;;THIS TEST USES BOTH THE RECEIVER AND TRANSMITTER LOGIC
2635                                    ;;MODE:  SYNC EXT (SYNEXT)
2636                                    ;;LENGTH:  EIGHT
2637                                    ;;
2638
2639                                    ;;*****************************************************************
2640  011742  000004                    TST26:  SCOPE
2641  011744  052777  000400  167754            BIS     #MRESET,@TXCSR  ;MASTER RESET
2642  011752  012777  020000  167742            MOV     #SYNEXT,@PARCSR ;SET THE MODE
2643  011760  052777  000400  167740            BIS     #MRESET,@TXCSR  ;MASTER RESET
2644
2645                                    ;SET MAINTENANCE MODE & SEND
2646                                            ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2647  011766  012777  004020  167732            MOV     #MINT!SEND,@TXCSR
2648
2649                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2650  011774  012777  026026  167720            MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
2651  012002  052777  000020  167702            BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2652  012010  016703  167702                    MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
2653  012014  012767  000002  167102            MOV     #2,COUNT        ;# OF TIMES
2654  012022  112777  000025  167702            MOVB    #25,@TXDBUF     ;ANY CHARACTER
2655                                            ;POKE CLK FOR SYNCRONIZATION
2656  012030  052777  020000  167670            BIS     #CLK,@TXCSR     ;POKE CLK UP
2657  012036  042777  020000  167662            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2658  012044  012700  000025                    MOV     #25,R0          ;EXPECTED
2659  012050  012767  000010  167044  1$:       MOV     #8.,SHIFT       ;# OF SHIFTS
2660
2661  012056                            2$:
2662  012056  052777  020000  167642            BIS     #CLK,@TXCSR     ;POKE CLK UP
2663  012064  042777  020000  167634            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2664  012072  005367  167024                    DEC     SHIFT
2665  012076  022767  000003  167016            CMP     #3,SHIFT
2666  012104  001003                            BNE     3$
2667  012106  112777  000024  167616            MOVB    #24,@TXDBUF     ;LOAD NEXT CHAR
2668  012114  005767  167002          3$:       TST     SHIFT
2669  012120  001356                            BNE     2$
2670  012122  105777  167564                    TSTB    @RXCSR          ;RXDONE=1?
2671  012126  100401                            BMI     .+4
2672  012130  104004                            ERROR   4
2673  012132  017701  167560                    MOV     @RXDBUF,R1      ;ACTUAL
2674  012136  020001                            CMP     R0,R1
2675  012140  001401                            BEQ     .+4
2676  012142  104003                            ERROR   3
2677  012144  052777  000001  167554            BIS     #BREAK,@TXCSR   ;SET BREAK
2678  012152  012700  000000                    MOV     #0,R0           ;EXPECTED
2679  012156  005367  166742                    DEC     COUNT
2680  012162  001332                            BNE     1$
2681
2682
2683                                    ;;THIS TEST VERIFYS THAT DSC CAUSES AN INTERRUPT
2684                                    ;;THIS TEST ONLY WORKS IN MAINT EXTERNAL MODE
2685                                    ;;INTERRUPT VECTOR:  DURIV
2686                                    ;;*****************************************************************
```

```
                                        INITIALIZE THE COMMON TAGS

 2687  012164  000004                   TST27:  SCOPE
 2688  012166  105767  166761                   TSTB    JMRBY           ;IN MAINT EXTERNAL?
 2689  012172  100073                           BPL     3$              ;IF ANSWER NO JUMP AROUND TEST
 2690  012174  052777  000400  167524           BIS     #MRESET,@TXCSR  ;MASTER RESET
 2691  012202  105767  166743                   TSTB    OPTCLR  ;IS THE OPTIONAL CLR JUMPER IN ?
 2692  012206  100405                           BMI     5$      ;YES
 2693  012210  012777  000000  167474           MOV     #0,@RXCSR       ;CLR THE UNRESETTABLE BITS
 2694  012216  005777  167470                   TST     @RXCSR  ;GET RID OF DSC BY READING RXCSR
 2695  012222  012777  012244  167506  5$:      MOV     #4$,@DURIV      ;SET UP TRAPCATCHER
 2696  012230  016777  004552  167502           MOV     DUPRT,@DURIS    ;
 2697  012236  106427  000000                   MTPS    #0      ;ALLOW INTERRUPTS
 2698  012242  000422                           BR      1$      ;JUMP AROUND INTERRUPT SVC ROUTINE
 2699                                  ;THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
 2700  012244  106427  000340           4$:     MTPS    #340    ;DON'T ALLOW ANYMORE INTERRUPTS
 2701  012250  005777  167436                   TST     @RXCSR          ;DSC=1?
 2702  012254  100401                           BMI     .+4
 2703  012256  104004                           ERROR   4                       ;FALSE INTERRUPT
 2704  012260  042777  000040  167424           BIC     #DSINTE,@RXCSR  ;CLEAR INTERRUPT ENABLE
 2705  012266  012716  012356                   MOV     #2$,(SP)        ;SET UP RETURN LOCATION
 2706  012272  016777  167442  167436           MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
 2707  012300  012777  000000  167432           MOV     #0,@DURIS       ;
 2708  012306  000002                           RTI
 2709
 2710  012310  052777  000040  167374  1$:      BIS     #DSINTE,@RXCSR  ;SET INTERRUPT ENABLE
 2711  012316  052777  000002  167366           BIS     #DTR,@RXCSR     ;TRY TO CAUSE INTERRUPT
 2712  012324  005000                           CLR     R0
 2713  012326  005200                           INC     R0              ;WAIT FOR INTERRUPT
 2714  012330  001376                           BNE     .-2
 2715  012332  016777  167402  167376           MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
 2716  012340  012777  000000  167372           MOV     #0,@DURIS       ;
 2717
 2718  012346  042777  000040  167336           BIC     #DSINTE,@RXCSR  ;CLEAR INTERRUPT ENABLE
 2719  012354  104004                           ERROR   4                       ;INTERRUPT FAILED TO OCCUR
 2720  012356  106427  000340           2$:     MTPS    #340
 2721  012362                          3$:
 2722
 2723
 2724                                  ;END OF PASS
 2725                                  ;TYPE NAME OF TEST
 2726                                  ;UPDATE PASS COUNT
 2727                                  ;CHECK FOR EXIT TO ACT-11
 2728                                  ;RESTART TEST
 2729
 2730  012362  000004           .EOP:   SCOPE
 2731  012364  004767  000344                   JSR     PC,CKSWR
 2732  012370  104401                           TYPE                            ;TYPE NAME OF TEST
 2733  012372  015524                           MEPASS
 2734  012374  104413  012626                   CONVRT  ,OUTCRY
 2735  012400  104401  015343                   TYPE    ,DEVICE
 2736  012404  105767  166542                   TSTB    MULTD   ;ARE YOU RUNNING MULTIPLE DEVICES ?
 2737  012410  001511                           BEQ     CCC     ;NO,JUMP AROUND
 2738  012412  005767  166550                   TST     ACTREG  ;ARE ANY DEVICES ACTIVE ?
 2739  012416  001007                           BNE     RUNIT   ;YES
 2740  012420  104401  015355                   TYPE    ,MCOW   ;NO
 2741  012424  016700  166536                   MOV     ACTREG,R0       ;DISPLAY ACTREG
 2742  012430  000000                           HALT            ;SELECT SOMETHING TO RUN @ ACTREG:
```

```
2743                                                    ;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
2744  012432  000167  167514                            JMP     .START  ;START OVER AGAIN.....YOU DESELECTED EVERYTHING
2745  012436  062767  000010  166510  RUNIT:  ADD     #10,BASEADD     ;NEXT BLOCK (ADDRESSES)
2746  012444  062767  000010  166510  ZERO:   ADD     #10,BASEIV      ;NEXT BLOCK (VECTORS)
2747  012452  000241                            CLC
2748  012454  006167  166510                    ROL     ROTADD  ;UP DATE ROTATING POINTER
2749  012460  103410                            BCS     2$      ;IS IT THE LAST DEVICE
2750                                                            ;TO BE TESTED IN THIS PASS ?
2751  012462  036767  166502  166476            BIT     ROTADD,ACTREG   ;TEST THIS DEVICE FOR ACTIVE STATUS
2752  012470  001762                            BEQ     RUNIT   ;IF NOT ACTIVE, TRY NEXT ADDRESS
2753  012472  004767  000034                    JSR     PC,REPLAY       ;CALCULATE NEW PARAMETERS
2754  012476  000167  000210                    JMP     RESTRT  ;YES IT WAS ACTIVE,TEST THIS DEVICE
2755  012502  012767  000001  166460  2$:     MOV     #1,ROTADD       ;OK!,NOW SET UP  ROTATING
2756                                                            ;POINTER FOR NEXT MULTIPLE PASS
2757  012510  016767  166442  166436            MOV     KEEPADD,BASEADD ;RESTORE BASE ADDRESS
2758  012516  016767  166442  166436            MOV     KEEPIV,BASEIV   ;RESTORE BASE INTERRUPT VECTORS
2759  012524  004767  000002                    JSR     PC,REPLAY       ;CALC NEW PARAMETERS
2760  012530  000441                            BR      CCC     ;JUMP AROUND REPLAY
2761  012532  016767  166416  004404  REPLAY: MOV     BASEADD,DUBASE  ;SET UP FOR NEW ADDRESSES
2762  012540  004767  004246                    JSR     PC,DUADDR       ;CREATE NEW ADDRESSES
2763  012544  016767  166412  167164            MOV     BASEIV,DURIV    ;CREATE DURIV
2764  012552  062767  000002  166402            ADD     #2,BASEIV
2765  012560  016767  166376  167152            MOV     BASEIV,DURIS    ;CREATE DURIS
2766  012566  062767  000002  166366            ADD     #2,BASEIV
2767  012574  016767  166362  167140            MOV     BASEIV,DUTIV    ;CREATE DUTIV
2768  012602  062767  000002  166352            ADD     #2,BASEIV
2769  012610  016767  166346  167126            MOV     BASEIV,DUTIS    ;CREATE DUTIS
2770  012616  016767  167114  166336            MOV     DURIV,BASEIV    ;RESTORE
2771  012624  000207                            RTS     PC
2772
2773  012626  000001                    OUTCRY: 1
2774  012630     006     002                    .BYTE   6,2
2775  012632  001712                            RXCSR
2776
2777  012634                            CCC:
2778  012634  005067  166542                    CLR     $TSTNM                  ;CLEAR TEST NUMBER
2779  012640  005067  166552                    CLR     $ERRPC                  ;CLEAR LAST ERROR PC
2780  012644  005067  166533                    CLR     $ERFLG                  ;CLEAR ERROR FLAG
2781  012650  005267  166236                    INC     PASCNT                  ;UPDATE PASS COUNT
2782  012654  016767  166232  166220            MOV     PASCNT,LIGHTS           ;DISPLAY PASS COUNT
2783  012662  016767  166224  166644            MOV     PASCNT,$PASS            ;PASS COUNT TO APT
2784  012670  013701  000042                    MOV     @#42,R1                 ;CHECK FOR ACT-11 OR DDP
2785  012674  001406                            BEQ     RESTRT          ;IF NO CONTINUE TESTING
2786  012676  000005                            RESET
2787  012700  000005                            RESET
2788  012702  004711                    $ENDAD: JSR     PC,(R1)
2789  012704  000240                            NOP
2790  012706  000240                            NOP
2791  012710  000240                            NOP
2792  012712  106427  000340            RESTRT: MTPS    #340    ;PREVENT INTERRUPTS (PRIO: 7)
2793  012716  004767  000012                    JSR     PC,CKSWR
2794  012722  012767  003364  166456            MOV     #TST1+2,$LPADR          ;SET LAST ADDRESS POINTER
2795  012730  000167  170426                    JMP     TST1
2796
2797                                            ;CHECK SWITCH REGISTER ROUTINE.
2798                                            ;CHECKS TO ALLOW FOR <↑G> TO ALLOW
```

INITIALIZE THE COMMON TAGS

