DU11

OFF-LINE COMBINED TEST

MD-11-DZDUF-C

EP-DZDUF-C-DL-A

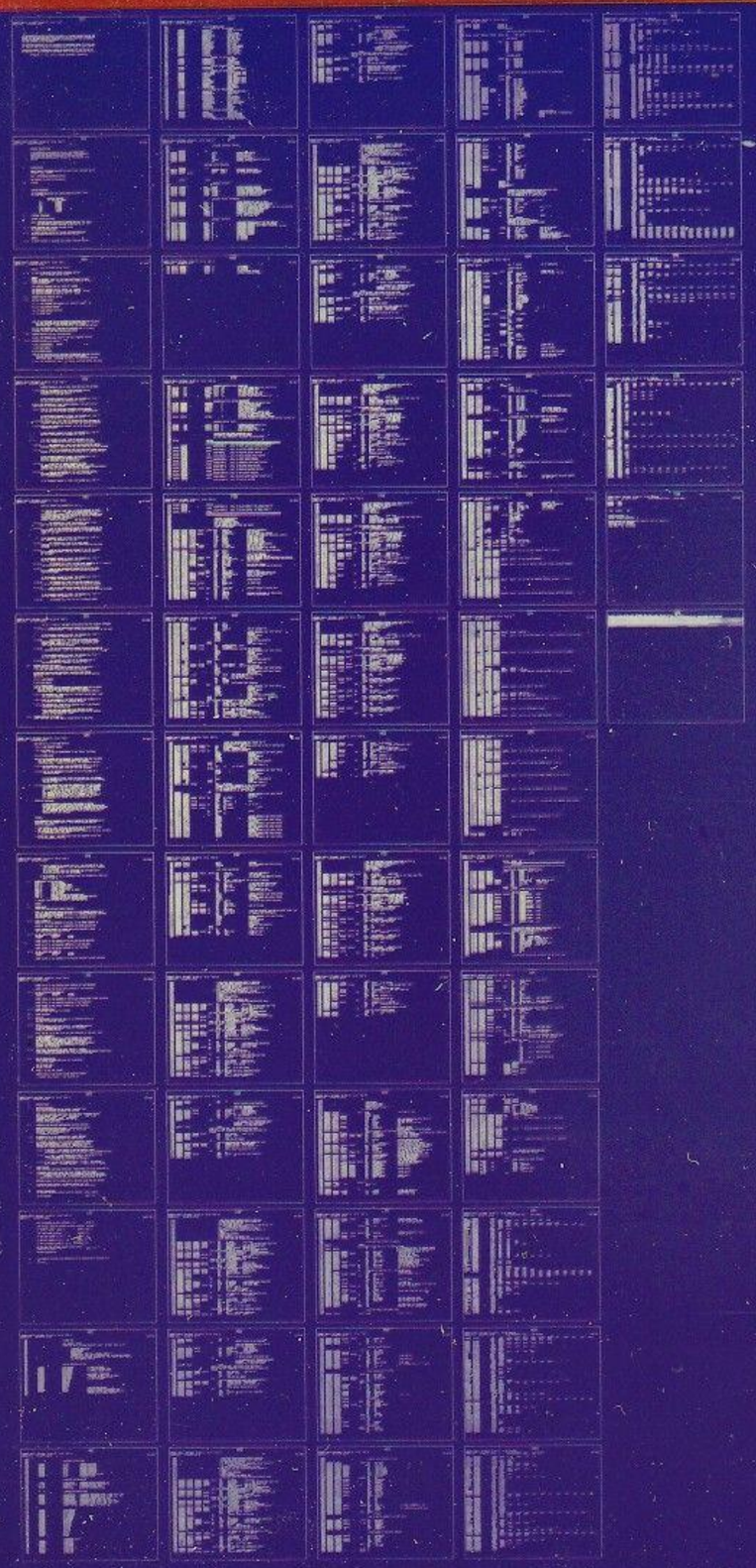
COPYRIGHT © 1976

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A.



THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OF RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973, 1976 BY DIGITAL EQUIPMENT CORPORATION

GENERAL DESCRIPTION

THIS DIAGNOSTIC CAN CHAIN 16 DU11'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.

2. REQUIREMENTS

PDP-11 FAMILY STANDARD COMPUTER WITH OR WITHOUT HARDWARE SWITCH
REGISTER (LOC. 177570)

DU11 SYNCRONOUS/ISOCRONOUS OPTION

ONE CONSOLE TELETYPE OR EQUIVALENT

2.2 STORAGE

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: SOFTWARE SWITCH REGISTER IS DEFINED AS LOC. 176, WHILE
THE SOFTWARE DISPLAY REGISTER IS DEFINED AS LOC. 174.

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 LOAD ADDRESS 000200

4.3.1.3 CLEAR CONSOLE SWITCHES

4.3.1.4 PRESS START

NOTE:IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING
WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:
SWR=XXXXXX NEW= (REFER TO SECTION 5. FOR OPERATOR'S OPTION)

4.3.1.7 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 LOAD ADDRESS 000200

4.3.3.2 SET SW00=1

4.3.3.3 PRESS START

NOTE:IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING
WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:
SWR=XXXXXX NEW= (REFER TO SECTION 5. FOR OPERATOR'S OPTION)

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 DU11 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 DU11 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 DU11 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 DU11 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.....
....SCHUMS 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 "DU PRIORITY LEVEL-" AND
WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.13 TYPE IN THE APPROPRIATE DEVICE PRIORITY LEVEL OF THE
DU11 OR DU11'S TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>
(NOTE THAT ALL MULTIPLE DEVICES MUST BE AT THE SAME PRIORITY
LEVEL). IE "5"

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

4.3.3.14 THE PROGRAM WILL TYPE "# OF SYNC CHARS
SELECTED (1 OR 2)-" AND WAIT FOR AN INPUT FROM THE TELETYPE
KEYBOARD

4.3.3.15 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 4.3.3.14

4.3.3.16 THE PROGRAM WILL TYPE " IS SEC XMIT JUMPER #6 IN ? (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 THAT ALL MULTIPLE DEVICES
MUST BE THE SAME)

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

4.3.3.18 THE PROGRAM WILL TYPE "IS SEC REC JUMPER # 5 IN ?
(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 "IS OPT CLR ENABLE JUMPER

```
                    # 4 IN ? (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 "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

          4.3.3.23          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 4.3.3.22

          4.3.3.24          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 LOAD 000200

             4.3.4.2 SET SW01=1

             4.3.4.3 PRESS START
             NOTE:IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING
                 WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:
                 SWR=XXXXXX NEW=   (REFER TO SECTION 5. FOR OPERATOR'S OPTION)

             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 LOAD ADDRESS 000200

4.3.5.2 SET SW02 =1
NOTE: IT MAY BE ADVANTAGEOUS TO SET SW01=1 (OPTIONAL)

4.3.5.3 PRESS START

NOTE:IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING
WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:
SWR=XXXXXX NEW= (REFER TO SECTION 5. FOR OPERATOR'S OPTION)

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

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH
REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS
THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER.
IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES
AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH
REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH
REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY
DOING THE FOLLOWING:

1) TYPE CONTROL G (↑G): THIS WILL ALLOW THE TTY TO ENTER DATA INTO
LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.
2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS
OF THE SOFTWARE SWITCH REGISTER.)

3) AFTER THE ''NEW=" HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE
OF THE FOLLOWING AT THE TTY:

A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>.
(ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS
WILL BE ALLOWED)
IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH
REGISTER CONTENTS WILL NOT BE CHANGED.

B) IF A CONTROL U <↑U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU
BACK TO STEP 2.

5.1 OPERATIONAL SWITCH SETTINGS

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
SW08 =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
THERE ARE FOUR DISTINCT ERROR TYPEOUTS

NOTE: IF THE SOFTWARE SWITCH REGISTER IS TO BE CHANGED AFTER A HALT
THE THE OPERATOR IS REQUIRED TO TYPE A <↑G> BEFORE DEPRESSING CONTINUE.
THE FOLLOWING WILL BE TYPED:
SWR=XXXXXX NEW= (REFER TO SECTION 5. FOR OPERATOR OPTION)

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 TRAP CATCHER. 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 SWDO =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...LOAD 000200 IN SWR AND DEPRESS START....
THE PROGRAM WILL CONTINUE WITH THE DEVICE IT WAS IN BEFORE HALTING.

7.2.2.2 .....OR .....LOAD 000200 WITH SWDO =1 AND DEPRESS START....
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.....LOAD & START AT 000200

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 11/20 MACHINE.
IF RUNNING ON AN 11/40 OR AN 11/45 "HOLD:" MUST BE PATCHED TO 40 .

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: 160040

VECTOR ADDRESS- DURIV: 770

```
            ARE YOU RUNNING MULTIPLE DEVICES ?- NO          MULTD: 0
            LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS-  LASTADD: 0
            DU PRIORITY LEVEL-  LEVEL 5             DUPRT: LEVEL 5
            # OF SYNC CHARS SELECTED -  2           SYNCNO: 377
            IS SEC XMIT JUMPER # 6 IN ?-  YES       SEXMIT: 377
            IS SEC REC JUMPER # 5 IN ?-  YES        SEREC: 377
            IS OPT CLR ENABLE JUMPER # 4 IN ?-  YES OPTCLR: 377
            DO YOU HAVE THE EXTERNAL MODEM BYPASS JUMPER
            CONNECTOR ON (H315)-  YES                       JMRBY: 377
    9.      PROGRAM DESCRIPTION


    10.     FLOW CHARTS: RECEIVER FLOW,TRANSMITTER FLOW,TRANSMITTER & RECEIVER FLOW
    11.     LISTINGS
```

```
                                .ENABLE ABS

                                ;DU11 DZDUF-C TAPE F
                                ;COPYRIGHT 1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

                                        ;STARTING PROCEDURE
                                        ;LOAD PROGRAM
                                        ;PRESS START
                                        ;PROGRAM WILL TYPE "DU11 DZDUF-C TAPE F "
                                        ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
                                        ;AT THE END OF A PASS, PROGRAM WILL TYPE "END OF PASS TAPE "
                                        ;AND THEN RESUME TESTING


                                ;SWITCH REGISTER OPTIONS

        100000                  SW15=100000             ;=1,HALT ON ERROR
        040000                  SW14=40000              ;=1,LOOP ON CURRENT TEST
        020000                  SW13=20000              ;=1,INHIBIT ERROR TYPEOUT
        010000                  SW12=10000
        004000                  SW11=4000               ;=1,INHIBIT ITERATIONS
        002000                  SW10=2000               ;=1,ESCAPE TO NEXT TEST ON ERROR
        001000                  SW09=1000               ;=1,LOOP WITH CURRENT DATA
        000400                  SW08=400                ;=1,LOOP ON ERROR
        000100                  SW06=100
        000040                  SW05=40
        000020                  SW04=20
        000010                  SW03=10
        000004                  SW02=4                  ;LOCK ON TEST SELECT
        000002                  SW01=2                  ;RESTART PROGRAM AT SELECTED TEST
        000001                  SW00=1                  ;RESELECT VECTOR AND CONTROL REGISTER
                                                        ;ADDRESS AFTER PROGRAM RESTART
```

```
                                ;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                          SP=%6           ;PROCESSOR STACK POINTER
000007                          PC=%7           ;PROGRAM COUNTER

                                ;LOCATION EQUIVALENCIES

177570                          DSWR=177570     ;HARDWARE SWITCH REGISTER LOC.
177570                          DLIGHTS=177570  ;HARDWARE DISPLAY REGISTER LOC.
177776                          PS=177776       ;PROCESSOR STATUS WORD
001100                          STACK=1100      ;START OF PROCESSOR STACK

                                ;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)+
                                .EQUIV  EMT,HLT ;BASIC DEFINITION OF ERROR CALL


100000                          BIT15=100000
040000                          BIT14=40000
020000                          BIT13=20000
010000                          BIT12=10000
004000                          BIT11=4000
002000                          BIT10=2000
001000                          BIT9=1000
000400                          BIT8=400
000200                          BIT7=200
000100                          BIT6=100
000040                          BIT5=40
000020                          BIT4=20
000010                          BIT3=10
000004                          BIT2=4
000002                          BIT1=2
000001                          BIT0=1

                                ;PROCESSER LEVELS
000340                          LEVEL7=340
000300                          LEVEL6=300
000240                          LEVEL5=240
000200                          LEVEL4=200
000140                          LEVEL3=140
000100                          LEVEL2=100
000040                          LEVEL1=040
000000                          LEVEL0=000
```

```
                                ;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                          ISYNOO=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
000040                          DNAINTE=BIT5            ;DNA INTR ENAB
000020                          SEND=BIT4               ;SEND
000010                          HDXEN=BIT3              ;HDX/FDX
000001                          BREAK=BIT0              ;BREAK
                                        ;TXCSR WRD DEFINITIONS
000000                          USER=0                  ;USER MODE
004000                          MINT=4000               ;MAINT INT MODE
010000                          MEXT=10000              ;MAINT EXT MODE
014000                          SYSTST=14000            ;SYSTEM TEST MODE
                                        ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
```

```
                                        ;STANDARD INTERRUPT VECTORS


        000024                  .=24
000024  007332                          .PFAIL                  ;POWER FAIL HANDLER
000026  000340                          340                     ;SERVICE AT LEVEL 7
000030  007062                          .HLT                    ;ERROR HANDLER
000032  000340                          340                     ;SERVICE AT LEVEL 7
000034  007030                          .TRPSRV                 ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                          340                     ;SERVICE AT LEVEL 7

                                ;SOFTWARE SWITCH REGISTER

        000174                  .=174
000174  000000                          DISPREG:.WORD 0         ;SOFTWARE DISPLAY REG.
000176  000000                          SWREG:  .WORD 0         ;SOFTWARE SWITCH REGISTER
000200  000167  001054                  JMP     .START          ;GO TO START OF PROGRAM



        001100                  .=1100

                                        ;INDIRECT POINTERS

001100  177570                  SWR:    177570                  ;SWITCH REGISTER POINTER
001102  177570                  LIGHTS:177570                   ;DISPLAY REGISTER POINTER
001104  177560                  TKCSR:  177560                  ;TELETYPE KEYBOARD CONTROL REGISTER
001106  177562                  TKDBR:  177562                  ;TELETYPE KEYBOARD DATA BUFFER
001110  177564                  TPCSR:  177564                  ;TELEPRINTER CONTROL REGISTER
001112  177566                  TPDBR:  177566                  ;TELEPRINTER DATA BUFFER

                                        ;PROGRAM CONTROL PARAMETERS

001114  000000                  RTRN:   0                       ;SCOPE ADDRESS FOR LOOP ON TEST
001116  000000                  NEXT:   0                       ;ADDRESS OF NEXT TEST TO BE EXECUTED
001120  000000                  LOCK:   0                       ;ADDRESS FOR LOCK ON CURRENT DATA
001122  000000                  ICOUNT: 0                       ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
001124  000000                  LPCNT:  0                       ;NUMBER OF ITERATIONS COMPLETED
001126  000000                  TSTNO:  0                       ;NUMBER OF TEST IN PROGRESS
001130  000000                  PCNT:   0                       ;NUMBER OF PASSES COMPLETED
001132  000000                  ECNT:   0                       ;TOTAL NUMBER OF ERRORS
001134  000000                  LSTERR: 0                       ;PC OF LAST ERROR CALL

                                        ;PROGRAM VARIABLES

001136  000020                  HOLD:   20                      ;TEMPORARY STORAGE=DELAY TIME FOR CABLES
001140  000000                  SHIFT:  0                       ;TEMPORARY STORAGE= # OF SHIFTS PER CHAR
001142  000000                  COUNT:  0                       ;TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
001144  000000                  T1:     0                       ;TEMPORARY STORAGE
001146  000000                  T2:     0                       ;TEMPORARY STORAGE
001150  000000                  T3:     0                       ;TEMPORARY STORAGE
001152  000000                  T4:     0                       ;TEMPORARY STORAGE
001154  000000                  T5:     0                       ;TEMPORARY STORAGE
001156  000000                  SAVR0:  0                       ;R0 STORAGE
001160  000000                  SAVR1:  0                       ;R1 STORAGE
001162  000000                  SAVR2:  0                       ;R2 STORAGE
```

```
752  001164  000000          SAVR3:  0          ;R3 STORAGE
753  001166  000000          SAVR4:  0          ;R4 STORAGE
754  001170  000000          SAVR5:  0          ;R5 STORAGE
755  001172  000000          SAVSP:  0          ;STACK POINTER STORAGE
756  001174  000000          SAVPC:  0          ;PROGRAM COUNTER STORAGE
```

```
                                        ;PROGRAM CONVERSATIONAL PARAMETERS
001176    377                   SYNCNO: .BYTE   377          ;# OF SYNC CHARS REQ'D FOR SYNC'ZATION
001177    377                   SEXMIT: .BYTE   377          ;SEC XMIT JUMPER "IN"
001200    377                   SEREC:  .BYTE   377          ;SEC REC JUMPER "IN"
001201    377                   OPTCLR: .BYTE   377          ;OPTIONAL JUMPER CLR "IN"
001202    000                   MULTD:  .BYTE   0            ;NO MULTIPLE DEVICE FLAG
001203    377                   JMRBY:  .BYTE   377          ;EXTERNAL MODEM BYPASS JUMPER "IN"
                                .EVEN

                                        ;PROGRAM MULTIPLE DEVICE PARAMETERS
001204  000000                  PCCFADD:        0            ;PROG CONTROLLED 1ST DEVICE ADDR
001206  000000                  KEEPADD:        0            ;SAVED 1ST DEVICE ADDR
001210  000000                  LASTADD:        0            ;LAST DEVICE RXCSR ADDR
001212  000000                  BASEIV:         0            ;PROG CONTROLLED IV
001214  000000                  KEEPIV:         0            ;SAVED INTR VECTOR
001216  000000                  ACTREG:         0            ;ACTIVE REGISTER ,,MODIFY THIS
                                                             ;LOCATION TO DISQUALIFY OR QUALIFY
                                                             ;DEVICES (1= RUN,,0= DON'T RUN)
001220  000000                  ROTADD:         0            ;ROTATING POINTER FOR ACTREG..POINTS
                                                             ;TO DEVICE PRESENTLY UNDER TEST WHEN RUNNING MULTIPLE DE

                                        ;PROGRAM CONTROL FLAGS

001222    000                   INIFLG: .BYTE   0            ;PROGRAM INITIALIZATION FLAG
001223    000                   STFLG:  .BYTE   0            ;TEST START FLAG
001224    000                   ERRFLG: .BYTE   0            ;ERROR OCCURRED FLAG
001225    000                   LOKFLG: .BYTE   0            ;LOCK ON CURRENT TEST FLAG

                                        ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
                                        ;POINTERS TO SUBROUTINES CAN BE FOUND
                                        ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS

001226                          .TRPTAB:
                                ;#######################################################################
                                ;#######################################################################
        104400                          SCOPE=TRAP+0     ;CALL TO SCOPE LOOP AND ITERATION HANDLER
001226  005614                  .SCOPE
        104401                          SCOP1=TRAP+1     ;CALL TO LOOP ON CURRENT DATA HANDLER
001230  006000                  .SCOP1
        104402                          TYPE=TRAP+2      ;CALL TO TELETYPE OUTPUT ROUTINE
001232  005720                  .TYPE
        104403                          INSTR=TRAP+3     ;CALL TO ASCII STRING INPUT ROUTINE
001234  005060                  .INSTR
        104404                          INSTER=TRAP+4    ;CALL TO INPUT ERROR HANDLER
001236  005176                  .INSTER
        104405                          PARAM=TRAP+5     ;CALL TO NUMERICAL DATA INPUT ROUTINE
001240  006230                  ..PARAM
        104406                          SAVOS=TRAP+6     ;CALL TO REGISTER SAVE ROUTINE
001242  005444                  .SAVOS
        104407                          RESOS=TRAP+7     ;CALL TO REGISTER RESTORE ROUTINE
001244  006504                  .RESOS
        104410                          CONVRT=TRAP+10   ;CALL TO DATA OUTPUT ROUTINE
001246  005536                  .CONVRT
        104411                          CNVRT=TRAP+11    ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF
001250  006542                  .CNVRT
        104412                          SETFLG=TRAP+12   ;CALL TO FLAG SET ROUTINE
```

```
001252  006762                  .SETFLG
        104413                          CKSWR=TRAP+13   ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
001254  007476                  .CKSWR
        104414                          CNTLU=TRAP+14   ;CALL TO ALLOW LOADING  OF SWREG FROM TTY
001256  007552                  .CNTLU
                                ;*******************************************************************
                                ;*******************************************************************