```
2799                                                    ;THE CHANGING OF LOCATION 176
2800
2801  012734  005737  000042          CKSWR:  TST     @#42
2802  012740  001040                          BNE     OUT
2803  012742  022767  000176  166470          CMP     #SWREG,SWR      ;SOFTWARE SWR PRESENT?
2804  012750  001034                          BNE     OUT             ;NO--LEAVE
2805  012752  105777  166466                  TSTB    @STKS           ;CHECK TTY READY
2806  012756  100031                          BPL     OUT             ;NO--LEAVE
2807  012760  017767  166462  000422          MOV     @STKB,.MSG      ;GET CHARACTER
2808  012766  042767  177600  000414          BIC     #177600,.MSG    ;STRIP JUNK
2809  012774  122767  000007  000406          CMPB    #7,.MSG         ;IS IT <^G> ?
2810  013002  001017                          BNE     OUT             ;NO
2811  013004  104401  016131                  TYPE    ,MCNTG
2812  013010  005137  013050          CNTLU:  COM     @#RDSW
2813  013014  104401  016141                  TYPE    ,MMSWR
2814  013020  104413                          CONVRT
2815  013022  013052                          SWREGL
2816  013024  104406  016152                  INSTR,MMNEW
2817  013030  104410                          PARAM
2818  013032  000000                          0
2819  013034  177777                          177777
2820  013036  000176                          SWREG
2821  013040     000     001          .BYTE   0,1
2822  013042  005037  013050          OUT:    CLR     @#RDSW
2823  013046  000207                          RTS     PC
2824  013050  000000                  RDSW:   .WORD   0
2825  013052  000001                  SWREGL: 1
2826  013054     006     002                  .BYTE   6,2
2827  013056  000176                          SWREG
2828
2829  013060  000005                          5
2830
2831                                          ;CHECK FOR FREEZE ON CURRENT DATA
2832
2833  013062  004767  177646          .SCOP1: JSR     PC,CKSWR
2834  013066  032777  001000  166344          BIT     #SW09,@SWR
2835  013074  001402                          BEQ     1$
2836  013076  016716  166006                  MOV     LOCK,(SP)
2837  013102  000002                  1$:     RTI
2838                                  .SBTTL  TYPE ROUTINE
2839
2840                                  ;;**********************************************************************
2841                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
2842                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
2843                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
2844                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
2845                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
2846                                  ;*
2847                                  ;*CALL:
2848                                  ;*1) USING A TRAP INSTRUCTION
2849                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
2850                                  ;*OR
2851                                  ;*      TYPE
2852                                  ;*      MESADR
2853                                  ;*
2854
```

```
2855  013104  105767  166347          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
2856  013110  100002                          BPL     1$              ;;BR IF YES
2857  013112  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
2858  013114  000430                          BR      3$              ;;LEAVE
2859  013116  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
2860  013120  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
2861  013124  122767  000001  166414          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
2862  013132  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
2863  013134  132767  000100  166405          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
2864  013142  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
2865  013144  010067  000004'                 MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
2866  013150  004767  000006'                 JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
2867  013154  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
2868  013156  132767  000040  166363  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
2869  013164  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
2870  013166  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
2871  013170  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
2872  013172  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
2873  013174  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
2874  013176  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
2875  013202  000002                          RTI                     ;;RETURN
2876  013204  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
2877  013210  001430                          BEQ     8$
2878  013212  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
2879  013216  001006                          BNE     5$
2880  013220  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
2881  013222  104401                          TYPE                    ;;TYPE A CR AND LF
2882  013224  001523                          $CRLF
2883  013226  105067  000130                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
2884  013232  000755                          BR      2$              ;;GET NEXT CHARACTER
2885  013234  004767  000056          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
2886  013240  126726  166212          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
2887  013244  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
2888  013246  016746  166202                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
2889                                                                  ;;AND THE NULL CHAR.
2890  013252  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
2891  013256  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
2892  013260  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
2893  013264  105367  000072                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
2894  013270  000770                          BR      7$              ;;LOOP
2895
2896                                  ;HORIZONTAL TAB PROCESSOR
2897
2898  013272  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
2899  013276  004767  000014          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
2900  013302  132767  000007  000052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
2901  013310  001372                          BNE     9$              ;;TAB STOP
2902  013312  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
2903  013314  000724                          BR      2$              ;;GET NEXT CHARACTER
2904  013316  105777  166126          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
2905  013322  100375                          BPL     $TYPEC
2906  013324  116677  000002  166120          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
2907  013332  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
2908  013340  001003                          BNE     1$              ;;BRANCH IF NO
2909  013342  105067  000014                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
2910  013346  000406                          BR      $TYPEX          ;;EXIT
```

```
                                        TYPE ROUTINE

2911  013350  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
2912  013356  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
2913  013360  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
2914  013362  000000                  $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
2915  013364  000207                  $TYPEX: RTS     PC
2916
2917
2918                                          ;ASCII STRING INPUT ROUTINE
2919
2920  013366  017667  000000  000014  .INSTR: MOV     @(SP),.MSG      ;PICK UP MESSAGE
2921  013374  062716  000002                  ADD     #2,(SP)         ;JUMP AROUND MESSAGE FOR RTI
2922  013400  105767  166142                  TSTB    $ENV            ;APT CONTROL
2923  013404  001036                          BNE     INSTR2          ;YES NO TYPE
2924  013406  104401                  .INST1: TYPE
2925  013410  000000                  .MSG:   0
2926  013412  012704  016164                  MOV     #INBUF,R4       ;GET STARTING LOC OF INBUF
2927  013416  012703  000007                  MOV     #7,R3           ;MAX # OF CHARS
2928  013422  105777  166016          1$:     TSTB    @$TKS   ;TTY FLAG
2929  013426  100375                          BPL     1$
2930  013430  117714  166012                  MOVB    @$TKB,(R4)      ;TAKE CHAR
2931  013434  142714  000200                  BICB    #200,(R4)       ;STRIP
2932  013440  121427  000025                  CMPB    (R4),#25        ;IS IT <^G>
2933  013444  001760                          BEQ     .INST1
2934  013446  122427  000015                  CMPB    (R4)+,#15       ;CHECK FOR CR
2935  013452  001413                          BEQ     INSTR2
2936  013454  105777  165770          2$:     TSTB    @$TPS   ;TEST FLAG
2937  013460  100375                          BPL     2$
2938  013462  117777  165760  165762          MOVB    @$TKB,@$TPB     ;ECHO CHARACTER
2939  013470  005303                          DEC     R3      ;DID YOU TYPE TOO MANY CHARS ?
2940  013472  001353                          BNE     1$
2941  013474  104401                  .INSTE: TYPE
2942  013476  015451                          MQM     ;?
2943  013500  000742                          BR      .INST1  ;RETRY
2944  013502  000002                  INSTR2: RTI
2945
2946                                          ;CONVERT ASCII STRING TO OCTAL
2947
2948  013504  011605                  .PARAM: MOV     (SP),R5 ;PUT CONTENTS OF SP INTO R5
2949  013506  012567  000162                  MOV     (R5)+,LOLIM     ;PUT LOW LIMIT INTO LOLIM
2950  013512  012567  000160                  MOV     (R5)+,HILIM     ;PUT HIGH LIMIT INTO HILIM
2951  013516  012567  000156                  MOV     (R5)+,DEVADR    ;PUT STORE LOC INTO DEVADR
2952  013522  112567  000154                  MOVB    (R5)+,LOBITS    ;PUT MASK INTO LOBITS
2953  013526  112567  000151                  MOVB    (R5)+,ADRCNT    ;PUT COUNT INTO ADRCNT
2954  013532  010516                          MOV     R5,(SP) ;RESTORE RETURN ADDR ON STACK FOR RTI
2955  013534  005005                  PARAM1: CLR     R5
2956  013536  012704  016164                  MOV     #INBUF,R4
2957  013542  122714  000015                  CMPB    #15,(R4)        ;CR ?
2958  013546  001420                          BEQ     PARERR  ;YOU TYPED CR TOO SOON !
2959  013550  121427  000060          1$:     CMPB    (R4),#60        ;LOW LIMIT ASCII 0
2960  013554  002415                          BLT     PARERR
2961  013556  121427  000067                  CMPB    (R4),#67        ;HIGH LIMIT ASCII 7
2962  013562  003012                          BGT     PARERR
2963  013564  142714  000060                  BICB    #60,(R4)        ;CONVERT TO OCTAL
2964  013570  152405                          BISB    (R4)+,R5        ;STORE AWAY ITS AN OK CHAR
2965  013572  122714  000015                  CMPB    #15,(R4)        ;CR ?
2966  013576  001414                          BEQ     LIMITS  ;NOW CHECK FOR HIGH &LOW LIMIT CONDS
```

```
2967  013600  006305                          ASL     R5      ;ALLOCATE ROOM FOR NEXT CHAR
2968  013602  006305                          ASL     R5
2969  013604  006305                          ASL     R5
2970  013606  000760                          BR      1$
2971  013610  122714  000015          PARERR: CMPB    #15,(R4)        ;CR?
2972  013614  001003                          BNE     120$
2973  013616  005737  013050                  TST     @#RDSW          ;CK SWR USED
2974  013622  001023                          BNE     PARTI
2975  013624  104407                  120$:   INSTER  ;RETRY
2976  013626  000742                          BR      PARAM1
2977
2978                                          ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2979
2980  013630  020567  000042          LIMITS: CMP     R5,HILIM
2981  013634  101365                          BHI     PARERR  ;THE # IS TOO HIGH
2982  013636  020567  000032                  CMP     R5,LOLIM
2983  013642  103762                          BLO     PARERR  ;THE # IS TOO LOW
2984  013644  136705  000032                  BITB    LOBITS,R5       ;TEST BY MASKINGTHE #
2985  013650  001357                          BNE     PARERR
2986
2987                                          ;STORE NUMBER AT SPECIFIED ADDRESS
2988
2989  013652  016704  000022                  MOV     DEVADR,R4       ;GET STARTING ADDR OF
2990  013656  010524                  1$:     MOV     R5,(R4)+        ;STORE AT THIS ADDR
2991  013660  062705  000002                  ADD     #2,R5
2992  013664  105367  000013                  DECB    ADRCNT  ;HOW MANY TIMES + 2 ?
2993  013670  001372                          BNE     1$
2994  013672  000002                  PARTI:  RTI
2995  013674  000000                  LOLIM:  0
2996  013676  000000                  HILIM:  0
2997  013700  000000                  DEVADR: 0
2998  013702  000000                  LOBITS: 0
2999          013703                  ADRCNT=LOBITS+1
3000
3001                                          ;SAVE PC OF TEST THAT FAILED AND R0-R5
3002
3003  013704  016667  000004  165214  .SAVOS: MOV     4(SP),SAVPC
3004
3005                                          ;SAVE R0-R5
3006
3007  013712  010567  165556          SVOS:   MOV     R5,$REG5
3008  013716  010467  165550                  MOV     R4,$REG4
3009  013722  010367  165542                  MOV     R3,$REG3
3010  013726  010267  165534                  MOV     R2,$REG2
3011  013732  010167  165526                  MOV     R1,$REG1
3012  013736  010067  165520                  MOV     R0,$REG0
3013  013742  000002                          RTI
3014
3015                                          ;RESTORE R0-R5
3016
3017  013744  016700  165512          .RESOS: MOV     $REG0,R0
3018  013750  016701  165510                  MOV     $REG1,R1
3019  013754  016702  165506                  MOV     $REG2,R2
3020  013760  016703  165504                  MOV     $REG3,R3
3021  013764  016704  165502                  MOV     $REG4,R4
3022  013770  016705  165500                  MOV     $REG5,R5
```

```
3023  013774  000002                          RTI
3024
3025                                          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
3026
3027  013776  104401                  .CONVR: TYPE
3028  014000  015455                          MCRLF   ;CR  LF
3029  014002  017601  000000                  MOV     @(SP),R1        ;PICK UP DATA POINTER
3030  014006  062716  000002                  ADD     #2,(SP) ;SET UP SP FOR RTI
3031  014012  012167  000130                  MOV     (R1)+,WRDCNT    ;PICK UP # OF WORDS FROM TABLE
3032  014016  112167  000126          1$:     MOVB    (R1)+,CHRCNT    ;PICK UP # OF CHARS FROM TABLE
3033  014022  112167  000123                  MOVB    (R1)+,SPACNT    ;PICK UP # OF SPACES FROM TABLE
3034  014026  013167  000120                  MOV     @(R1)+,BINWRD   ;PICK UP ADDRESS OF  MSG
3035                                                                  ;FROM TABLE
3036  014032  016704  000114          2$:     MOV     BINWRD,R4       ;SAVE
3037  014036  116705  000106                  MOVB    CHRCNT,R5       ;SAVE
3038  014042  012700  016226                  MOV     #TEMP,R0        ;STARTING ADDRESS OF TEMP BLOCK
3039  014046  010403                  3$:     MOV     R4,R3           ;SAVE
3040  014050  042703  177770                  BIC     #177770,R3      ;CLR OUT UPPER BITS .. SAVE CHAR
3041  014054  062703  000260                  ADD     #260,R3 ;CONVERT TO ASCII
3042  014060  110320                          MOVB    R3,(R0)+        ;STORE AWAY
3043  014062  006204                          ASR     R4      ;SHIFT FOR NEXT #
3044  014064  006204                          ASR     R4      ;DITTO
3045  014066  006204                          ASR     R4      ;DITTO
3046  014070  005305                          DEC     R5      ;DEC CHAR COUNT
3047  014072  001365                          BNE     3$      ;DO IT AGAIN ?
3048  014074  012703  016270                  MOV     #MDATA,R3       ;STARTING ADDRESS OF MDATA BLOCK
3049  014100  114023                  4$:     MOVB    -(R0),(R3)+     ;REVERSE THE ORDER OF NUMBERS
3050  014102  105367  000042                  DECB    CHRCNT  ;DEC CHAR COUNT
3051  014106  001374                          BNE     4$      ;DO IT AGAIN ?
3052  014110  105767  000035                  TSTB    SPACNT  ;HOW MANY SPACES ?
3053  014114  001405                          BEQ     6$      ;TYPE # IF BR =0
3054  014116  112723  000240          5$:     MOVB    #240,(R3)+      ;"SPACE" IN ASCII
3055  014122  105367  000023                  DECB    SPACNT  ;DEC # OF SPACE COUNT
3056  014126  001373                          BNE     5$      ;DO IT AGAIN ?
3057  014130  105013                  6$:     CLRB    (R3)    ;INSERT "0" FOR TTY OUTPUT ROUTINE
3058  014132  104401                          TYPE
3059  014134  016270                          MDATA   ;THIS MESSAGE
3060  014136  005367  000004                  DEC     WRDCNT  ;HOW MANY #'S ?
3061  014142  001325                          BNE     1$      ;DO THIS ROUTINE AGAIN IF NOT EQUAL TO 0
3062  014144  000002                          RTI     ;RETURN TO PROGRAM
3063  014146  000000                  WRDCNT: 0
3064  014150  000000                  CHRCNT: 0
3065           014151                  SPACNT=CHRCNT+1
3066  014152  000000                  BINWRD: 0
3067
3068                                          ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
3069                                          ;BUFFER TO THE CHARACTERS "N" AND "Y".
3070                                          ;IF THE CHARACTER IS "N" CLEAR THE FLAG
3071                                          ;IF THE CHARACTER IS "Y" SET THE FLAG
3072
3073  014154  017605  000000          .SETFLG:MOV     @(SP),R5
3074  014160  122767  000116  001776          CMPB    #'N,INBUF       ;IS IT "N" ?
3075  014166  001002                          BNE     1$
3076  014170  105015                          CLRB    (R5)    ;000
3077  014172  000406                          BR      2$
3078  014174  122767  000131  001762  1$:     CMPB    #'Y,INBUF       ;IS IT "Y" ?
```

```
3079  014202  001005                          BNE     3$
3080  014204  112715  177777                  MOVB    #-1,(R5)        ;377
3081  014210  062716  000002          2$:     ADD     #2,(SP)
3082  014214  000002                          RTI
3083  014216  104407                  3$:     INSTER  ;RETRY
3084  014220  000755                          BR      .SETFLG
3085                                  .SBTTL  ERROR HANDLER ROUTINE
3086
3087                                  ;;*********************************************************************
3088                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
3089                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
3090                                  ;*AND GO TO SAVIT ON ERROR
3091                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3092                                  ;*SW15=1         HALT ON ERROR
3093                                  ;*SW13=1         INHIBIT ERROR TYPEOUTS
3094                                  ;*SW10=1         BELL ON ERROR
3095                                  ;*SW09=1         LOOP ON ERROR
3096                                  ;*CALL
3097                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
3098
3099  014222                          $ERROR:
3100  014222  105267  165155          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
3101  014226  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
3102  014230  016777  165146  165204          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
3103  014236  032777  002000  165174          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
3104  014244  001402                          BEQ     1$              ;;NO - SKIP
3105  014246  104401  001516                  TYPE    ,$BELL          ;;RING BELL
3106  014252  005267  165134          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
3107  014256  011667  165134                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
3108  014262  162767  000002  165126          SUB     #2,$ERRPC
3109  014270  117767  165122  165116          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
3110  014276  032777  020000  165134          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
3111  014304  001004                          BNE     20$             ;;SKIP TYPEOUTS
3112  014306  004767  000072                  JSR     PC,SAVIT        ;;GO TO USER ERROR ROUTINE
3113  014312  104401  001523                  TYPE    ,$CRLF
3114  014316                          20$:
3115  014316  122767  000001  165222          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
3116  014324  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
3117  014326  116767  165062  000004          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
3118  014334  004767  000016'                 JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
3119  014340     000          21$:    .BYTE   0
3120  014341     000                  .BYTE   0
3121  014342  000777          22$:    BR      22$             ;;APT ERROR LOOP
3122  014344  005777  165070          2$:     TST     @SWR            ;;HALT ON ERROR
312_  014350  100001                  BPL     3$              ;;SKIP IF CONTINUE
3124  014352  000000                  HALT                    ;;HALT ON ERROR!
3125  014354  032777  001000  165056  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
3126  014362  001402                  BEQ     4$              ;;BR IF NO
3127  014364  016716  165020                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
3128  014370  005767  165120          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
3129  014374  001402                  BEQ     5$              ;;BR IF NONE
3130  014376  016716  165112                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
3131  014402                          5$:
3132  014402  000002                  RTI                     ;;RETURN
3133  014404  010067  164520          SAVIT:  MOV     R0,HLD0
3134  014410  010167  164516                  MOV     R1,HLD1
```

```
3135  014414  010267  164514                  MOV     R2,HLD2
3136  014420  010367  164512                  MOV     R3,HLD3
3137  014424  010467  164510                  MOV     R4,HLD4
3138  014430  010567  164506                  MOV     R5,HLD5
3139  014434  016767  164742  164502          MOV     $TSTNM,HLD6
3140
3141                                  .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
3142
3143                                  ;;**********************************************************************
3144                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
3145                                  ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
3146                                  ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
3147
3148  014442                          $ERRTYP:
3149  014442  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3150  014446  010046                          MOV     R0,-(SP)        ;;SAVE R0
3151  014450  005000                          CLR     R0              ;;PICKUP THE ITEM INDEX
3152  014452  153700  001414                  BISB    @#$ITEMB,R0
3153  014456  001004                          BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
3154                                                                  ;;TYPE THE PC OF THE ERROR
3155  014460  016746  164732                  MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
3156                                                                  ;;ERROR ADDRESS
3157  014464  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3158  014466  000426                          BR      6$              ;;GET OUT
3159  014470  005300                  1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
3160  014472  006300                          ASL     R0              ;;      WORK FOR THE ERROR TABLE
3161  014474  006300                          ASL     R0
3162  014476  006300                          ASL     R0
3163  014500  062700  001652                  ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
3164  014504  012067  000004                  MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
3165  014510  001404                          BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
3166  014512  104401                          TYPE                    ;;TYPE THE "ERROR MESSAGE"
3167  014514  000000                  2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
3168  014516  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3169  014522  012067  000004          3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
3170  014526  001404                          BEQ     5$              ;;SKIP TYPEOUT IF 0
3171  014530  104401                          TYPE                    ;;TYPE THE "DATA HEADER"
3172  014532  000000                  4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
3173  014534  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3174  014540  011000                  5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
3175  014542  001004                          BNE     7$              ;;GO TYPE THE DATA
3176  014544  012600                  6$:     MOV     (SP)+,R0        ;;RESTORE R0
3177  014546  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3178  014552  000207                          RTS     PC              ;;RETURN
3179  014554                          7$:
3180  014554  013046                          MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
3181  014556  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3182  014560  005710                          TST     (R0)            ;;IS THERE ANOTHER NUMBER?
3183  014562  001770                          BEQ     6$              ;;BR IF NO
3184  014564  104401  014572                  TYPE    ,8$             ;;TYPE TWO(2) SPACES
3185  014570  000771                          BR      7$              ;;LOOP
3186  014572  020040  000             8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
3187            014576                        .EVEN
3188                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3189
3190                                  ;;**********************************************************************
```

```
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;;M=1 OR 0
                                ;*                                      ;;1=TYPE LEADING ZEROS
                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                ;*$TYPOS OR $TYPOC
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPON                   ;;CALL FOR TYPEOUT
                                ;*
                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOC                   ;;CALL FOR TYPEOUT

014576  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
014602  116667  000001  000211          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
014610  112667  000207                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
014614  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
014620  000406                          BR      $TYPON
014622  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
014630  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
014636  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
014644  010346                          MOV     R3,-(SP)        ;;SAVE R3
014646  010446                          MOV     R4,-(SP)        ;;SAVE R4
014650  010546                          MOV     R5,-(SP)        ;;SAVE R5
014652  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
014656  005404                          NEG     R4
014660  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
014664  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
014670  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
014674  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
014700  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
014702  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
014704  000404                          BR      3$              ;;GO DO MSB
014706  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
014710  006105                          ROL     R5
014712  006105                          ROL     R5
014714  010503                          MOV     R5,R3
014716  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
014720  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
014724  100016                          BPL     7$              ;;BR IF NO
014726  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
014732  001002                          BNE     4$              ;;TEST FOR 0
014734  005704                          TST     R4              ;;SUPPRESS THIS 0?
014736  001403                          BEQ     5$              ;;BR IF YES
014740  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
014742  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
014746  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
```

BINARY TO OCTAL (ASCII) AND TYPE