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

001260  012767  000340  176510  .START: MOV     #340,PS                 ;LOCK OUT INTERRUPTS
001266  012706  001100                  MOV     #STACK,SP               ;SET UP STACK
001272  012737  007332  000024          MOV     #.PFAIL,@#24            ;SET UP POWER FAIL VECTOR
001300  005067  177620                  CLR     LPCNT                   ;CLEAR # OF ITERATION COMPLETED LOCATION
001304  105067  177713                  CLRB    STFLG                   ;CLEAR START FLAG
001310  00 67   177614                  CLR     PASCNT                  ;CLEAR PASS COUNT
001314  10 67   177704                  CLRB    ERRFLG                  ;CLEAR ERROR FLAG
001320  005067  177606                  CLR     ERRCNT                  ;CLEAR ERROR COUNT
001324  005067  177604                  CLR     LSTERR                  ;CLEAR LAST ERROR POINTER
001330  012767  000001  177570          MOV     #1,TSTNO                ;SET UP FOR TEST 1
001336  012767  001260  177550          MOV     #.START,RTRN            ;SET UP FOR POWER FAIL BEFORE
                                                                        ;TESTING STARTS
001344  105767  177652                  TSTB    INIFLG                  ;HAS INITIALIZATION BEEN PERFORMED
001350  001004                          BNE     ONCE
001352  104402  007652                  TYPE    ,MTITLE                 ;TYPE TITLE MESSAGE
001356  105167  177640                  COMB    INIFLG                  ;IF NOT SET FLAG AND DO
001362  012767  177570  177510  ONCE:  MOV     #DSWR,SWR               ;RELOAD HARDWARE SWITCH REGISTER INTO POINTER
001370  012767  177570  177504          MOV     #DLIGHTS,LIGHTS         ;RELOAD HARDWARE DISPLAY REGISTER INTO POINTER
001376  013746  000006                  MOV     @#6,-(SP)               ;SAVE VECTORS
001402  013746  000004                  MOV     @#4,-(SP)
001406  012737  001426  000004          MOV     #64$,@#4                ;SET UP FOR TIMEOUT
001414  022777  177777  177456          CMP     #-1,@SWR                ;REFERENCE HARDWARE SWITCH REGISTER
001422  001402                          BEQ     65$
001424  000407                          BR      66$
001426  022626                  64$:    CMP     (SP)+,(SP)+             ;ADJUST STACK
001430  012767  000176  177442  65$:    MOV     #SWREG,SWR              ;POINT TO SOFTWARE SWITCH REG
001436  012767  000174  177436          MOV     #DISREG,LIGHTS          ;POINT TO SOFT DISPLAY REG
001444  012637  000004          66$:    MOV     (SP)+,@#4               ;RESTORE VECTORS
001450  012637  000006                  MOV     (SP)+,@#6
001454  005737  000042                  TST     @#42                    ;UNDER MONITOR
001460  001005                          BNE     67$
001462  022767  000176  177410          CMP     #SWREG,SWR              ;IS SWREG USED
001470  001001                          BNE     67$
001472  104414                          CNTLU
001474  032777  000001  177376  67$:    BIT     #SW00,@SWR              ;RESELECT VECTOR & CONTROL REG?
001502  001002                          BNE     1$
001504  000167  000446                  JMP     .BEGIN
001510  012700  000300          1$:     MOV     #300,R0                 ;RESTORE VECTOR AREA TO TRAPCATCHER
001514  012701  000302                  MOV     #302,R1                 ;START AT LOCATION 300
001520  012702  000004                  MOV     #4,R2
001524  010110                  2$:     MOV     R1,(R0)
```

```
001526  000011                          CLR     (R1)
001530  000000                          ADD     R2,R0
001532  000001                          ADD     R2,R1
001534  001001  001000                  CMP     #1000,R1                ;END AT LOCATION 7776
001540  000771                          BLT     2$
001542  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
001544  007726                          MREGAD                          ;MESSAGE
001546  104405                          PARAM                           ;CONVERT STRING
001550  160000                          160000                          ;LOW LIMIT
001552  167776                          167776                          ;HIGH LIMIT
001554  011562                          DUBASE                          ;STORE AT THIS LOCATION
001556     001                  .BYTE   1                               ;MASK
001557     001                  .BYTE   1                               ;HOW MANY TIMES + 2
001560  016767  007776  177420          MOV     DUBASE,KEEPADD  ;SAVE
001566  004767  007636                  JSR     PC,DUADDR
001572  016767  177410  177404          MOV     KEEPADD,BASEADD ;RESTORE FOR ROTATION
001600  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
001602  007704                          MVECTO                          ;MESSAGE
001604  104405                          PARAM                           ;CONVERT STRING
001606  000300                          300                             ;LOW LIMIT
001610  000776                          776                             ;HIGH LIMIT
001612  012104                          DURIV                           ;STORE AT THIS LOCATION
001614     001                  .BYTE   1                               ;MASK
001615     004                  .BYTE   4                               ;HOW MANY TIMES + 2
001616  016767  010262  177370          MOV     DURIV,KEEPIV    ;SAVE
001624  016767  010254  177360          MOV     DURIV,BASEIV    ;SET UP FOR ROTATION
001632  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
001634  010007                          MMULT                           ;MESSAGE
001636  104412                          SETFLG                          ;SET FLAG BASED UPON INPUT STRING
001640  001202                          MULTD                           ;THIS FLAG
001642  105767  177334                  TSTB    MULTD   ;ARE THERE MULTIPLE DEVICES
                                                        ;ON THE SYSTEM ?
001646  100406                          BMI     BBB     ;YES,ASK NEXT QUESTION
001650  005067  177342                  CLR     ACTREG
001654  005067  177340                  CLR     ROTADD
001660  000167  000140                  JMP     OUTMUL  ;JUMP AROUND NEXT QUESTION
001664                          BBB:
001664  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
001666  010066                          MLASTD                          ;MESSAGE
001670  104405                          PARAM                           ;CONVERT STRING
001672  160000                          160000                          ;LOW LIMIT
001674  167776                          167776                          ;HIGH LIMIT
001676  001210                          LASTADD                         ;STORE AT THIS LOCATION
001700     001                  .BYTE   1                               ;MASK
001701     001                  .BYTE   1                               ;HOW MANY TIMES + 2
                                ;THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
001702  012767  000001  177310  1$:     MOV     #1,ROTADD       ;SET UP POINTER
001710  005067  177302                  CLR     ACTREG  ;CLR ACTIVE REGISTER
001714  005067  177300  177274  2$:     BIS     ROTADD,ACTREG   ;MAKE THIS DEVICE ACTIVE
001722  000241                          CLC
001724  006167  177270                  ROL     ROTADD  ;SET UP POINTER
001730  103421                          BCS     3$      ;ARE YOU OUT OF RANGE ?
001732  062767  000010  177244          ADD     #10,BASEADD     ;SET UP BASE ADDRESS
001740  026767  177244  177236          CMP     LASTADD,BASEADD ;IS THIS THE LAST DEVICE ?
001746  101362                          BHI     2$      ;NO DO IT AGAIN
001750  056767  177244  177240          BIS     ROTADD,ACTREG   ;THIS ASSUMES THAT THERE ARE AT
```

```
                                                        ;LEAST TWO DEVICES WHEN YOU ANSWER YES TO
                                                        ;MULTIPLE DEVICE QUESTION
001756  012767  000001  177234  4$:     MOV     #1,ROTADD       ;SET UP FOR LATER USE IN END OF PASS ROUTINE
001764  016767  177216  177212          MOV     KEEPADD,BASEADD ;DITTO
001772  000414                          BR      OUTMUL  ;CONTINUE QUESTIONS
001774  016767  177206  177202  3$:     MOV     KEEPADD,BASEADD ;RESTORE
0[illegible]72  104403                  INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
0[illegible]4   010251                  MRANGE                          ;MESSAGE
0[illegible]6   10[illegible]05         PARAM                           ;CONVERT STRING
0[illegible]0   1[illegible]0           160000                          ;LOW LIMIT
0[illegible]2   167776                  167776                          ;HIGH LIMIT
0[illegible]14  001210                  LASTADD                         ;STORE AT THIS LOCATION
0[illegible]16  001                     .BYTE   1                       ;MASK
0[illegible]17  001                     .BYTE   1                       ;HOW MANY TIMES + 2
0[illegible]0   000167  177656          JMP     1$      ;DO IT AGAIN
0[illegible]4                           OUTMUL:
0[illegible]4   104403                  INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
0[illegible]6   010535                  MLEVEL                          ;MESSAGE
0[illegible]0   104405                  PARAM                           ;CONVERT STRING
0[illegible]2   000004                  4                               ;LOW LIMIT
0[illegible]4   000007                  7                               ;HIGH LIMIT
0[illegible]6   011424                  DUPRT                           ;STORE AT THIS LOCATION
0[illegible]40  000                     .BYTE   0                       ;MASK
0[illegible]41  001                     .BYTE   1                       ;HOW MANY TIMES + 2
000042  004767  007306                  JSR     PC,DULEV
                                        ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
                                        ;BUFFER TO THE CHARACTERS "1" AND "2".
                                        ;IF THE CHARACTER IS "1" CLEAR THE FLAG
                                        ;IF THE CHARACTER IS "2" SET THE FLAG
000046                          AAA:
000046  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
0[illegible]0   010[illegible]52        MSYNC                           ;MESSAGE
0[illegible]2   12[illegible]67  000061  007134  3$:     CMPB    #'1,INBUF       ;IS IT "1" ?
0[illegible]0   001003                  BNE     1$
0[illegible]2   10[illegible]367  177110          CLRB    SYNCNO          ;000
0[illegible]6   0[illegible]412                 BR      4$
0[illegible]0   1[illegible]767  000062  007116  1$:     CMPB    #'2,INBUF       ;IS IT "2" ?
0[illegible]76  001004                  BNE     2$
0[illegible]100 112767  177777  177070          MOVB    #-1,SYNCNO              ;377
0[illegible]106 0[illegible]402                 BR      4$
0[illegible]110 1[illegible]4                   2$:     INSTER          ;RETRY
0[illegible]112 0[illegible]57                  BR      3$
0[illegible]114 0[illegible]40                  4$:     NOP
0[illegible]2116 10[illegible]403               INSTR                   ;OUTPUT MESSAGE & GET INPUT STRING
0[illegible]2120 010630                         MWIRE6                  ;MESSAGE
0[illegible]2122 104412                         SETFLG                  ;SET FLAG BASED UPON INPUT STRING
002124  001177                          SEXMIT                          ;THIS FLAG
002126  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
002130  010676                          MWIRE5                          ;MESSAGE
002132  104412                          SETFLG                          ;SET FLAG BASED UPON INPUT STRING
002134  001200                          SREC                            ;THIS FLAG
002136  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
002140  010743                          MWIRE4                          ;MESSAGE
002142  104412                          SETFLG                          ;SET FLAG BASED UPON INPUT STRING
002144  001201                          CTCLR                           ;THIS FLAG
002146  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
```

```
002150  011017                          MEXTJ                           ;MESSAGE
002152  104412                          SETFLG                          ;SET FLAG BASED UPON INPUT STRING
002154  001203                          JMRBY                           ;THIS FLAG