```
3247  014752  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
3248  014756  104401  015016                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
3249  014762  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
3250  014766  003347                          BGT     2$              ;;BR IF MORE TO DO
3251  014770  002402                          BLT     6$              ;;BR IF DONE
3252  014772  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
3253  014774  000744                          BR      2$              ;;GO DO THE LAST DIGIT
3254  014776  012605          6$:     MOV     (SP)+,R5        ;;RESTORE R5
3255  015000  012604                  MOV     (SP)+,R4        ;;RESTORE R4
3256  015002  012603                  MOV     (SP)+,R3        ;;RESTORE R3
3257  015004  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
3258  015012  012616                  MOV     (SP)+,(SP)
3259  015014  000002                  RTI                     ;;RETURN
3260  015016  000            8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
3261  015017  000                    .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
3262  015020  000            $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
3263  015021  000            $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
3264  015022  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
3265                                  ;ENTER HERE ON POWER FAILURE
3266
3267
3268  015024                  $PWRDN:
3269  015024  010046          .PFAIL: MOV     R0,-(SP)                ;SAVE R0-R5 ON PROCESSOR STACK
3270  015026  010146                  MOV     R1,-(SP)
3271  015030  010246                  MOV     R2,-(SP)
3272  015032  010346                  MOV     R3,-(SP)
3273  015034  010446                  MOV     R4,-(SP)
3274  015036  010546                  MOV     R5,-(SP)
3275  015040  016746  162760          MOV     24,-(SP)
3276  015044  010667  164046          MOV     SP,SAVSP                ;SAVE STACK POINTER
3277  015050  012767  015062  162746  MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
3278  015056  000000                  HALT                            ;HALT ON POWER DOWN NORMAL
3279  015060  000777                  BR      .
3280
3281                                  ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3282
3283  015062  016706  164030  RESTAR: MOV     SAVSP,SP                ;RESTORE STACK POINTER
3284  015066  012605                  MOV     (SP)+,R5                ;RESTORE R0-R5
3285  015070  012604                  MOV     (SP)+,R4
3286  015072  012603                  MOV     (SP)+,R3
3287  015074  012602                  MOV     (SP)+,R2
3288  015076  012601                  MOV     (SP)+,R1
3289  015100  012600                  MOV     (SP)+,R0
3290  015102  012767  015024  162714  MOV     #.PFAIL,24              ;SET UP FOR POWER FAILURE
3291  015110  106427  000340          MTPS    #340
3292  015114  012706  001100          MOV     #STACK,SP
3293  015120  005067  001102          CLR     TEMP
3294  015124  005267  001076          INC     TEMP
3295  015130  001375                  BNE     .-4
3296  015132  104413                  CONVRT
3297  015134  015156                  PFTAB
3298  015136  104401                  TYPE
3299  015140  015460                  MPFAIL
3300  015142  005067  164235          CLR     $ERFLG
3301  015146  005067  164244          CLR     $ERRPC
3302  015152  000177  163726          JMP     @RETURN
```

BINARY TO OCTAL (ASCII) AND TYPE

```
3303  015156  000001                    PFTAB:  1
3304  015160     006     002                    .BYTE   6,2
3305  015162  000207                            RETURN
3306  015164  005015  042012  053125  MTITLE: .ASCIZ  <15><12><12>/DUV11 DZDUU-A TAPE E /<15><12>
3307  015172  030461  042040  042132
3308  015200  052525  040455  052040
3309  015206  050101  020105  020105
3310  015214  005015     000
3311  015217     015  053012  041505  MVECTO: .ASCIZ  <15><12>/VEC ADD-/
3312  015224  040440  042104  000055
3313  015232  005015  051461  020124  MREGAD: .ASCIZ  <15><12>/1ST DEV: REC CSR ADD-/
3314  015240  042504  035126  051040
3315  015246  041505  041440  051123
3316  015254  040440  042104  000055
3317  015262  005015  052515  052114  MMULT:  .ASCIZ  <15><12>/MULT DEV ? (Y OR N)-/
3318  015270  042040  053105  037440
3319  015276  024040  020131  051117
3320  015304  047040  026451     000
3321  015311     015  046012  051501  MLASTD: .ASCIZ  <15><12>/LAST DEV: REC CSR ADDR-/
3322  015316  020124  042504  035126
3323  015324  051040  041505  041440
3324  015332  051123  040440  042104
3325  015340  026522     000
3326  015343     075  042504  044526  DEVICE: .ASCIZ  /=DEVICE  /
3327  015350  042503  020040     000
3328  015355     015  051412  046105  MCOW:   .ASCIZ  <15><12>/SELECT TO RUN @ACTREG/
3329  015362  041505  020124  047524
3330  015370  051040  047125  040040
3331  015376  041501  051124  043505
3332  015404     000
3333  015405     015  047412  043126  MRANGE: .ASCIZ  <15><12>/OVFLO:RETYPE LAST DEV RXCSR ADDS-/
3334  015412  047514  051072  052105
3335  015420  050131  020105  040514
3336  015426  052123  042040  053105
3337  015434  051040  041530  051123
3338  015442  040440  042104  026523
3339  015450     000
3340  015451     040  037440     000  MQM     .ASCIZ  /  ?/
3341  015455     015  000012          MCR     .ASCIZ   <15><12>
3342  015460  043120  044501  026114  MPF     .ASCIZ  /PFAIL,  RESTART AT TEST IN PROGRESS/
3343  015466  020040  042522  052123
3344  015474  051101  020124  052101
3345  015502  052040  051505  020124
3346  015510  047111  050040  047522
3347  015516  051107  051505  000123
3348  015524  005015  047105  020104  MEPASS: .ASCIZ  <15><12>/END OF PASS TAPE E/
3349  015532  043117  050040  051501
3350  015540  020123  040524  042520
3351  015546  042440     000
3352  015551     015  051012     000  MR:     .ASCIZ  <15><12>/R/
3353  015555     015  052012  051505  MTSTPC: .ASCIZ  <15><12>/TEST PC-/
3354  015562  020124  041520  000055
3355  015570  005015  047514  045503  MLOCK:  .ASCIZ  <15><12>/LOCK ON  TEST? (Y OR N)-/
3356  015576  047440  020116  052040
3357  015604  051505  037524  024040
3358  015612  020131  051117  047040
```

```
3359  015620  026451     000
3360  015623     015  021412  047440  MSYNC:  .ASCIZ  <15><12>/# OF SYNC CHARS SELECTED ( 1 OR 2)-/
3361  015630  020106  054523  041516
3362  015636  041440  040510  051522
3363  015644  051440  046105  041505
3364  015652  042524  020104  020050
3365  015660  020061  051117  031040
3366  015666  026451     000
3367  015671     015  044412  020123  MWIRE6: .ASCIZ  <15><12>/IS SEC XMIT SWITCH E55-2 IN? (Y OR N)-/
3368  015676  042523  020103  046530
3369  015704  052111  051440  044527
3370  015712  041524  020110  032505
3371  015720  026465  020062  047111
3372  015726  020077  054450  047440
3373  015734  020122  024516  000055
3374  015742  005015  051511  051440  MWIRE5: .ASCIZ  <15><12>/IS SEC REC SWITCH E55-3 IN? (Y OR N)-/
3375  015750  041505  051040  041505
3376  015756  051440  044527  041524
3377  015764  020110  032505  026465
3378  015772  020063  047111  020077
3379  016000  054450  047440  020122
3380  016006  024516  000055
3381  016012  005015  051511  047440  MWIRE4: .ASCIZ  <15><12>/IS OPT CLR ENABLE SWITCH E55-1 IN? (Y OR N)-/
3382  016020  052120  041440  051114
3383  016026  042440  040516  046102
3384  016034  020105  053523  052111
3385  016042  044103  042440  032465
3386  016050  030455  044440  037516
3387  016056  024040  020131  051117
3388  016064  047040  026451     000
3389  016071     015  005012  031510  MEXTJ:  .ASCIZ  <15><12><12>/H315 CONNECTOR ON ?(Y OR N)-/
3390  016076  032461  041440  047117
3391  016104  042516  052103  051117
3392  016112  047440  020116  024077
3393  016120  020131  051117  047040
3394  016126  026451     000
3395  016131     015  020012  043536  MCNTG:  .ASCIZ  <15><12>/ ↑G  /
3396  016136  020040     000
3397  016141     040  053523  036522  MMSWR:  .ASCIZ  / SWR=   /
3398  016146  020040  000040
3399  016152  020040  047040  053505  MMNEW:  .ASCIZ  /   NEW=  /
3400  016160  020075  000040
3401                                  .EVEN
3402
3403                                          ;BUFFERS FOR INPUT-OUTPUT
3404
3405  016164  000000                  INBUF:  0
3406          016226                  .=.+40
3407  016226  000000                  TEMP:   0
3408          016270                  .=.+40
3409  016270  000000                  MDATA:  0
3410          016332                  .=.+40
3411                                  .SBTTL  SCOPE HANDLER ROUTINE
3412
3413                                  ;;**************************************************************
3414                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
```

SCOPE HANDLER ROUTINE

```
3415                                    ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
3416                                    ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
3417                                    ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3418                                    ;*SW14=1        LOOP ON TEST
3419                                    ;*SW11=1        INHIBIT ITERATIONS
3420                                    ;*SW09=1        LOOP ON ERROR
3421                                    ;*SW08=1        LOOP ON TEST IN SWR<7:0>
3422                                    ;*CALL
3423                                    ;*      SCOPE           ;;SCOPE=IOT
3424
3425  016332                            $SCOPE:
3426
3427                                            ;SCOPE LOOP AND INTERATION HANDLER
3428
3429  016332                            .SCOPE:
3430  016332  004767  174376                    JSR     PC,CKSWR
3431  016336  005067  163054                    CLR     $ERRPC          ;CLEAR LAST ERROR PC
3432  016342  022716  003364                    CMP     #TST1+2,(SP)    ;IS SCOPE AT BEGINING OF TEST 1?
3433  016346  001413                            BEQ     $XTSTR          ;YES NO LOOP.
3434
3435  016350  000406                    TTST:   BR      1$              ;GO TO 1$ (IF LOCK SW02=1)
3436  016352  105777  163066                    TSTB    @$TKS           ;KEYBOARD DONE?
3437  016356  100123                            BPL     $OVER           ;BR IF NO
3438  016360  017766  163062  177776            MOV     @$TKB,-2(SP)    ;CLEAR DONE BIT
3439  016366  032777  040000  163044  1$:       BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
3440  016374  001114                            BNE     $OVER           ;;YES IF SW14=1
3441                                    ;#####START OF CODE FOR THE XOR TESTER#####
3442  016376  000416                    $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
3443                                                                    ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
3444  016400  013746  000004                    MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
3445  016404  012737  016424  000004            MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
3446  016412  005737  177060                    TST     @#177060        ;;TIME OUT ON XOR?
3447  016416  012637  000004                    MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
3448  016422  000463                            BR      $SVLAD          ;;GO TO THE NEXT TEST
3449  016424  022626                    5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
3450  016426  012637  000004                    MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
3451  016432  000423                            BR      7$              ;;LOOP ON THE PRESENT TEST
3452  016434                            6$:;#####END OF CODE FOR THE XOR TESTER#####
3453  016434  032777  000400  162776            BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
3454  016442  001404                            BEQ     2$              ;;BR IF NO
3455  016444  127767  162770  162730            CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
3456  016452  001465                            BEQ     $OVER           ;;BR IF YES
3457  016454  105767  162723            2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
3458  016460  001421                            BEQ     3$              ;;BR IF NO
3459  016462  126767  162727  162713            CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
3460  016470  101015                            BHI     3$              ;;BR IF NO
3461  016472  032777  001000  162740            BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
3462  016500  001404                            BEQ     4$              ;;BR IF NO
3463  016502  016767  162702  162676  7$:       MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
3464  016510  000446                            BR      $OVER
3465  016512  105067  162665            4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
3466  016516  005067  162770                    CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
3467  016522  000415                            BR      1$              ;;ESCAPE TO THE NEXT TEST
3468  016524  032777  004000  162706  3$:       BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
3469  016532  001011                            BNE     1$              ;;BR IF YES
3470  016534  005767  162774                    TST     $PASS           ;;IF FIRST PASS OF PROGRAM
```

```
3471  016540  001406                          BEQ     1$                ;;       INHIBIT ITERATIONS
3472  016542  005267  162636                  INC     $ICNT             ;;INCREMENT ITERATION COUNT
3473  016546  026767  162740  162630          CMP     $TIMES,$ICNT      ;;CHECK THE NUMBER OF ITERATIONS MADE
3474  016554  002024                          BGE     $OVER             ;;BR IF MORE ITERATION REQUIRED
3475  016556  012767  000001  162620  1$:     MOV     #1,$ICNT          ;;REINITIALIZE THE ITERATION COUNTER
3476  016564  016767  000056  162720          MOV     $MXCNT,$TIMES     ;;SET NUMBER OF ITERATIONS TO DO
3477  016572  105267  162604          $SVLAD: INCB    $TSTNM            ;;COUNT TEST NUMBERS
3478  016576  116767  162600  162726          MOVB    $TSTNM,$TESTN     ;;SET TEST NUMBER IN APT MAILBOX
3479  016604  011667  162576                  MOV     (SP),$LPADR       ;;SAVE SCOPE LOOP ADDRESS
3480  016610  011667  162574                  MOV     (SP),$LPERR       ;;SAVE ERROR LOOP ADDRESS
3481  016614  005067  162674                  CLR     $ESCAPE           ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
3482  016620  112767  000001  162567          MOVB    #1,$ERMAX         ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
3483  016626  016777  162550  162606  $OVER:  MOV     $TSTNM,@DISPLAY   ;;DISPLAY TEST NUMBER
3484  016634  016716  162546                  MOV     $LPADR,(SP)       ;;FUDGE RETURN ADDRESS
3485  016640  000002                  4$:     RTI
3486  016642  001407                  BRW:    1407
3487  016644  000432                  BRX:    432
3488  016646  000005                  $MXCNT: 5                         ;;MAX. NUMBER OF ITERATIONS
3489                                   .SBTTL  TRAP DECODER
3490
3491                                   ;;**************************************************************
3492                                   ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3493                                   ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3494                                   ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
3495                                   ;*GO TO THAT ROUTINE.
3496
3497  016650  010046                  $TRAP:  MOV     R0,-(SP)          ;;SAVE R0
3498  016652  016600  000002                  MOV     2(SP),R0          ;;GET TRAP ADDRESS
3499  016656  005740                          TST     -(R0)             ;;BACKUP BY 2
3500  016660  111000                          MOVB    (R0),R0           ;;GET RIGHT BYTE OF TRAP
3501  016662  006300                          ASL     R0                ;;POSITION FOR INDEXING
3502  016664  016000  016704                  MOV     $TRPAD(R0),R0     ;;INDEX TO TABLE
3503  016670  000200                          RTS     R0                ;;GO TO ROUTINE
3504
3505
3506                                   ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
3507
3508  016672  011646                  $TRAP2: MOV     (SP),-(SP)        ;;MOVE THE PC DOWN
3509  016674  016666  000004  000002          MOV     4(SP),2(SP)       ;;MOVE THE PSW DOWN
3510  016702  000002                          RTI                       ;;RESTORE THE PSW
3511
3512                                   .SBTTL  TRAP TABLE
3513
3514                                   ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
3515                                   ;*BY THE "TRAP" INSTRUCTION.
3516
3517                                   ;       ROUTINE
3518                                   ;       -------
3519  016704  016672                  $TRPAD: .WORD   $TRAP2
3520  016706  013104                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
3521  016710  014622                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
3522  016712  014576                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
3523  016714  014636                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
3524
3525
3526  016716  013062                          .SCOP1  ;;CALL=SCOP1    TRAP+5(104405)
```

```
3527  016720  013366                          .INSTR  ;;CALL=INSTR    TRAP+6(104406)
3528  016722  013474                          .INSTER ;;CALL=INSTER   TRAP+7(104407)
3529  016724  013504                          .PARAM  ;;CALL=PARAM    TRAP+10(104410)
3530  016726  013704                          .SAVOS  ;;CALL=SAVOS    TRAP+11(104411)
3531  016730  013744                          .RESOS  ;;CALL=RESOS    TRAP+12(104412)
3532  016732  013776                          .CONVRT ;;CALL=CONVRT   TRAP+13(104413)
3533  016734  014154                          .SETFLG ;;CALL=SETFLG   TRAP+14(104414)
3534                                  ;*****************************************************************
3535                                  ;UTILITIES
3536                                  ;*****************************************************************
3537
3538                                  ;THIS UTILITY CALCULATES PRIORITY LEVEL
3539  016736  006367  000044          DULEV:  ASL     DUPRT   ;SHIFT LEFT
3540  016742  006367  000040                  ASL     DUPRT   ;
3541  016746  006367  000034                  ASL     DUPRT   ;
3542  016752  006367  000030                  ASL     DUPRT   ;
3543  016756  006367  000024                  ASL     DUPRT   ;
3544  016762  016767  000020  000020         MOV     DUPRT,LESS1     ;MOVE THIS TO LESS1
3545  016770  162767  000001  000012         SUB     #1,LESS1        ;CREATE LESS1
3546  016776  042767  000037  000004         BIC     #37,LESS1       ;CLEAR TNZVC
3547  017004  000207                          RTS     PC
3548  017006  000240                  DUPRT:  PR5
3549  017010  000200                  LESS1:  PR4     ;LEVEL TO ALLOW INTERRUPTS
3550
3551                                  ;NEW DU ADDRESSES
3552  017012  016767  000126  162672  DUADDR: MOV     DUBASE,RXCSR    ;XXX0
3553  017020  005267  000120                  INC     DUBASE
3554  017024  016767  000114  162662         MOV     DUBASE,HRXCSR   ;XXX1
3555  017032  005267  000106                  INC     DUBASE
3556  017036  016767  000102  162652         MOV     DUBASE,RXDBUF   ;XXX2
3557  017044  016767  000074  162650         MOV     DUBASE,PARCSR   ;XXX2
3558  017052  005267  000066                  INC     DUBASE
3559  017056  016767  000062  162634         MOV     DUBASE,HRXDBUF  ;XXX3
3560  017064  016767  000054  162632         MOV     DUBASE,HPARCSR  ;XXX3
3561  017072  005267  000046                  INC     DUBASE
3562  017076  016767  000042  162622         MOV     DUBASE,TXCSR    ;XXX4
3563  017104  005267  000034                  INC     DUBASE
3564  017110  016767  000030  162612         MOV     DUBASE,HTXCSR   ;XXX5
3565  017116  005267  000022                  INC     DUBASE
3566  017122  016767  000016  162602         MOV     DUBASE,TXDBUF   ;XXX6
3567  017130  005267  000010                  INC     DUBASE
3568  017134  016767  000004  162572         MOV     DUBASE,HTXDBUF  ;XXX7
3569  017142  000207                          RTS     PC
3570  017144  000000                  DUBASE: 0
3571
3572                                  ;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3573                                  ;INFORMATION CONTAINED IN $TMP1 AND IT IS
3574                                  ;SHIFTED IN BY THE CONTENTS OF SHIFT
3575  017146  042777  040000  162552  RPOKE:  BIC     #MTDATA,@TXCSR
3576  017154  005067  162322                  CLR     $TMP2
3577  017160  006067  162314                  ROR     $TMP1   ;FORCE CARRY
3578  017164  006067  162312                  ROR     $TMP2   ;PICK UP CARRY IN BIT 15
3579  017170  006267  162306                  ASR     $TMP2   ;SHIFT INTO BIT 14
3580  017174  042767  100000  162300         BIC     #BIT15,$TMP2    ;CLR BIT 15
3581  017202  056777  162274  162516         BIS     $TMP2,@TXCSR    ;POKE MAINT DATA
3582  017210  042777  020000  162510         BIC     #CLK,@TXCSR     ;POKE CLK
```

```
3583  017216  052777  020000  162502          BIS     #CLK,@TXCSR     ;
3584  017224  005367  161672                  DEC     SHIFT
3585  017230  001346                          BNE     RPOKE
3586  017232  000207                          RTS     PC
3587
3588                                          ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3589  017234  016767  162240  162240  ODD8:   MOV     $TMP1,$TMP2     ;SAVE TEMP1
3590  017242  005067  162236                  CLR     $TMP3
3591  017246  012727  000010                  MOV     #8.,(PC)+
3592  017252  000000                  4$:     0
3593  017254  006067  162222          1$:     ROR     $TMP2
3594  017260  005567  162220                  ADC     $TMP3
3595  017264  005367  177762                  DEC     4$
3596  017270  001371                          BNE     1$
3597  017272  006067  162206                  ROR     $TMP3
3598  017276  103404                          BCS     2$
3599  017300  052767  000400  162172          BIS     #BIT8,$TMP1     ;SET ODD PARITY
3600  017306  000403                          BR      3$
3601  017310  042767  000400  162162  2$:     BIC     #BIT8,$TMP1     ;CLR EVEN PARITY
3602                                          ;$TMP1 NOW HAS ODD PARITY CHARACTER
3603  017316  000207                  3$:     RTS     PC
3604
3605                                          ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3606  017320  016767  162154  162154  EVEN8:  MOV     $TMP1,$TMP2     ;SAVE TEMP1
3607  017326  005067  162152                  CLR     $TMP3
3608  017332  012727  000010                  MOV     #8.,(PC)+
3609  017336  000000                  4$:     0
3610  017340  006067  162136          1$:     ROR     $TMP2
3611  017344  005567  162134                  ADC     $TMP3
3612  017350  005367  177762                  DEC     4$
3613  017354  001371                          BNE     1$
3614  017356  006067  162122                  ROR     $TMP3
3615  017362  103004                          BCC     2$
3616  017364  052767  000400  162106          BIS     #BIT8,$TMP1     ;SET EVEN PARITY
3617  017372  000403                          BR      3$
3618  017374  042767  000400  162076  2$:     BIC     #BIT8,$TMP1     ;CLR ODD PARITY
3619                                          ;$TMP1 NOW HAS EVEN PARITY CHARACTER
3620  017402  000207                  3$:     RTS     PC
3621  017404  062716  000002          TRPREG: ADD     #2,(SP) ;ALLOW IT TO "CRUNCH" INTO HLT BACK
3622                                                          ;IN MAIN PART OF THE PROGRAM
3623  017410  000002                          RTI
3624          000001                  .END
```