                                        ;TEST START AND RESTART

002156  012767  000340  175612  .BEGIN: MOV     #340,PS                 ;LOCK OUT INTERRUPTS
002164  012706  001100                  MOV     #STACK,SP               ;SET UP STACK
002170  005737  000042                  TST     @#42                    ;IS PROGRAM UNDER MONITOR CONTROL
002174  001056                          BNE     3$
002176  105767  177000                  TSTB    MULTD   ;DON'T ALLOW LOCK ON TEST IF RUNNING
                                                        ;MULTIPLE DEVICES
002202  001407                          BEQ     5$                      ;IF NO,TEST FOR LOCK ON TEST
002204  016767  003564  003464          MOV     BRW,TTST                ;RESTORE NORMAL SCOPE LOOP
002212  015767  003560  003460          MOV     BRX,TTST+2              ;DITTO
002220  000444                          BR      3$                      ;JUMP AROUND IF YES
002222  005777  000004  176650  5$:     BIT     #BIT2,@SWR              ;CHECK FOR LOCK ON TEST
002230  001416                          BEQ     1$
002232  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
002234  010472                          MLOCK                           ;MESSAGE
002236  104412                          SETFLG                          ;SET FLAG BASED UPON INPUT STRING
002240  001225                          LOKFLG                          ;THIS FLAG
002242  105767  176757                  TSTB    LOKFLG                  ;IS LOCK ON TEST OPTION SELECTED
002246  001407                          BEQ     1$
002250  012767  000240  003420          MOV     #NOP,TTST
002256  012767  000240  003414          MOV     #NOP,TTST+2             ;SET UP TO LOCK
002264  000436                          BR      2$
002266  016767  003502  003402  1$:     MOV     BRW,TTST
002274  016767  003476  003376          MOV     BRX,TTST+2              ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
002302  005777  000002  176570  2$:     BIT     #SW01,@SWR              ;IF SW01=1, GET STARTING PC
002310  001410                          BEQ     3$
002312  104403                          INSTR                           ;OUTPUT MESSAGE & GET INPUT STRING
002314  010437                          MTSTPC                          ;MESSAGE
002316  104405                          PARAM                           ;CONVERT STRING
002320  000000                          TST1                            ;LOW LIMIT
002322  000004                          TLAST                           ;HIGH LIMIT
002324  001114                          RTRN                            ;STORE AT THIS LOCATION
002326     001                  .BYTE   1                               ;MASK
002327     001                  .BYTE   1                               ;HOW MANY TIMES + 2
002330  000403                          BR      4$
002332  012767  002350  176554  3$:     MOV     #TST1,RTRN              ;START AT TEST 1
002340  104402  010453          4$:     TYPE    ,MR                     ;TYPE R
002344  000177  176344                  JMP     @RTRN                   ;START TESTING
```

```
                                        ;;THIS TEST VERIFYS MATCH DETECT & DATA RDY
                                        ;;FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
                                        ;;BY OBSERVING RECACT BIT
                                        ;;IT WILL TAKE TWO SYNC # CHARACTERS TO GET RECACT BIT
                                        ;;#:    DEPENDENT ON MONITOR .............
                                        ;;IF ONE SYNC STRAP IS SELECTED ,IT WILL
                                        ;;ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
                                        ;;ASSERT
                                        ;;MODE: SYNC INTERNAL
                                        ;;LENGTH: FIVE
                                        ;;SYNC CHARACTER FOR MATCH: B/C
                                        ;;THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
                                        ;;
002350  012767  000001  176550  TST1:   MOV     #1,TSTNO        ;SAVE THIS
002356  012767  002676  176532          MOV     #TST2,NEXT              ;GO TO THIS TEST WHEN THRU
002364  012767  000000  176526          MOV     #3$,LOCK        ;SET UP FOR SCOPE LOOP
002372  052777  000400  007474          BIS     #MRESET,@TXCSR  ;MASTER RESET
002400  016703  007460                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
                                ;SET SYNC INTERNAL,FIVE,NO PARITY,0 SYNC REGISTER
002404  012704  030000                  MOV     #SYNINT!FIVE!NOPAR,R4   ;CREATE PARAMETERS
002410  012777  000100  007456  1$:     MOV     #MINT!SEND,@TXCSR       ;SET SEND & MAINT INTER
002416  010477  007446                  MOV     R4,@PARCSR      ;LOAD CSR
002422  052777  000020  007430          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
                                ;POKE CLK TO GET INTO SYNCRONIZATION
                                ;BOTH THE LOGIC & RECEIVER
002430  052777  020000  007436          BIS     #CLK,@TXCSR     ;POKE CLK UP
002436  042777  020000  007430          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
002444  110477  007430                  MOVB    R4,@TXDBUF      ;LOAD DATA CHARACTER
                                ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCRONIZATION
002450  052777  020000  007416          BIS     #CLK,@TXCSR     ;POKE CLK UP
002456  042777  020000  007410          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
002464  032777  004000  007366          BIT     #RECACT,@RXCSR  ;RECACT ?
002472  001401                          BEQ     2$
002474  104000                          HLT     ;RECACT SHOULD NOT BE SET
002476                          2$:
002476  000404                          BR      4$
002500  010477  007364          3$:     MOV     R4,@PARCSR      ;LOAD PARCSR WITH PARAMETERS
002504  110477  007370                  MOVB    R4,@TXDBUF      ;LOAD SYNC CHAR
002510  012767  000002  176424  4$:     MOV     #2,COUNT        ;# OF SYNC CHARS
002516  005777  007352          5$:     TST     @TXCSR  ;DNA ?
002522  100001                          BPL     6$      ;BR IF NOT SET
002524  104000                          HLT     ;DNA SHOULD NOT BE SET OR....
002526                          6$:
                                        ;IT SHOULD BE CLEARED FROM PREVIOUS READ
002526  012767  000005  176404          MOV     #5,SHIFT        ;# OF SHIFTS
002534                          7$:
002534  052777  020000  007332          BIS     #CLK,@TXCSR     ;POKE CLK UP
002542  042777  020000  007324          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
002550  005367  176364                  DEC     SHIFT   ;# OF SHIFTS
002554  001367                          BNE     7$
002556  005367  176360                  DEC     COUNT   ;# OF SYNC CHARS
002562  001403                          BEQ     8$
                                ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
002564  105767  176406                  TSTB    SYNCNO
002570  100752                          BMI     5$      ;TWO SYNC CHARACTERS..
002572  032777  004000  007260  8$:     BIT     #RECACT,@RXCSR  ;RECACT ?
```

```
002600  001001                          BNE     9$
002602  104000                          HLT             ;RECACT FAILED TO SET,POSSIBLE
002604                          9$:
                                                        ;THAT THE RECEIVER FAILED TO MATCH
                                                        ;THE SYNC CHARACTER
002604  017701  007254                  MOV     @RXDBUF,R1      ;SAVE ACTUAL
002610  010400                          MOV     R4,R0           ;SAVE EXPECTED
002612  042700  177400                  BIC     #177400,R0      ;CLR UPPER BYTE
002616  020001                          CMP     R0,R1   ;DO THEY COMPARE ?
002620  001401                          BEQ     10$
002622  104002                          HLT     2       ;IF RECACT FAILED ALONG WITH THIS
002624                          10$:
                                                        ;...IT PROBABLY IS A TRANSMITTER ERROR
                                                        ;HOWEVER,...IF ONLY THIS FAILED IT
                                                        ;PROBABLY IS A RECEIVER ERROR
002624  104401                          SCOP1
                                ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
                                ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
                                ;TXDBUF
002626  052777  020000  007240          BIS     #CLK,@TXCSR     ;POKE CLK UP
002634  005777  007234                  TST     @TXCSR  ;DNA?
002640  100401                          BMI     11$
002642  104000                          HLT     ;DNA DID NOT ASSERT
002644                          11$:
                                                        ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
002644  052777  000400  007222          BIS     #MRESET,@TXCSR  ;MASTER RESET
002652  032777  000020  007200          BIT     #SYNSCH,@RXCSR  ;SYNC SEARCH = 0 ?
002660  001401                          BEQ     12$
002662  104000                          HLT     ;SYNC SEARCH SHOULD BE NOT SET
002664                          12$:
002664  005204                          INC     R4
002666  122704  000040                  CMPB    #40,R4  ;IS THIS THE LAST CHARACTER ?
002672  001246                          BNE     1$      ;NO
002674  104400                          SCOPE
```

```
                                                ;;THIS TEST VERIFYS MATCH DETECT & DATA RDY
                                                ;;FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
                                                ;;BY OBSERVING RECACT BIT
                                                ;;IT WILL TAKE TWO SYNC # CHARACTERS TO GET RECACT BIT
                                                ;;#:    DEPENDENT ON MONITOR .............
                                                ;;IF ONE SYNC STRAP IS SELECTED ,IT WILL
                                                ;;ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
                                                ;;ASSERT
                                                ;;MODE: SYNC INTERNAL
                                                ;;LENGTH: SIX
                                                ;;SYNC CHARACTER FOR MATCH: B/C
                                                ;;THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
                                                ;;
002676  012767  000002  176222  TST2:   MOV     #2,TSTNO        ;SAVE THIS
002704  012767  003224  176204          MOV     #TST3,NEXT              ;GO TO THIS TEST WHEN THRU
002712  012767  003026  176200          MOV     #3$,LOCK        ;SET UP FOR SCOPE LOOP
002720  052777  000400  007146          BIS     #MRESET,@TXCSR  ;MASTER RESET
002726  016703  007132                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
                                ;SET SYNC INTERNAL,SIX,NO PARITY,0 SYNC REGISTER
002732  012704  032000                  MOV     #SYNINT!SIX!NOPAR,R4    ;CREATE PARAMETERS
002736  012777  004020  007130  1$:     MOV     #MINT!SEND,@TXCSR       ;SET SEND & MAINT INTER
002744  010477  007120                  MOV     R4,@PARCSR      ;LOAD CSR
002750  052777  000020  007102          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
                                ;POKE CLK TO GET INTO SYNCRONIZATION
                                ;BOTH THE LOGIC & RECEIVER
002756  052777  020000  007110          BIS     #CLK,@TXCSR     ;POKE CLK UP
002764  042777  020000  007102          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
002772  110477  007102                  MOVB    R4,@TXDBUF      ;LOAD DATA CHARACTER
                                ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCRONIZATION
002776  052777  020000  007070          BIS     #CLK,@TXCSR     ;POKE CLK UP
003004  042777  020000  007062          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
003012  032777  004000  007040          BIT     #RECACT,@RXCSR  ;RECACT ?
003020  001401                          BEQ     2$
003022  104000                          HLT     ;RECACT SHOULD NOT BE SET
003024                          2$:
003024  000404                          BR      4$
003026  010477  007036          3$:     MOV     R4,@PARCSR      ;LOAD PARCSR WITH PARAMETERS
003032  110477  007042                  MOVB    R4,@TXDBUF      ;LOAD SYNC CHAR
003036  012767  000002  176076  4$:     MOV     #2,COUNT        ;# OF SYNC CHARS
003044  005777  007024          5$:     TST     @TXCSR  ;DNA ?
003050  100001                          BPL     6$      ;BR IF NOT SET
003052  104000                          HLT     ;DNA SHOULD NOT BE SET OR....
003054                          6$:
                                                ;IT SHOULD BE CLEARED FROM PREVIOUS READ
003054  012767  000006  176056          MOV     #6,SHIFT        ;# OF SHIFTS
003062                          7$:
003062  052777  020000  007004          BIS     #CLK,@TXCSR     ;POKE CLK UP
003070  042777  020000  006776          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
003076  005367  176036                  DEC     SHIFT   ;# OF SHIFTS
003102  001367                          BNE     7$
003104  005367  176032                  DEC     COUNT   ;# OF SYNC CHARS
003110  001403                          BEQ     8$
                                ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
003112  105767  176060                  TSTB    SYNCNO
003116  100752                          BMI     5$      ;TWO SYNC CHARACTERS..
003120  032777  004000  006732  8$:     BIT     #RECACT,@RXCSR  ;RECACT ?
```