```
AAA     003200            1292#
ABASE = 000000             875     916
ACDW1 = 000000             875     918
ACDW2 = 000000             875     919
ACPUOP= 000000             875     890
ACTREG  001166             734#   1248#   1262#   1263#   1270#   2738    2741    2751
ADDW0 = 000000             875     920
ADDW1 = 000000             875     921
ADDW10= 000000             875     930
ADDW11= 000000             875     931
ADDW12= 000000             875     932
ADDW13= 000000             875     933
ADDW14= 000000             875     934
ADDW15= 000000             875     935
ADDW2 = 000000             875     922
ADDW3 = 000000             875     923
ADDW4 = 000000             875     924
ADDW5 = 000000             875     925
ADDW6 = 000000             875     926
ADDW7 = 000000             875     927
ADDW8 = 000000             875     928
ADDW9 = 000000             875     929
ADEVCT= 000000             875     881
ADEVM = 000000             875     917
ADRCNT= 013703            2953#   2992#   2999#
AEMV  = 000000             875     886
AEMVM = 000000             875     887
AFATAL= 000000             875     878
AMADR1= 000000             875     903
AMADR2= 000000             875     907
AMADR3= 000000             875     910
AMADR4= 000000             875     913
AMAMS1= 000000             875     897
AMAMS2= 000000             875     905
AMAMS3= 000000             875     908
AMAMS4= 000000             875     911
AMSGAD= 000000             875     883
AMSGLG= 000000             875     884
AMSGTY= 000000             875     877
AMTYP1= 000000             875     898
AMTYP2= 000000             875     906
AMTYP3= 000000             875     909
AMTYP4= 000000             875     912
APASS = 000000             875     880
APRIOR= 000000             875
APTCSU= 000040             535#   2868
APTEMV= 000001             535#   2861    3115
APTSIZ= 000200             535#   1167
APTSPO= 000100             535#   2863
ASWREG= 000000             875     888
ATESTN= 000000             875     879
AUNIT = 000000             875     882
AUSWR = 000000             875     889
AVECT1= 000000             875     914
AVECT2= 000000             875     915
BASEAD  001154             729#   1230#   1267#   1268    1274#   1276#   2745#   2757#   2761
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
BASEIV  001162            732#   1240#  2746#  2758#  2763   2764#  2765   2766#  2767   2768#  2769   2770#
B88     003026           1247    1251#
BINWRD  014152           3034#   3036   3066#
BITW  = 002000            801#    994#  1372   1425   1478   1531   1584   1637   1691   1744   1797   1850   1903
                         1956    2273   2320
BIT0  = 000001            664#    777    808    970   1001
BIT00 = 000001            654#    664
BIT01 = 000002            653#    663
BIT02 = 000004            652#    662
BIT03 = 000010            651#    661
BIT04 = 000020            650#    660
BIT05 = 000040            649#    659
BIT06 = 000100            648#    658
BIT07 = 000200            647#    657
BIT08 = 000400            646#    656   3453
BIT09 = 001000            645#    655   3125   3461
BIT1  = 000002            663#    776    969
BIT10 = 002000            644#    767    801    960    994   3103
BIT11 = 004000            643#    766    959   3468
BIT12 = 010000            642#    765    782    958    975
BIT13 = 020000            641#    764    781    800    957    974    993   3110
BIT14 = 040000            640#    763    780    799    956    973    992   3439
BIT15 = 100000            639#    762    779    798    955    972    991   3580
BIT2  = 000004            662#    775    968
BIT3  = 000010            661#    774    807    967   1000
BIT4  = 000020            660#    773    806    966    999
BIT5  = 000040            659#    772    805    965    998
BIT6  = 000100            658#    771    804    964    997
BIT7  = 000200            657#    770    803    963    996
BIT8  = 000400            656#    769    785    802    962    978    995   3599   3601   3616   3618
BIT9  = 001000            655#    768    784    961    977
BPTVEC= 000014            671#
BREAK = 000001            808#   1001#  2003   2060   2093   2351   2415   2677
BRW     016642           3486#
BRX     016644           3487#
CARDET= 010000            765#    958#
CCC     012634           2737    2760   2777#
CHRCNT  014150           3032#   3037   3050#  3064#  3065
CKSWR   012734           2731    2793   2801#  2833   3430
CLK   = 020000            800#    993#  1367   1368   1374   1375   1385   1420   1421   1427   1428   1438   1473
                         1474    1480   1481   1491   1526   1527   1533   1534   1544   1579   1580   1586   1587
                         1597    1632   1633   1639   1640   1650   1686   1687   1693   1694   1704   1739   1740
                         1746    1747   1757   1792   1793   1799   1800   1810   1845   1846   1852   1853   1863
                         1898    1899   1905   1906   1916   1951   1952   1958   1959   1969   2003   2009   2010
                         2012    2013   2026   2027   2029   2030   2060   2065   2066   2068   2069   2078   2079
                         2093    2098   2099   2111   2112   2146   2147   2152   2153   2193   2194   2197   2198
                         2222    2223   2267   2268   2276   2277   2314   2315   2323   2324   2351   2359   2360
                         2362    2363   2415   2422   2423   2425   2426   2560   2566   2574   2580   2603   2609
                         2656    2657   2662   2663   3582   3583
CNTLU   013010           1197    2812#
CONVRT= 104413           2734    2814   3296   3532#
COUNT   001124            709#   2031#  2038#  2192#  2204#  2213#  2226#  2365#  2369#  2373#  2427#  2432#  2505#
                         2510#   2514#  2653#  2679#
CR    = 000015            579#   2907   2917
CRLF  = 000200            580#   2878   2917
CTS   = 020000            764#    957#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
DDISP = 177570            586*    842    1155
DEVADR  013700           2951*   2989    2997*
DEVICE  015343           2735    3326*
DF1     002132           1026    1030    1034    1038    1084*
DH1     002067           1024    1028    1032    1073*
DISPLA  001442            642*   1155*   1163*   1188*   3102*   3483*
DISPRE  000174            683*   1163    1188
DNA   = 100000            798*    991*
DNAINT= 000040            805*    998*
DSC   = 100000            762*    955*
DSINTE= 000040            772*    965*   2704    2710    2718
DSR   = 001000            768*    961*
DSWR  = 177570            585*    841    1154
DTR   = 000002            776*    969*   2711
DT1     002116           1025    1029    1033    1079*
DT4     002126           1037    1082*
DUADDR  017012           1229    2762    3552*
DUBASE  017144           1225    1228    2761*   3552    3553*   3554    3555*   3556    3557    3558*   3559    3560    3561*
                         3562    3563*   3564    3565*   3566    3567*   3568    3570*
DULEV   016736           1287    3539*
DUPRT   017006           1286*   2696    3539*   3540*   3541*   3542*   3543*   3544    3548*
DURIS   001740           1053*   2696*   2706    2707*   2715    2716*   2765*
DURIV   001736           1052*   1236    1239    1240    2695*   2706*   2715*   2763*   2770
DUTIS   001744           1055*   2769*
DUTIV   001742           1054*   2767*
EIGHT = 006000            793*    986*   1467    1520    1573    1626    1680    1733    1786    1839    1892    1945    2006
                         2063    2096    2140    2188    2260    2308    2354    2418    2556    2650
EMTVEC= 000030            674*   1139*   1140*
EM1     001762           1023    1059*
EM2     002022           1027    1065*
EM3     002043           1031    1068*
EM4     001746           1035    1057*
ERRCNT  001114            702*
ERRVEC= 000004            667*   1152    1153*   1164*   3444    3445*   3447*   3450*
EVENB   017320           3606*
EVEPAR= 001400            796*    989*   1361    1467    1573    1680    1786    1892
EVPAR = 000400            785*    978*
FIVE  = 000000            790*    983*
FRMERR= 020000            781*    974*
GNS   = ****** U         3520    3521    3522    3523    3526    3527    3528    3529    3530    3531    3532    3533
HOXEN = 000010            807*   1000*   2600
HILIM   013676           2950*   2980    2996*
HLD0    001130            711*   1079    3133*
HLD1    001132            712*   1079    3134*
HLD2    001134            713*   3135*
HLD3    001136            714*   3136*
HLD4    001140            715*   3137*
HLD5    001142            716*   3138*
HLD6    001144            717*   3139*
HOLD    001120            707*   2562    2568    2576    2582    2605    2611
HPARCS  001724           1046*   3560*
HRXCSR  001714           1042*   3554*
HRXDBU  001720           1044*   3559*
HT    = 000011            577*   2876    2917
HTXCSR  001730           1048*   3564*
HTXDBU  001734           1050*   3568*
```