```
003126  001001                          BNE     9$
003130  104000                          HLT             ;RECACT FAILED TO SET,POSSIBLE
003132                          9$:
                                                        ;THAT THE RECEIVER FAILED TO MATCH
                                                        ;THE SYNC CHARACTER
003132  017701  006726                  MOV     @RXDBUF,R1      ;SAVE ACTUAL
003136  010400                          MOV     R4,R0           ;SAVE EXPECTED
003140  042700  177400                  BIC     #177400,R0      ;CLR UPPER BYTE
003144  020001                          CMP     R0,R1   ;DO THEY COMPARE ?
003146  001401                          BEQ     10$
003150  104002                          HLT     2       ;IF RECACT FAILED ALONG WITH THIS
003152                          10$:
                                                        ;...IT PROBABLY IS A TRANSMITTER ERROR
                                                        ;HOWEVER,...IF ONLY THIS FAILED IT
                                                        ;PROBABLY IS A RECEIVER ERROR
003152  104401                          SCOP1
                                ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
                                ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
                                ;TXDBUF
003154  052777  020000  006712          BIS     #CLK,@TXCSR     ;POKE CLK UP
003162  005777  006706                  TST     @TXCSR  ;DNA?
003166  100401                          BMI     11$
003170  104000                          HLT     ;DNA DID NOT ASSERT
003172                          11$:
                                                ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
003172  052777  000400  006674          BIS     #MRESET,@TXCSR  ;MASTER RESET
003200  032777  000020  006652          BIT     #SYNSCH,@RXCSR  ;SYNC SEARCH = 0 ?
003206  001401                          BEQ     12$
003210  104000                          HLT     ;SYNC SEARCH SHOULD BE NOT SET
003212                          12$:
003212  005204                          INC     R4
003214  122704  000100                  CMPB    #100,R4 ;IS THIS THE LAST CHARACTER ?
003220  001246                          BNE     1$      ;NO
003222  104400                          SCOPE
```

```
                                        ;;THIS TEST VERIFYS MATCH DETECT & DATA RDY
                                        ;;FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
                                        ;;BY OBSERVING RECACT BIT
                                        ;;IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
                                        ;;*:    DEPENDENT ON MONITOR .............
                                        ;;IF ONE SYNC STRAP IS SELECTED ,IT WILL
                                        ;;ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
                                        ;;ASSERT
                                        ;;MODE: SYNC INTERNAL
                                        ;;LENGTH: SEVEN
                                        ;;SYNC CHARACTER FOR MATCH: B/C
                                        ;;THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
                                        ;;
003224  012767  000003  175674  TST3:   MOV     #3,TSTNO        ;SAVE THIS
003232  012767  003552  175656          MOV     #TST4,NEXT              ;GO TO THIS TEST WHEN THRU
003240  012767  003354  175652          MOV     #3$,LOCK        ;SET UP FOR SCOPE LOOP
003246  052777  000400  006620          BIS     #MRESET,@TXCSR  ;MASTER RESET
003254  016703  006604                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
                                ;SET SYNC INTERNAL,SEVEN,NO PARITY,0 SYNC REGISTER
003260  012704  034000                  MOV     #SYNINT!SEVEN!NOPAR,R4  ;CREATE PARAMETERS
003264  012777  004020  006602  1$:     MOV     #MINT!SEND,@TXCSR       ;SET SEND & MAINT INTER
003272  010477  006572                  MOV     R4,@PARCSR      ;LOAD CSR
003276  052777  000020  006554          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
                                ;POKE CLK TO GET INTO SYNCRONIZATION
                                ;BOTH THE LOGIC & RECEIVER
003304  052777  020000  006562          BIS     #CLK,@TXCSR     ;POKE CLK UP
003312  042777  020000  006554          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
003320  110477  006554                  MOVB    R4,@TXDBUF      ;LOAD DATA CHARACTER
                                ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCRONIZATION
003324  052777  020000  006542          BIS     #CLK,@TXCSR     ;POKE CLK UP
003332  042777  020000  006534          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
003340  032777  004000  006512          BIT     #RECACT,@RXCSR  ;RECACT ?
003346  001401                          BEQ     2$
003350  104000                          HLT     ;RECACT SHOULD NOT BE SET
003352                          2$:
003352  000404                          BR      4$
003354  010477  006510          3$:     MOV     R4,@PARCSR      ;LOAD PARCSR WITH PARAMETERS
003360  110477  006514                  MOVB    R4,@TXDBUF      ;LOAD SYNC CHAR
003364  012767  000002  175550  4$:     MOV     #2,COUNT        ;# OF SYNC CHARS
003372  005777  006476          5$:     TST     @TXCSR  ;DNA ?
003376  100001                          BPL     6$      ;BR IF NOT SET
003400  104000                          HLT     ;DNA SHOULD NOT BE SET OR....
003402                          6$:
                                        ;IT SHOULD BE CLEARED FROM PREVIOUS READ
003402  012767  000007  175530          MOV     #7,SHIFT        ;# OF SHIFTS
003410                          7$:
003410  052777  020000  006456          BIS     #CLK,@TXCSR     ;POKE CLK UP
003416  042777  020000  006450          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
003424  005367  175510                  DEC     SHIFT   ;# OF SHIFTS
003430  001367                          BNE     7$
003432  005367  175504                  DEC     COUNT   ;# OF SYNC CHARS
003436  001403                          BEQ     8$
                                ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
003440  105767  175532                  TSTB    SYNCNO
003444  100152                          BMI     5$      ;TWO SYNC CHARACTERS..
003446  032777  004000  006404  8$:     BIT     #RECACT,@RXCSR  ;RECACT ?
```

```
003454  001001                          BNE     9$
003456  104000                          HLT             ;RECACT FAILED TO SET,POSSIBLE
003460                          9$:
                                                        ;THAT THE RECEIVER FAILED TO MATCH
                                                        ;THE SYNC CHARACTER
003460  017701  006400                  MOV     @RXDBUF,R1      ;SAVE ACTUAL
003464  010400                          MOV     R4,R0           ;SAVE EXPECTED
003466  042700  177400                  BIC     #177400,R0      ;CLR UPPER BYTE
003472  020001                          CMP     R0,R1   ;DO THEY COMPARE ?
003474  001401                          BEQ     10$
003476  104002                          HLT     2       ;IF RECACT FAILED ALONG WITH THIS
003500                          10$:
                                                        ;...IT PROBABLY IS A TRANSMITTER ERROR
                                                        ;HOWEVER,...IF ONLY THIS FAILED IT
                                                        ;PROBABLY IS A RECEIVER ERROR
003500  104401                          SCOP1
                                ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
                                ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
                                ;TXDBUF
003502  052777  020000  006364          BIS     #CLK,@TXCSR     ;POKE CLK UP
003510  005777  006360                  TST     @TXCSR  ;DNA?
003514  100401                          BMI     11$
003516  104000                          HLT     ;DNA DID NOT ASSERT
003520                          11$:
                                                ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
003520  052777  000400  006346          BIS     #MRESET,@TXCSR  ;MASTER RESET
003526  032777  000020  006324          BIT     #SYNSCH,@RXCSR  ;SYNC SEARCH = 0 ?
003534  001401                          BEQ     12$
003536  104000                          HLT     ;SYNC SEARCH SHOULD BE NOT SET
003540                          12$:
003540  005204                          INC     R4
003542  122704  000200                  CMPB    #200,R4 ;IS THIS THE LAST CHARACTER ?
003546  001246                          BNE     1$      ;NO
003550  104400                          SCOPE
```

```
                                                ;;THIS TEST VERIFYS MATCH DETECT & DATA RDY
                                                ;;FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
                                                ;;BY OBSERVING RECACT BIT
                                                ;;IT WILL TAKE TWO SYNC # CHARACTERS TO GET RECACT BIT
                                                ;;#:    DEPENDENT ON MONITOR ............
                                                ;;IF ONE SYNC STRAP IS SELECTED IT WILL
                                                ;;ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
                                                ;;ASSERT
                                                ;;MODE: SYNC INTERNAL
                                                ;;LENGTH: EIGHT
                                                ;;SYNC CHARACTER FOR MATCH: B/C
                                                ;;THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
                                                ;;
003552  012767  000004  175346  TST4:   MOV     #4,TSTNO        ;SAVE THIS
003560  012767  004100  175330          MOV     #TST5,NEXT              ;GO TO THIS TEST WHEN THRU
003566  012767  003702  175324          MOV     #3$,LOCK        ;SET UP FOR SCOPE LOOP
003574  052777  000400  006272          BIS     #MRSET,@TXCSR   ;MASTER RESET
003602  016703  006256                  MOV     RXBUF,R3        ;SET UP FOR ERROR MESSAGE
                                ;SET SYNC INTERNAL,EIGHT,NO PARITY,0 SYNC REGISTER
003606  012704  036000                  MOV     #SYNINT!EIGHT!NOPAR,R4  ;CREATE PARAMETERS
003612  012777  004020  006254  1$:     MOV     #MINT!SEND,@TXCSR       ;SET SEND & MAINT INTER
003620  010477  006244                  MOV     R4,@PARCSR      ;LOAD CSR
003624  052777  000020  006226          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
                                ;POKE CLK TO GET INTO SYNCRONIZATION
                                ;BOTH THE LOGIC & RECEIVER
003632  052777  020000  006234          BIS     #CLK,@TXCSR     ;POKE CLK UP
003640  042777  020000  006226          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
003646  110477  006226                  MOVB    R4,@TXCBUF      ;LOAD DATA CHARACTER
                                ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCRONIZATION
003652  052777  020000  006214          BIS     #CLK,@TXCSR     ;POKE CLK UP
003660  042777  020000  006206          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
003666  032777  004000  006164          BIT     #RECACT,@RXCSR  ;RECACT ?
003674  001401                          BEQ     2$
003676  104000                          HLT     ;RECACT SHOULD NOT BE SET
003700                          2$:
003700  000404                          BR      4$
003702  010477  006162          3$:     MOV     R4,@PARCSR      ;LOAD PARCSR WITH PARAMETERS
003706  110477  006166                  MOVB    R4,@TXCBUF      ;LOAD SYNC CHAR
003712  012767  000002  175222  4$:     MOV     #2,COUNT        ;# OF SYNC CHARS
003720  005777  006150          5$:     TST     @TXCSR  ;DNA ?
003724  100001                          BPL     6$      ;BR IF NOT SET
003726  104000                          HLT     ;DNA SHOULD NOT BE SET OR....
003730                          6$:
                                                ;IT SHOULD BE CLEARED FROM PREVIOUS READ
003730  012767  000010  175202          MOV     #8.,SHIFT       ;# OF SHIFTS
003736                          7$:
003736  052777  020000  006130          BIS     #CLK,@TXCSR     ;POKE CLK UP
003744  042777  020000  006122          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
003752  005367  175162                  DEC     SHIFT   ;# OF SHIFTS
003756  001367                          BNE     7$
003760  005367  175156                  DEC     COUNT   ;# OF SYNC CHARS
003764  001403                          BEQ     8$
                                ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
003766  105767  175204                  TSTB    SYNCNO
003772  100752                          BMI     5$      ;TWO SYNC CHARACTERS..
003774  032777  004000  006056  8$:     BIT     #RECACT,@RXCSR  ;RECACT ?
```