```
INBUF   016164          1295    1299    2926    2956    3074    3078    3405#
INIFLG  001172           742#   1198    1201#
INSTER= 104407          1303    2975    3083    3528#
INSTR = 104406          1220    1231    1241    1252    1277    1293    1306    1310    1314    1318    1329    2816    3527#
INSTR2  013502          2923    2935    2944#
IOTVEC= 000020           672#   1137#   1138#
ISYMOD= 000000           789#    982#   1353    1361    1406    1414    1565    1573    1618    1626
JMRBY   001153           725#   1206#   1321    2544    2688
KEEPAD  001156           730#   1228#   1230    1274    1276    2757
KEEPIV  001164           733#   1239#   2758
LASTAD  001160           731#   1257    1268    1282
LESS1   017010          3544#   3545#   3546#   3549#
LF    = 000012           578#   2911    2917
LIGHTS  001102           692#   2782#
LIMITS  013630          2966    2980#
LOBITS  013702          2952#   2984    2998#   2999
LOCK    001110           700#   2836
LOKFLG  001174           744#
LOLIM   013674          2949#   2982    2995#
MCNTG   016131          2811    3395#
MCOM    015355          2740    3328#
MCRLF   015455          3028    3341#
MDATA   016270          3048    3059    3409#
MEPASS  015524          2733    3348#
MEXT  = 010000           812#   1005#   2553
MEXTJ   016071          1319    3389#
MINT  = 004000           811#   1004#   1358    1411    1464    1517    1570    1623    1677    1730    1783    1836    1889
                        1942    2003    2060    2093    2137    2185    2257    2305    2351    2415    2647
MLASTD  015311          1253    3321#
MLOCK   015570          3355#
MMNEW   016152          2816    3399#
MMSHR   016141          2813    3397#
MMULT   015262          1242    3317#
MPFAIL  015460          3299    3342#
MQM     015451          2942    3340#
MR      015551          1339    3352#
MRANGE  015405          1278    3333#
MREGAD  015232          1221    3313#
MRESET= 000400           802#    995#   1352    1354    1405    1407    1458    1460    1511    1513    1564    1566    1617
                        1619    1671    1673    1724    1726    1777    1779    1830    1832    1883    1885    1936    1938
                        1998    2000    2055    2057    2088    2090    2131    2133    2179    2181    2251    2253    2299
                        2301    2346    2348    2410    2412    2547    2549    2641    2643    2690
MSYNC   015623          1294    3360#
MTDATA= 040000           799#    992#   2003    2060    2073    2077    2093    2105    2109    2351    2415    3575
MTITLE  015164          1200    3306#
MTSTPC  015555          1330    3353#
MULTD   001152           724#   1244    1245    2736
MVECTO  015217          1232    3311#
MWIRE4  016012          1315    3381#
MWIRE5  015742          1311    3374#
MWIRE6  015671          1307    3367#
NEXT    001106           699#
NOPAR = 000000           794#    987#   2006    2063    2096    2140    2188    2260    2308    2354    2418    2556    2650
ODDPAR= 001000           795#    988#   14.4    1520    1626    1733    1839    1945
ODDB    017234          3589#
ONCE    002610          1199    1202#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
OPTCLR  001151            723#  1317  2691
OUT     013042           2802  2804  2806  2810  2822#
OUTCRY  012626           2734  2773#
OUTMUL  003166           1250  1275  1286#
OVRRUN= 040000            780#  973#  2231
PARAM = 104410           1222  1233  1254  1279  1331  2817  3529#
PARAM1  013534           2955#  2976
PARCSR  001722           1045#  1353#  1361#  1406#  1414#  1459#  1467#  1512#  1520#  1565#  1573#  1618#  1626#
                         1672#  1680#  1725#  1733#  1778#  1786#  1831#  1839#  1884#  1892#  1937#  1945#  1990#
                         2006#  2056#  2063#  2089#  2096#  2132#  2140#  2180#  2188#  2252#  2260#  2300#  2308#
                         2347#  2354#  2411#  2418#  2548#  2556#  2642#  2650#  3557#
PAREN = 001000            784#  977#
PARER = 010000            782#  975#
PARERR  013610           2958  2960  2962  2971#  2981  2983  2985
PARTI   013672           2974  2994#
PASCNT  001112            701#  1175#  2781#  2782  2783
PFTAB   015156           3297  3303#
PIRQ  = 177772            584#
PIRQVE= 000240            678#
POPR0 = 012600            757#  950#
POP1SP= 005726            755#  948#
POP2SP= 022626            759#  952#
PR0   = 000000            601#
PR1   = 000040            602#
PR2   = 000100            603#
PR3   = 000140            604#
PR4   = 000200            605#  3549
PR5   = 000240            606#  3548
PR6   = 000300            607#
PR7   = 000340            608#
PS    = 177776            581#  582
PSW   = 177776            582#
PUSHR0= 010046            756#  949#
PUSH1S= 005746            754#  947#
PUSH2S= 024646            758#  951#
PWRVEC= 000024            673#  1143#  1144#
RDSW    013050           2812#  2822#  2824#  2973
RECACT= 004000            766#  959#  2017  2032  2040  2375  2438  2467  2474  2483  2491  2516
REPLAY  012532           2753  2759  2761#
RESTAR  015062           3277  3283#
RESTRT  012712           2754  2785  2792#
RESVEC= 000010            668#
RESOS = 104412           3531#
RETURN  001104            698#  1180#  1338#  1340  3302
RING  = 040000            763#  956#
RINTEN= 000100            771#  964#
ROTADD  001170            737#  1249#  1261#  1263  1265#  1270  1273#  2748#  2751  2755#
RPOKE   017146           2016  2022  2037  2372  2383  2431  2452  2466  2481  2490  2513  2524  3575#
                         3585
RTS   = 000004            775#  968#
RUNA  = ****** U             1    51   138   512
RUNB  = ****** U             1    51   138   512
RUNC  = ****** U             1    51   138   512
RUND  = ****** U             1    51   138   512
RUNE  = 000000               1    51   138   512
RUNF  = ****** U            21    57   141   517
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
RUNIT   012436          2739    2745#   2752
RXCSR   001712          1041#   2007*   2017    2032    2040    2190*   2201    2228    2356*   2375    2378    2384    2420*
                        2438    2441    2453    2467    2470    2473*   2474    2482*   2483    2486    2491    2494    2516
                        2519    2525    2557*   2592    2617    2651*   2670    2693*   2694    2701    2704*   2710*   2711*
                        2718*   2775    3552*
RXDBUF  001716          1043#   2072    2081    2114    2191    2207    2232    2364    2388    2428    2456    2498    2529
                        2546    2596    2621    2652    2673    3556*
RXDONE= 000200           770#    963#
RXERR = 100000           779#    972#   2231
SAVIT   014404          3112    3133#
SAVPC   001126           710#   3003*
SAVSP   001116           703#   3276*   3283
SAVOS = 104411          3530#
SCOPI = 104405          3526#
SEND  = 000020           806#    999#   1358    1411    1464    1517    1570    1623    1677    1730    1783    1836    1889
                        1942    2137    2185    2257    2262    2265    2305    2334    2553    2647
SEREC   001150           722#   1313
SETFLG= 104414          1243    1308    1312    1316    1320    3533#
SEVEN = 004000           792#    985#   1361    1414
SEXMIT  001147           721#   1309
SHIFT   001122           708#   1365*   1382*   1418*   1435*   1471*   1488*   1524*   1541*   1577*   1594*   1630*   1647*
                        1684*   1701*   1737*   1754*   1790*   1807*   1843*   1860*   1896*   1913*   1949*   1966*   2014*
                        2020*   2035*   2070*   2086*   2100*   2120*   2195*   2199*   2219*   2224*   2269*   2284*   2312*
                        2331*   2332    2335    2370*   2381*   2429*   2450*   2464*   2480*   2489*   2511*   2522*   2572*
                        2586*   2587    2590    2601*   2615*   2659*   2664*   2665    2668    3584*
SIX   = 002000           791#    984#
SPACNT= 014151          3033*   3052    3055*   3065#
SRD   = 002000           767#    960#
STACK = 001100           572#   1135    1171    1325    3292
STD   = 000010           774#    967#
STFLG   001173           743#   1174*
STKLMT= 177774           583#
STPSYN= 000400           769#    962#   2190    2356
SVOS    013712          3007#
SWR     001440           841#   1133    1154*   1156    1162*   1169*   1185    1187*   1192    1208    1327    2803    2834
                        3103    3110    3122    3125    3439    3453    3455    3461    3468
SWREG   000176           684#   1162    1187    1192    2803    2820    2827
SWREGL  013052          2815    2825#
SW0   = 000001           636#
SW00  = 000001           626#    636    1208
SW01  = 000002           625#    635    1327
SW02  = 000004           624#    634
SW03  = 000010           623#    633
SW04  = 000020           622#    632
SW05  = 000040           621#    631
SW06  = 000100           620#    630
SW07  = 000200           619#    629
SW08  = 000400           618#    628
SW09  = 001000           617#    627    2834
SW1   = 000002           635#
SW10  = 002000           616#
SW11  = 004000           615#
SW12  = 010000           614#
SW13  = 020000           613#
SW14  = 040000           612#
SW15  = 100000           611#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
SW2    = 000004            634#
SW3    = 000010            633#
SW4    = 000020            632#
SW5    = 000040            631#
SW6    = 000100            630#
SW7    = 000200            629#
SW8    = 000400            628#
SW9    = 001000            627#
SYNCNO  001146             720#  1297#  1301#  1996   2367   2435   2508
SYNEXT= 020000             788#   981#  2089   2096   2180   2188   2252   2260   2300   2308   2548   2556   2642
                          2650
SYNINT= 030000             787#   980#  1459   1467   1512   1520   1672   1680   1725   1733   1778   1786   1831
                         1839   1884   1892   1937   1945   1999   2006   2056   2063   2132   2140   2347   2354
                         2411   2418
SYNSCH= 000020             773#   966#  2007   2190   2356   2420   2473   2482   2557   2651
SYSTST= 014000             813#  1006#
TBITVE= 000014             669#
TEMP    016226            3038   3293#  3294#  3407#
TKVEC  = 000060            676#
TPVEC  = 000064            677#
TRAPVE= 000034             675#  1141#  1142#
TRPREG  017404            3621#
TRTVEC= 000014             670#
TST1    003362            1332   1338   1351#  2794   2795   3432
TST10   005234            1723#
TST11   005442            1776#
TST12   005650            1829#
TST13   006056            1882#
TST14   006264            1935#
TST15   006472            1995#
TST16   006760            2054#
TST17   007342            2129#
TST2    003570            1404#
TST20   007470            2177#
TST21   007756            2249#
TST22   010150            2297#
TST23   010350            2345#
TST24   010610            2408#
TST25   011400            2543#
TST26   011742            2640#
TST27   012164            2687#
TST3    003776            1457#
TST4    004204            1510#
TST5    004412            1563#
TST6    004620            1616#
TST7    005026            1670#
TTST    016350            3435#
TXCSR   001726            1047#  1352#  1354#  1358#  1362   1367#  1368#  1374#  1375#  1376   1385#  1387   1405#
                          1407#  1411#  1415   1420#  1421#  1427#  1428#  1429   1438#  1440   1452#  1460#  1464#
                          1468   1473#  1474#  1480#  1481#  1482   1491#  1493   1511#  1513#  1517#  1521   1526#
                          1527#  1533#  1534#  1535   1544#  1546   1564#  1566#  1570#  1574   1579#  1580#  1586#
                          1587#  1588   1597#  1599   1617#  1619#  1623#  1627   1632#  1633#  1639#  1640#  1641
                          1650#  1652   1671#  1673#  1677#  1681   1686#  1687#  1693#  1694#  1695   1704#  1706
                          1724#  1726#  1730#  1734   1739#  1740#  1746#  1747#  1748   1757#  1759   1777#  1779#
                          1783#  1787   1792#  1793#  1799#  1800#  1801   1810#  1812   1830#  1832#  1836#  1840
                          1845#  1846#  1852#  1853#  1854   1863#  1865   1883#  1885#  1889#  1893   1898#  1899#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
                        1905*  1906*  1907   1916*  1918   1936*  1938*  1942*  1946   1951*  1952*  1958*  1959*
                        1960   1969*  1971   1998*  2000*  2003*  2009*  2010*  2012*  2013*  2026*  2027*  2029*
                        2030*  2055*  2057*  2060*  2065*  2066*  2068*  2069*  2073*  2077*  2078*  2079*  2088*
                        2090*  2093*  2098*  2099*  2105*  2109*  2111*  2112*  2131*  2133*  2137*  2142   2145*
                        2147*  2148   2152*  2153*  2154   2179*  2181*  2185*  2193*  2194*  2197*  2198*  2222*
                        2223*  2251*  2253*  2257*  2261   2262*  2265*  2267*  2268*  2276*  2277*  2278   2299*
                        2301*  2305*  2309   2314*  2315*  2323*  2324*  2325   2334*  2346*  2348*  2351*  2359*
                        2360*  2362*  2363*  2410*  2412*  2415*  2422*  2423*  2425*  2426*  2547*  2549*  2553*
                        2560*  2566*  2574*  2580*  2600*  2603*  2609*  2641*  2643*  2647*  2656*  2657*  2662*
                        2663*  2677*  2690*  3562*  3575*  3581*  3582*  3583*
TXDBUF  001732          1049*  1363*  1416*  1469*  1522*  1575*  1628*  1682*  1735*  1788*  1841*  1894*  1947*
                        2145*  2189*  2214*  2264*  2310*  2558*  2589*  2654*  2667*  3566*
TXDONE= 000200           803*   996*
TXINTE= 000100           804*   997*
TYPE  = 104401          1200   1339   2732   2735   2740   2811   2813   2881   2924   2941   3027   3058   3105
                        3113   3149   3166   3168   3171   3173   3177   3184   3248   3298   3520*
TYPOC = 104402          3157   3181   3521*
TYPON = 104404          3523*
TYPOS = 104403          3522*
USER  = 000000           810*  1003*
VOID  = 000001           777*   970*
WRDCNT  014146          3031*  3060*  3063*
ZERO    012444          2746*
$APTHD  002136          1106   1112*
$ASTAT= ****** U         535
$ATYC   000024R          535*
$ATY1   000000R          535*
$ATY3   000006R          535*  2866
$ATY4   000016R          535*  3118
$AUTOB  001434           838*
$BASE   001602           916*
$BDADR  001422           833*
$BDDAT  001426           835*
$BELL   001516           867*  3105   3133
$CDW1   001606           918*
$CDW2   001610           919*
$CHARC  013362          2883*  2893*  2900   2909*  2914*
$CKSUM= ****** U        3526
$CMTAG  001400           821*  1130   1131   1139   1145   1146
$CM1  = 000006           853*   854*   855*   856*   857*   858*   859*
$CM2  = 000014           853*   854*   855*   856*   857*   858*   859*
$CM3  = 000006           851*   853
$CM4  = 000006           859*   860*   861*   862*   863*   864*   865*
$CPUOP  001554           890*
$CRLF   001523           869*  2882   2917   3113   3133   3149   3168   3173   3177
$DDW0   001612           920*
$DDW1   001614           921*
$DDW10  001636           930*
$DDW11  001640           931*
$DDW12  001642           932*
$DDW13  001644           933*
$DDW14  001646           934*
$DDW15  001650           935*
$DDW2   001616           922*
$DDW3   001620           923*
$DDW4   001622           924*
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
$DDW5   001624            925#
$DDW6   001626            926#
$DDW7   001630            927#
$DDW8   001632            928#
$DDW9   001634            929#
$DEVCT  001536            881#
$DEVM   001604            917#
$E    = 000002            533#
$ENDAD  012702           1093    2788#
$ENV    001546            535     886#   1202   2861   2922   3115
$ENVM   001547            535     887#   1167   2863   2868
$ERFLG  001403            824#   1176*  2780*  3100*  3133   3300*  3416   3457   3459   3465*  3489
$ERMAX  001415            830#   1147*  3459   3482*  3489
$ERROR  014222           1139    3099#
$ERRPC  001416            831#   1079   1082   1178*  2779*  3107*  3108*  3109   3133   3155   3301*  3431*
$ERRTB  001652           1021#   3163
$ERRTY  014442           3148#
$ERTTL  001412            828#   1177*  3106*  3133
$ESCAP  001514            866#   1146*  3128   3130   3133   3481*
$ETABL  001546            885#
$ETEND  001652            938#   1118
$FATAL  001530            535*    878#
$FFLG   000244R           535#*
$FILLC  001456            849#   2886   2917
$FILLS  001455            848#   2917
$GDAOR  001420            832#
$GDDAT  001424            834#
$GTSWR= ****** U         3525
$HIBTS  002136           1113#
$ICNT   001404            825#   3472*  3473   3475*  3488
$INTAG  001435            839#
$ITEMB  001414            829#   3109*  3117   3133   3152
$LF     001524            870#   2917   3133
$LFLG   000243R           535#*
$LPAOR  001406            826#   1148*  2794*  3463*  3479*  3484   3488
$LPERR  001410            827#   1149*  3127   3463   3480*  3488
$MAOR1  001560            903#
$MAOR2  001564            907#
$MAOR3  001570            910#
$MAOR4  001574            913#
$MAIL   001526            876#   1114   1118   1166   2861   3115   3478
$MAMS1  001556            897#
$MAMS2  001562            905#
$MAMS3  001566            908#
$MAMS4  001572            911#
$MBAOR  002140           1114#
$MFLG   000242R           535#*
$MSGAD  001542            535*    883#
$MSGLG  001544            535*    884#
$MSGTY  001526            535*    877#
$MTYP1  001557            898#
$MTYP2  001563            906#
$MTYP3  001567            909#
$MTYP4  001573            912#
$MXCNT  016646           3476    3488#
$N    = 000000            533#   2724#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
$NULL   001454            847#  2888    2317
$NWTST= 000000           1350#  1403#   1456#  1509#  1562#  1615#  1669#  1722#  1775#  1828#  1881#  1934#  1994#
                         2053#  2128#   2176#  2248#  2296#  2344#  2407#  2542#  2639#  2686#
$OCNT   015020           3220*  3249*   3262#
$OMODE  015022           3215*  3219*   3224   3227*  3238*  3264#
$OVER   016626           3437   3440    3456   3464   3474   3483#
$PASS   001534            880#  1166*   2783*  3470   3489
$PASTM  002144           1116#
$PWRDN  015024           1143   3268#
$QUES   001522            868#  2917    3133
$RDCHR= ****** U         3526
$RDDEC= ****** U         3526
$RDLIN= ****** U         3526
$RDOCT= ****** U         3526
$REGAD  001460            851#
$REG0   001462            853#  3012*   3017
$REG1   001464            854#  3011*   3018
$REG2   001466            855#  3010*   3019
$REG3   001470            856#  3009*   3020
$REG4   001472            857#  3008*   3021
$REG5   001474            858#  3007*   3022
$R2A  = ****** U         3526
$SAVRE= ****** U         3526
$SCOPE  016332           1137   3425#
$SETUP= 000017           1119#  1136    1137   1139   1141   1143   1145   1146   1148   3100   3125   3132   3426
$STUP = 177777           1119#
$SVLAD  016572           3448   3477#
$SVPC = 002136           1091#  1096
$SWR  = 177400            524#   865    866    867    1145   1146   1148   1149   1352   1405   1458   1511   1564
                         1617   1671    1724   1777   1830   1883   1936   1996   2055   2130   2178   2250   2298
                         2346   2409    2544   2641   2688   3091   3092   3093   3094   3095   3103   3110   3122
                         3125   3133    3417   3418   3419   3420   3421   3439   3451   3453   3454   3457   3458
                         3459   3466    3467   3468   3480   3483   3488
$SWREG  001550            888#  1169
$SWRMK= 000000           3421   3422    3455
$TESTN  001532            879#  3478*
$TIMES  001512            869#  1145*   3466*  3473   3476*  3488
$TKB    001446            844#  2807    2930   2938   3438
$TKS    001444            843#  2805    2928   3436
$TMP0   001476            879#
$TMP1   001500            860#  1364*   1370*  1417*  1423*  1470*  1476*  1523*  1529*  1576*  1582*  1629*  1635*
                         1683*  1689*   1736*  1742*  1789*  1795*  1842*  1848*  1895*  1901*  1948*  1954*  2015*
                         2021*  2036*   2071*  2075*  2101*  2107*  2263*  2271*  2311*  2318*  2371*  2382*  2430*
                         2451*  2465*   2512*  2523*  3577*  3589   3599*  3601*  3606   3616*  3618*
$TMP2   001502            861#  2102*   2103*  2110*  2113   3576*  3578*  3579*  3580*  3581   3589*  3593*  3606*
                         3610*
$TMP3   001504            862#  3590*   3594*  3597*  3607*  3611*  3614*
$TMP4   001506            863#
$TMPS   001510            864#
$TN   = 000030            535#  1350    1352#  1403   1405#  1456   1458#  1509   1511#  1562   1564#  1615   1617#
                         1669   1671#   1722   1724#  1775   1777#  1828   1830#  1881   1883#  1934   1936#  1994
                         1996#  2053    2055#  2128   2130#  2176   2178#  2248   2250#  2296   2298#  2344   2346#
                         2407   2409#   2542   2544#  2639   2641#  2686   2688#
$TPB    001452            846#  2906*   2917   2938*
$TPFLG  001457            850#  2855    2917
$TPS    001450            845#  2904    2917   2936
```

D07

CROSS REFERENCE TABLE -- USER SYMBOLS