```
004002  001001                          BNE     9$
004004  104000                          HLT             ;RECACT FAILED TO SET,POSSIBLE
004006                          9$:
                                                        ;THAT THE RECEIVER FAILED TO MATCH
                                                        ;THE SYNC CHARACTER
004006  017701  006052                  MOV     @RXDBUF,R1      ;SAVE ACTUAL
004012  010400                          MOV     R4,R0           ;SAVE EXPECTED
004014  042700  177400                  BIC     #177400,R0      ;CLR UPPER BYTE
004020  020001                          CMP     R0,R1   ;DO THEY COMPARE ?
004022  001401                          BEQ     10$
004024  104002                          HLT     2               ;IF RECACT FAILED ALONG WITH THIS
004026                          10$:
                                                        ;...IT PROBABLY IS A TRANSMITTER ERROR
                                                        ;HOWEVER,...IF ONLY THIS FAILED IT
                                                        ;PROBABLY IS A RECEIVER ERROR
004028  104401                          SCOPI
                                ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
                                ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
                                ;TXDBUF
004030  052777  020000  006036          BIS     #CLK,@TXCSR     ;POKE CLK UP
004036  005777  006032                  TST     @TXCSR  ;DNA?
004042  100401                          BMI     11$
004044  104000                          HLT     ;DNA DID NOT ASSERT
004046                          11$:
                                                        ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
004046  052777  000400  006020          BIS     #RESET,@TXCSR   ;MASTER RESET
004054  005777  000020  005776          BIT     #SYNSCH,@RXCSR  ;SYNC SEARCH = 0 ?
004062  001401                          BEQ     12$
004064  104000                          HLT     ;SYNC SEARCH SHOULD BE NOT SET
004066                          12$:
004066  005204                          INC     R4
004070  122704  000000                  CMPB    #0,R4   ;IS THIS THE LAST CHARACTER ?
004074  001246                          BNE     1$      ;NO
004076  104400                          SCOPE
```

```
                                                ;;THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
                                                ;;BOTH THE TRANSMITTER AND RECEIVER LOGIC
                                                ;;MODE:SYNC EXTERNAL (SYNEXT)
                                                ;;LENGTH:EIGHT PLUS PARITY
                                                ;;PARITY:EVEPAR
                                                ;;MAINT. MODE:MINT
                                                ;;
004100  012767  000005  175020  TSTS:   MOV     #5,TSTNO        ;SAVE THIS
004106  012767  004312  175002          MOV     #T26,NEXT               ;GO TO THIS TEST WHEN THRU
004114  012777  000400  005752          BIS     #RESET,@TXCSR   ;MASTER RESET
004122  012777  000000  005740          MOV     #SYNEXT,@PARCSR ;SET THE MODE
004130  052777  000400  005736          BIS     #MRESET,@TXCSR  ;MASTER RESET

                                ;SET MAINTENANCE MODE & SEND
                                                ;NOTE:BIT WIDOW,CLK ARE CLEARED (MTDATA=0)
004136  012777  004020  005730          MOV     #MINT!SEND,@TXCSR

                                ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
004144  012777  027426  005716          MOV     #SYNEXT!EIGHT!EVEPAR!26,@PARCSR
004152  016703  005706                  MOV     RXDBUF,R3       ;SETUP FOR ERROR MSG
004156  005004                          CLR     R4      ;FOR DATA CHAR CREATION
004160  110477  005714                  MOVB    R4,@TXDBUF      ;LOAD CHARACTER
004164  052777  000020  005666          BIS     #SYNCH,@RXCSR   ;SET SEARCH SYNC
                                ;GET INTO SYNCRONIZATION
004172  052777  020000  005674          BIS     #CLK,@TXCSR     ;POKE CLK UP
004200  042777  020000  005656          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
004206  012767  000011  174724  1$:     MOV     #9.,SHIFT       ;# OF SHIFTS
004214  010400                          MOV     R4,R0   ;EXPECTED
004216                          2$:
004216  052777  020000  005650          BIS     #CLK,@TXCSR     ;POKE CLK UP
004224  042777  020000  005642          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
004232  005367  174702                  DEC     SHIFT   ;# OF SHIFTS
004236  022767  000003  174674          CMP     #3,SHIFT        ;TIME TO LOAD NEXT CHAR ?
004244  001003                          BNE     3$      ;NO ?
004246  005204                          INC     R4      ;GENERATE NEXT CHAR
004250  110477  005624                  MOVB    R4,@TXDBUF      ;LOAD NEXT CHARACTER
004254  005767  174660          3$:     TST     SHIFT   ;IS IT 0 ?
004260  001356                          BNE     2$
004262  105777  005572                  TSTB    @TXCSR  ;RXDONE = 1 ?
004266  100401                          BMI     5$
004270  104000                          HLT     ;RXDONE SHOULD BE SET
004272                          5$:
004272  017701  005566                  MOV     @RXDBUF,R1      ;ACTUAL
004276  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
004300  001401                          BEQ     6$
004302  104002                          HLT     2       ;CHARACTERS SHOULD COMPARE
004304                          6$:
004304  105704                          TSTB    R4      ;LAST CHARACTER ?
004306  001337                          BNE     1$      ;NO
004310  104400                  4$:     SCOPE
```

```
                                                ;;THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
                                                ;;BOTH THE TRANSMITTER AND RECEIVER LOGIC
                                                ;;MODE:SYNC EXTERNAL (SYNEXT)
                                                ;;LENGTH:EIGHT PLUS PARITY
                                                ;;PARITY:ODDPAR
                                                ;;MAINT. MODE:MINT
                                                ;;
004312  012767  000006  174606  TST6:   MOV     #6,TSTNO        ;SAVE THIS
004320  012767  004524  174570          MOV     #TST7,NEXT              ;GO TO THIS TEST WHEN THRU
004336  052777  000400  005540          BIS     #MRESET,@TXCSR  ;MASTER RESET
004334  012777  020000  005526          MOV     #SYNEXT,@PARCSR ;SET THE MODE
004342  052777  000400  005524          BIS     #MRESET,@TXCSR  ;MASTER RESET

                                        ;SET MAINTENANCE MODE & SEND
                                                ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
004350  012777  004020  005516          MOV     #MINT!SEND,@TXCSR

                                        ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
004356  012777  027026  005504          MOV     #SYNEXT!EIGHT!ODDPAR!26,@PARCSR
004364  016703  005474                  MOV     RXDBUF,R3       ;SETUP FOR ERROR MSG
004370  005004                          CLR     R4      ;FOR DATA CHAR CREATION
004372  110477  005502                  MOVB    R4,@TXDBUF      ;LOAD CHARACTER
004376  052777  000020  005454          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
                                        ;GET INTO SYNCRONIZATION
004404  052777  020000  005462          BIS     #CLK,@TXCSR     ;POKE CLK UP
004412  042777  020000  005454          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
004420  012767  000011  174512  1$:     MOV     #9.,SHIFT       ;# OF SHIFTS
004426  010400                          MOV     R4,R0   ;EXPECTED
004430                          2$:
004430  052777  020000  005436          BIS     #CLK,@TXCSR     ;POKE CLK UP
004436  042777  020000  005430          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
004444  005367  174470                  DEC     SHIFT   ;# OF SHIFTS
004450  022767  000003  174462          CMP     #3,SHIFT        ;TIME TO LOAD NEXT CHAR ?
004456  001003                          BNE     3$      ;NO ?
004460  005204                          INC     R4      ;GENERATE NEXT CHAR
004462  110477  005412                  MOVB    R4,@TXDBUF      ;LOAD NEXT CHARACTER
004466  005767  174446          3$:     TST     SHIFT   ;IS IT 0 ?
004472  001356                          BNE     2$
004474  105777  005360                  TSTB    @RXCSR  ;RXDONE = 1 ?
004500  100401                          BMI     5$
004502  104000                          HLT     ;RXDONE SHOULD BE SET
004504                          5$:
004504  017701  005354                  MOV     @RXDBUF,R1      ;ACTUAL
004510  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
004512  001401                          BEQ     6$
004514  104002                          HLT     2       ;CHARACTERS SHOULD COMPARE
004516                          6$:
004516  105704                          TSTB    R4      ;LAST CHARACTER ?
004520  001337                          BNE     1$      ;NO
004522  104400                  4$:     SCOPE
```

```
                                                ;;THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
                                                ;;BOTH THE TRANSMITTER AND RECEIVER LOGIC
                                                ;;MODE:SYNC EXTERNAL (SYNEXT)
                                                ;;LENGTH:EIGHT PLUS PARITY
                                                ;;PARITY:EVEPAR
                                                ;;MAINT. MODE:MEXT
                                                ;;
004524  012767  000007  174374  TST7:   MOV     #7,TSTNO        ;SAVE THIS
004532  012767  000004  174356          MOV     #TST8,NEXT              ;GO TO THIS TEST WHEN THRU
004540  105767  174437                  TSTB    JMRBY   ;JUMP AROUND TEST ?
004544  100116                          BPL     4$      ;YES ?
004546  052777  000400  005320          BIS     #MRESET,@TXCSR  ;MASTER RESET
004554  012777  020000  005306          MOV     #SYNEXT,@PARCSR ;SET THE MODE
004562  052777  000400  005304          BIS     #MRESET,@TXCSR  ;MASTER RESET