```
$TRAP   016650           1141    3497#
$TRAP2  016672           3508#   3519
$TRP  = 000015           3512#   3521#   3522#   3523#   3524#   3526    3527#   3528#   3529#   3530#   3531#   3532#   3533#
                         3534#
$TRPAD  016704           3502    3519#
$TSTM   002142           1115#
$TSTNM  001402            823#   1179#   2778#   3102    3133    3139    3416    3455    3477#   3478    3483    3489
$TYPBN= ****** U         3524
$TYPOS= ****** U         3524
$TYPE   013104            535    2855#   3512    3520
$TYPEC  013316           2885    2892    2899    2904#   2905
$TYPEX  013364           2910    2912    2915#
$TYPOC  014622           3218#   3521
$TYPON  014636           3217    3220#   3523
$TYPOS  014576           3213#   3522
$UNIT   001540            882#
$UNITM  002146           1117#
$USWR   001552            889#   1204
$VECT1  001576            914#
$VECT2  001600            915#
$XTSTR  016376           3433    3442#
$OFILL  015021           3214*   3218*   3228    3263#
$40CAT= ****** U         3112    3439
.       = 017412          535#    569#    682#    685#    690#    745#    746#    820#    871    1077#   1091    1092#   1094#
                         1096#   1102    1103#   1105#   1107#   1134    1148    1149    1379    1390    1432    1443    1485
                         1496    1538    1549    1591    1602    1644    1655    1698    1709    1751    1762    1804    1815
                         1857    1868    1910    1921    1963    1974    2018    2033    2041    2083    2116    2143    2149
                         2155    2202    2209    2229    2234    2281    2328    2376    2379    2385    2390    2439    2442
                         2454    2459    2468    2471    2475    2484    2487    2492    2495    2500    2517    2520    2526
                         2531    2564    2570    2578    2584    2593    2598    2607    2613    2618    2623    2671    2675
                         2702    2714    2917    3133    3187#   3279    3295    3406#   3408#   3410#   3488    3489
.BEGIN  003310           1196    1207    1210    1325#
.CONVR  013776           3027#   3532
.EOP    012362           2730#
.INSTE  013474           2941#   3528
.INSTR  013366           2920#   3527
.INST1  013406           2924#   2933    2943
.MSG    013410           2807*   2808*   2809    2920*   2925#
.PARAM  013504           2948#   3529
.PFAIL  015024           1173    3269#   3290
.RESOS  013744           3017#   3531
.SAVOS  013704           3003#   3530
.SCOPE  016332           3429#
.SCOP1  013062           2833#   3526
.SETFL  014154           3073#   3084    3533
.START  002152            686    1128#   1180    2744
.$ASTA= ****** U          535
.$X   = 002136           1102#   1107
```

CROSS REFERENCE TABLE -- MACRO NAMES

```
COMMEN    679#
ENDCOM    679#
ERROR     573#  1380  1391  1433  1444  1486  1497  1539  1550  1592  1603  1645  1656  1699  1710
          1752  1763  1805  1816  1858  1869  1911  1922  1964  1975  2019  2034  2042  2084  2117
          2144  2150  2156  2203  2210  2230  2235  2282  2329  2377  2380  2386  2391  2440  2443
          2455  2460  2469  2472  2476  2485  2488  2493  2496  2501  2518  2521  2527  2532  2594
          2599  2619  2624  2672  2676  2703  2719
ESCAPE    679#
GETPRI    679#
GETSWR    679#
MULT      679#
NEWTST    679#  1350  1403  1456  1509  1562  1615  1669  1722  1775  1828  1881  1934  1994  2053
          2128  2176  2248  2296  2344  2407  2542  2639  2686
POP       535    679#
PRGEND    557#  2723
PRGFRT    536#   558
PUSH      535    679#
PUSSYF    554#  2560  2573  2602
REPORT    524#   679#
RSETUP    541#  1998  2055  2088  2346  2410
SC        533#  3426
SCOPE     574#  1351  1404  1457  1510  1563  1616  1670  1723  1776  1829  1882  1935  1995  2054
          2129  2177  2249  2297  2345  2408  2543  2640  2687  2730
SC1       533#  3485
SETPRI    679#
SETTRA    524#  3512#  3521  3522  3523  3526  3527  3528  3529  3530  3531  3532  3533
SETUP     679#  1128
SKIP      679#
SLASH     679#
SPACE     679#
STARS     535    679#   816   871   874  1089  1099  1101  1108  1350  1403  1456  1509  1562  1615
          1669  1722  1775  1828  1881  1934  1994  2053  2128  2176  2248  2296  2344  2407  2542
          2639  2686  2840  3087  3143  3190  3413  3491
SWRSU     679#  1150#
TRMTRP   3512#
TSETUP    542#  1352  1405  1458  1511  1564  1617  1671  1724  1777  1830  1883  1936  2131  2179
          2251  2299  2547  2641
TYPBIN    679#
TYPDEC    679#
TYPNAM    679#
TYPNUM    679#
TYPOCS    679#
TYPOCT    679#  3155  3179
TYPTXT    679#
$BEGIN    533#  1322
$BINAR    553#
$BUFFE    535#  3402
$CABLE    555#
$CATCH    527#   569
$CK       534#  2796
$CLRVE    533#  1202
$CONVR    535#  3024
$DMA      551#
$EOP      534#  2723
$GETFL    533#  1241  1306  1310  1314  1318
$GETPA    533#  1220  1231  1251  1277  1329
```

CROSS REFERENCE TABLE -- MACRO NAMES

```
$GETSY     533#   1288
$HEADE     524#    558
$INSTR     535#   2917
$ISOB      549#
$MATCH     546#
$MRR       540#
$MRRW      538#
$MRW       539#
$MSG       535#   3306
$NEED      556#   3133
$PARAM     535#   2945
$PFAIL     535#   3265
$POKE      544#   1367   1373   1420   1426   1473   1479   1526   1532   1579   1585   1632   1638   1686   1692
          1739    1745   1792   1798   1845   1851   1898   1904   1951   1957   2146   2152   2193   2196   2221
          2267    2275   2314   2322   2656   2661
$POKER     545#   2011   2025   2028   2064   2067   2097   2361   2424
$RCNET     556#
$RECAC     544#
$REG       535#   3000
$RESET     537#   1352   1354   1405   1407   1458   1460   1511   1513   1564   1566   1617   1619   1671   1673
          1724    1726   1777   1779   1830   1832   1883   1885   1936   1938   1998   2000   2055   2057   2088
          2090    2131   2133   2179   2181   2251   2253   2299   2301   2346   2348   2410   2412   2547   2549
          2641    2643   2690
$RXACT     543#
$SCOPE     533#   3411
$SCOP1     534#   2830
$SETFL     533#   3067
$SETVE     528#    679
$START     533#   1120
$STRIP     550#
$SYMBO     524#    747    940
$SYNCR     546#   2007   2420
$TRPAR     552#   1342   1395   1448   1501   1554   1607   1661   1714   1767   1820   1873   1926
$TSTNO     533#   1350   1403   1456   1509   1562   1615   1669   1722   1775   1828   1881   1934   1994   2053
          2128    2176   2248   2296   2344   2407   2542   2639   2686
$UNIBU     536#
$VARIA     531#    689
$WORDF     547#
$WORDO     546#
$WORDP     548#
$$CMRE     814#    853    854    855    856    857    858
$$CMTM     814#    859    860    861    862    863    864
$$ESCA     679#
$$NEWT     679#   1350   1403   1456   1509   1562   1615   1669   1722   1775   1828   1881   1934   1994   2053
          2128    2176   2248   2296   2344   2407   2542   2639   2686
$$SET     3512#   3521   3522   3523   3526   3527   3528   3529   3530   3531   3532   3533
$$SETM    1166#
$$SKIP     679#
.EQUAT     524#    569
.HEADE     524#
.SETUP     524#   1119
.$ACT1     524#   1087
.$APTB     524#    372#
.$APTH     524#   1097
.$APTY     524#    535
.$CATC     524#
```

```
DZDUU-A MACY11 27(1006)  03-FEB-77  08:00  PAGE 88
DZDUUA.M11    13-OCT-76 08:28          CROSS REFERENCE TABLE -- MACRO NAMES

.$CMTA     524#    814
.$EOP      524#
.$ERRO     524#   3085
.$ERRT     524#   3141
.$POWE     524#
.$SCOP     524#   3411
.$TRAP     524#   3489
.$TYPE     524#   2838
.$TYPO     524#   3188


. ABS.  017412     000


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DZDUUA,DZDUUA.SEQ/SOL/CRF=DZDUU1/EQ:RUNE,DZDUU2,DZDUUA
 RUN-TIME:  17  12  1 SECONDS
 RUN-TIME RATIO: 213/31=6.7
 CORE USED:  31K  (62 PAGES)


EOF1DZDUUASEQ              00010000      770323             PDP10 411
```