                                ;SET MAINTENANCE MODE & SEND
                                        ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
004570  012777  010020  005276          MOV     #MEXT!SEND,@TXCSR

                                ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
004576  012777  027426  005264          MOV     #SYNEXT!EIGHT!EVEPAR!26,@PARCSR
004604  016703  005254                  MOV     RXDBUF,R3       ;SETUP FOR ERROR MSG
004610  005004                          CLR     R4      ;FOR DATA CHAR CREATION
004612  110477  005262                  MOVB    R4,@TXDBUF      ;LOAD CHARACTER
004616  052777  000020  005234          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
                                ;GET INTO SYNCRONIZATION
004624  052777  020000  005242          BIS     #CLK,@TXCSR     ;POKE CLK UP
                                        ;WAIT FOR CABLE & DRIVER DELAYS
004632  016702  174300                  MOV     HOLD,R2 ;WAIT THIS AMT
004636                          64$:
004636  005302                          DEC     R2      ;WAIT
004640  001376                          BNE     64$
                                        ;EXIT...
004642  042777  020000  005224          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
                                        ;WAIT FOR CABLE & DRIVER DELAYS
004650  016702  174262                  MOV     HOLD,R2 ;WAIT THIS AMT
004654                          65$:
004654  005302                          DEC     R2      ;WAIT
004656  001376                          BNE     65$
                                        ;EXIT...
004660  012767  000011  174252  1$:     MOV     #9.,SHIFT       ;# OF SHIFTS
004666  010400                          MOV     R4,R0   ;EXPECTED
004670                          2$:
004670  052777  020000  005176          BIS     #CLK,@TXCSR     ;POKE CLK UP
                                        ;WAIT FOR CABLE & DRIVER DELAYS
004676  016702  174234                  MOV     HOLD,R2 ;WAIT THIS AMT
004702                          66$:
004702  005302                          DEC     R2      ;WAIT
004704  001376                          BNE     66$
                                        ;EXIT...
004706  042777  020000  005160          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
                                        ;WAIT FOR CABLE & DRIVER DELAYS
004714  016702  174216                  MOV     HOLD,R2 ;WAIT THIS AMT
004720                          67$:
004720  005302                          DEC     R2      ;WAIT
004722  001376                          BNE     67$
```

```
                                                ;EXIT...
004724  005367  174210                          DEC     SHIFT   ;# OF SHIFTS
004730  022767  000003  174202                  CMP     #3,SHIFT        ;TIME TO LOAD NEXT CHAR ?
004736  [illegible]                             BNE     3$      ;NO ?
004740  [illegible]                             INC     R4      ;GENERATE NEXT CHAR
004742  110477  005132                          MOVB    R4,@TXDBUF      ;LOAD NEXT CHARACTER
004746  005767  174166          3$:             TST     SHIFT   ;IS IT 0 ?
004752  001345                                  BNE     2$
004754  105777  005100                          TSTB    @RXCSR  ;RXDONE = 1 ?
004760  100401                                  BMI     5$
004762  104000                                  HLT     ;RXDONE SHOULD BE SET
004764                          5$:
004764  017701  005074                          MOV     @RXDBUF,R1      ;ACTUAL
004770  [illegible]                             CMP     R0,R1   ;COMPARE EXP VS ACT
004772  001401                                  BEQ     6$
004774  104002                                  HLT     2       ;CHARACTERS SHOULD COMPARE
004776                          6$:
                                                ;CHECK OUT MODEM BYPASS JUMPER
004776  105704                                  TSTB    R4      ;LAST CHARACTER ?
005000  001327                                  BNE     1$      ;NO
005002  104400                  4$:             SCOPE
```

```
                                        ;;THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
                                        ;;BOTH THE TRANSMITTER AND RECEIVER LOGIC
                                        ;;MODE:SYNC EXTERNAL (SYNEXT)
                                        ;;LENGTH:EIGHT PLUS PARITY
                                        ;;PARITY:ODDPAR
                                        ;;MAINT. MODE:MEXT
                                        ;;
005004  012767  000010  174114  TSTB:   MOV     #8,TSTNO        ;SAVE THIS
005012  012767  005264  174076          MOV     #.FOP,NEXT              ;GO TO THIS TEST WHEN THRU
0[illegible]20  105767  174157          TSTB    JMPBY   ;JUMP AROUND TEST ?
0[illegible]24  100116                  BPL     4$      ;YES ?
0[illegible]26  052777  000400  005040  BIS     #MRESET,@TXCSR  ;MASTER RESET
005034  012777  020000  005026          MOV     #SYNEXT,@PARCSR ;SET THE MODE
005042  052777  000400  005024          BIS     #MRESET,@TXCSR  ;MASTER RESET

                                ;SET MAINTENANCE MODE & SEND
                                        ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
005050  012777  010020  005016          MOV     #MEXT!SEND,@TXCSR

                                ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
005056  012777  027026  005004          MOV     #SYNEXT!EIGHT!ODDPAR!26,@PARCSR
0[illegible]4   016703  004774          MOV     RXDBUF,R3       ;SETUP FOR ERROR MSG
0[illegible]10  000004                  CLR     R4      ;FOR DATA CHAR CREATION
0[illegible]2   110477  005002          MOVB    R4,@TXDBUF      ;LOAD CHARACTER
005076  052777  000020  004754          BIS     #SYNCCH,@RXCSR  ;SET SEARCH SYNC
                                ;GET INTO SYNCRONIZATION
005104  052777  020000  004762          BIS     #CLK,@TXCSR     ;POKE CLK UP
                                        ;WAIT FOR CABLE & DRIVER DELAYS
005112  016702  174020                  MOV     HOLD,R2 ;WAIT THIS AMT
005116                          64$:
005116  005302                          DEC     R2      ;WAIT
005120  001376                          BNE     64$
                                        ;EXIT...
005122  042777  020000  004744          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
                                        ;WAIT FOR CABLE & DRIVER DELAYS
005130  016702  174002                  MOV     HOLD,R2 ;WAIT THIS AMT
005134                          65$:
005134  005302                          DEC     R2      ;WAIT
005136  001376                          BNE     65$
                                        ;EXIT...
005140  012767  000011  173772  1$:     MOV     #9.,SHIFT       ;# OF SHIFTS
005146  010400                          MOV     R4,R0   ;EXPECTED
005150                          2$:
005150  052777  020000  004716          BIS     #CLK,@TXCSR     ;POKE CLK UP
                                        ;WAIT FOR CABLE & DRIVER DELAYS
005156  016702  173754                  MOV     HOLD,R2 ;WAIT THIS AMT
005162                          66$:
005162  005302                          DEC     R2      ;WAIT
005164  001376                          BNE     66$
                                        ;EXIT...
005166  042777  020000  004700          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
                                        ;WAIT FOR CABLE & DRIVER DELAYS
005174  016702  173736                  MOV     HOLD,R2 ;WAIT THIS AMT
0[illegible]00                  67$:
0[illegible]0   005302                  DEC     R2      ;WAIT
0[illegible]02  001376                  BNE     67$
```

```
                                        ;EXIT...
005204  005367  173730                  DEC     SHIFT   ;# OF SHIFTS
0[illegible]210 022767  000003  173722  CMP     #3,SHIFT        ;TIME TO LOAD NEXT CHAR ?
0[illegible]216 001003                  BNE     3$      ;NO ?
[illegible]     [illegible]             INC     R4      ;GENERATE NEXT CHAR
0[illegible]2   110477  004652          MOVB    R4,@TXDBUF      ;LOAD NEXT CHARACTER
0[illegible]6   005767  173706  3$:     TST     SHIFT   ;IS IT 0 ?
0[illegible]2   001346                  BNE     2$
0[illegible]34  105777  004620          TSTB    @RXCSR  ;RXDONE = 1 ?
0[illegible]40  100401                  BMI     5$
0[illegible]242 104000                  HLT     ;RXDONE SHOULD BE SET
0[illegible]244                 5$:
0[illegible]244 017701  004614          MOV     @RXDBUF,R1      ;ACTUAL
0[illegible]50  020001                  CMP     R0,R1   ;COMPARE EXP VS ACT
0[illegible]2   001401                  BEQ     6$
0[illegible]4   104002                  HLT     2       ;CHARACTERS SHOULD COMPARE
005256                          6$:
                                        ;CHECK OUT MODEM BYPASS JUMPER
005256  105704                          TSTB    R4      ;LAST CHARACTER ?
0[illegible]60  001327                  BNE     1$      ;NO
005262  104400                  4$:     SCOPE
```

```
                                                ;END OF PASS
                                                ;TYPE NAME OF TEST
                                                ;UPDATE PASS COUNT
                                                ;CHECK FOR EXIT TO ACT-11
                                                ;RESTART TEST

005264  104402                  .EOP:   TYPE                            ;TYPE NAME OF TEST
005266  010426                          MEPASS
005270  104410  005522                  CONVRT  ,OUTCRY
005274  104402  010147                  TYPE    ,DEVICE
005300  105767  173676                  TSTB    MULTD                   ;ARE YOU RUNNING MULTIPLE DEVICES ?
005304  001511                          BEQ     CCC                     ;NO,JUMP AROUND
005306  005767  173704                  TST     ACTREG                  ;ARE ANY DEVICES ACTIVE ?
005312  001007                          BNE     RUNIT                   ;YES
005314  104402  010161                  TYPE    ,MCOW                   ;NO
005320  016700  173672                  MOV     ACTREG,R0               ;DISPLAY ACTREG
005324  000000                          HALT                            ;SELECT SOMETHING TO RUN @ ACTREG:
                                        ;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
005326  000167  173726                  JMP     .START                  ;START OVER AGAIN.....YOU DESELECTED EVERYTHING
005332  012767  000010  173644  RUNIT:  ADD     #10,BASEADD             ;NEXT BLOCK (ADDRESSES)
005340  012767  000010  173644  ZERO:   ADD     #10,BASEIV              ;NEXT BLOCK (VECTORS)
005346  000241                          CLC
005350  006167  173644                  ROL     ROTADD                  ;UP DATE ROTATING POINTER
005354  103410                          BCS     2$                      ;IS IT THE LAST DEVICE
                                                                        ;TO BE TESTED IN THIS PASS ?
005356  036767  173636  173632          BIT     ROTADD,ACTREG           ;TEST THIS DEVICE FOR ACTIVE STATUS
005364  001762                          BEQ     RUNIT                   ;IF NOT ACTIVE, TRY NEXT ADDRESS
005366  004767  000034                  JSR     PC,REPLAY               ;CALCULATE NEW PARAMETERS
005372  000167  000174                  JMP     RESTRT                  ;YES IT WAS ACTIVE,TEST THIS DEVICE
005376  012767  000001  173614  2$:     MOV     #1,ROTADD               ;OK!,NOW SET UP  ROTATING
                                                                        ;POINTER FOR NEXT MULTIPLE PASS
005404  016767  173576  173572          MOV     KEEPADD,BASEADD         ;RESTORE BASE ADDRESS
005412  016767  173576  173572          MOV     KEEPIV,BASEIV           ;RESTORE BASE INTERRUPT VECTORS
005420  004767  000002                  JSR     PC,REPLAY               ;CALC NEW PARAMETERS
005424  000441                          BR      CCC                     ;JUMP AROUND REPLAY
005426  016767  173552  004126  REPLAY: MOV     BASEADD,DUBASE          ;SET UP FOR NEW ADDRESSES
005434  004767  003770                  JSR     PC,DUADDR               ;CREATE NEW ADDRESSES
005440  016767  173546  004136          MOV     BASEIV,DURIV            ;CREATE DURIV
005446  062767  000002  173536          ADD     #2,BASEIV
005454  016767  173532  004134          MOV     BASEIV,DURIS            ;CREATE DURIS
005462  062767  000002  173522          ADD     #2,BASEIV
005470  016767  173516  004112          MOV     BASEIV,DUTIV            ;CREATE DUTIV
005476  062767  000002  173506          ADD     #2,BASEIV
005504  016767  173502  004100          MOV     BASEIV,DUTIS            ;CREATE DUTIS
005512  016767  004366  173472          MOV     DURIV,BASEIV            ;RESTORE
005520  000207                          RTS     PC

005522  000001                  OUTCRY: 1
005524     006     002                  .BYTE   6,2
005526  012060                          RXCSR

005530                          CCC:
005530  005067  173400                  CLR     LSTERR                  ;CLEAR LAST ERROR PC
005534  005067  173464                  CLR     ERRFLG                  ;CLEAR ERROR FLAG
005540  005267  173364                  INC     PASCNT                  ;UPDATE PASS COUNT
```

```
005544  016777  173360  173330          MOV     PASCNT,@LIGHTS          ;DISPLAY PASS COUNT
005552  013701  000042                  MOV     @#42,R1                 ;CHECK FOR ACT-11 OR DOP
005556  001405                          BEQ     RESTRT                  ;IF NOT, CONTINUE TESTING
005560  000005                          RESET
005562  004711                  LOGICAL:        JSR     PC,(R1)
005564  000240                          NOP
005566  000240                          NOP
005570  000240                          NOP
005572  012767  000340  172176  RESTRT: MOV     #340,PS                 ;PREVENT INTERRUPTS (PRIO: 7)
005600  104413                          CKSWR                           ;CHECK FOR ↑G
005602  012767  002350  173304          MOV     #TST1,RTRN
005610  000167  174534                  JMP     TST1

                                        ;SCOPE LOOP AND INTERATION HANDLER

005614                          .SCOPE:
                                ;**** START OF CODE FOR THE X OR TESTER ******
005614  000424                          BR      4$                      ;IF RUNNING ON THE X OR TESTER CHANGE
                                                                        ;THIS INSTRUCTION TO A "NOP"(NOP=240)
005616  013746  000004                  MOV     @#4,-(SP)               ;SAVE CONTENTS OF ERROR VECTOR
005622  012737  005642  000004          MOV     #1$,@#4                 ;SET FOR TIME OUT
005630  005737  177060                  TST     @#177060                ;TIME OUT ON  X OR ?
005634  012637  000004                  MOV     (SP)+,@#4               ;RESTORE ERROR VECTOR
005640  000404                          BR      2$                      ;GO TO NEXT TEST
005642  022626                  1$:     CMP     (SP)+,(SP)+             ;CLEAR THE STACK AFTER A TIMEOUT
005644  012637  000004                  MOV     (SP)+,@#4               ;RESTORE ERROR VECTOR
005650  000403                          BR      3$                      ;LOOP ON PRESENT TEST
005652  016767  173240  173234  2$:     MOV     NEXT,RTRN               ;SET UP NEXT TEST IN RTRN
005660  016716  173230          3$:     MOV     RTRN,(SP)               ;SET UP STACK FOR RTI
005664  000002                          RTI
005666                          4$:     ;**** END OF CODE FOR THE  X OR TESTER ******
005666  104413                          CKSWR                           ;CHECK FOR ↑G
005670  032777  040000  173202          BIT     #SW14,@SWR              ;LOOP ON CURRENT TEST ?
005676  001407                  TTST:   BEQ     1$
005700  000432                          BR      3$
005702  105777  173176                  TSTB    @TKCSR                  ;TEST TTY FLAG
005706  100027                          BPL     3$
005710  017700  173172                  MOV     @TKDBR,R0               ;CLR DONE BIT
005714  000412                          BR      2$                      ;IF A TTY KEY IS STRUCK GO TO NEXT TST
005716  032777  004000  173154  1$:     BIT     #SW11,@SWR              ;INHIBIT ITERATIONS ?
005724  001006                          BNE     2$
005726  005267  173172                  INC     LPCNT
005732  026767  173166  173162          CMP     LPCNT,ICOUNT            ;CHECK FOR ITERATION CNT FINISH
005740  101412                          BLOS    3$
005742  105067  173256          2$:     CLRB    ERRFLG
005746  005067  173152                  CLR     LPCNT
005752  012767  000005  173142          MOV     #5,ICOUNT               ;SET UP ITERATION COUNT
005760  016767  173132  173126          MOV     NEXT,RTRN               ;SET UP NEXT TEST IN RTRN
005766  016716  173122          3$:     MOV     RTRN,(SP)               ;SET UP STACK FOR RTI
005772  000002                          RTI
005774  001407                  BRW:    1407                            ;RESTORE "BEQ 1$" INSTRUCTION
005776  000432                  BRX:    432                             ;RESTORE "BR 3$" INSTRUCTION

                                        ;CHECK FOR FREEZE ON CURRENT DATA

006000  104413                  .SCOP1: CKSWR                           ;CHECK  FOR ↑G
```

```
006002  032777  001000  173070          BIT     #SW09,@SWR
006010  001402                          BEQ     1$
006012  016716  173102                  MOV     LOCK,(SP)
006016  000002                  1$:     RTI

                                        ;TELETYPE OUTPUT ROUTINE

006020  010546                  .TYPE:  MOV     R5,-(SP)
006022  017605  000002                  MOV     @2(SP),R5
006026  062766  000002  000002          ADD     #2,2(SP)
006034  105715                  1$:     TSTB    (R5)                    ;LOOK FOR "0"
006036  001406                          BEQ     3$
006040  105777  173044          2$:     TSTB    @TPCSR                  ;TEST DONE BIT
006044  100375                          BPL     2$
006046  112577  173040                  MOVB    (R5)+,@TPDBR            ;TYPE CHAR
006052  000770                          BR      1$                      ;DO IT AGAIN UNTIL "0" IS SEEN
006054  012605                  3$:     MOV     (SP)+,R5
006056  000002                          RTI

                                        ;ASCII STRING INPUT ROUTINE

006060  010346                  .INSTR: MOV     R3,-(SP)
006062  010446                          MOV     R4,-(SP)
006064  017667  000004  000010          MOV     @4(SP),.MSG
006072  062766  000002  000004          ADD     #2,4(SP)
006100  104402                  .INST1: TYPE
006102  000000                  .MSG:   0
006104  012704  011214                  MOV     #INBUF,R4
006110  012703  000007                  MOV     #7,R3
006114  105777  172764          1$:     TSTB    @TKCSR
006120  100375                          BPL     1$
006122  117714  172760                  MOVB    @TKDBR,(R4)
006126  142714  000200                  BICB    #200,(R4)
006132  121427  000025                  CMPB    (R4),#25                ;IS IT <↑U>
006136  001003                          BNE     200$
006140  104402  010336                  TYPE,MCRLF
006144  000755                          BR      .INST1
006146  122427  000015          200$:   CMPB    (R4)+,#15
006152  001423                          BEQ     INSTR2
006154  117777  172726  172730          MOVB    @TKDBR,@TPDBR
006162  105777  172722          2$:     TSTB    @TPCSR
006166  100375                          BPL     2$
006170  005303                          DEC     R3
006172  001350                          BNE     1$
006174  000402                          BR      .INSTG
006176  010346                  .INSTE: MOV     R3,-(SP)
006200  010446                          MOV     R4,-(SP)
006202  104402                  .INSTG: TYPE
006204  010332                          MOM
006206  005737  007474                  TST     @#RDSW
006212  001402                          BEQ     400$
006214  104402  010336                  TYPE,MCRLF
006220  000727                  400$:   BR      .INST1
006222  012604                  INSTR2: MOV     (SP)+,R4
006224  012603                          MOV     (SP)+,R3
006226  000002                          RTI
```

```
                                        ;CONVERT ASCII STRING TO OCTAL

006230  010546                  .PARAM: MOV     R5,-(SP)
006232  010446                          MOV     R4,-(SP)
006234  016605  000004                  MOV     4(SP),R5
006240  012567  000170                  MOV     (R5)+,LOLIM
006244  012567  000166                  MOV     (R5)+,HILIM
006250  012567  000164                  MOV     (R5)+,DEVADR
006254  112567  000162                  MOVB    (R5)+,LOBITS
006260  112567  000157                  MOVB    (R5)+,ADRCNT
006264  010566  000004                  MOV     R5,4(SP)
006270  005005                  PARAM1: CLR     R5
006272  012704  011214                  MOV     #INBUF,R4
006276  122714  000015                  CMPB    #15,(R4)
006302  001420                          BEQ     PARERR
006304  121427  000060          1$:     CMPB    (R4),#60
006310  002415                          BLT     PARERR
006312  121427  000067                  CMPB    (R4),#67
006316  003012                          BGT     PARERR
006320  142714  000060                  BICB    #60,(R4)
006324  152405                          BISB    (R4)+,R5
006326  122714  000015                  CMPB    #15,(R4)
006332  001414                          BEQ     LIMITS
006334  006305                          ASL     R5
006336  006305                          ASL     R5
006340  006305                          ASL     R5
006342  000760                          BR      1$
006344  122714  000015          PARERR: CMPB    #15,(R4)                ;IS FIRST CHARACTER A <CR>
006350  001003                          BNE     120$
006352  005737  007474                  TST     @#RDSW                  ;IS CKSWR ROUTINE BEING USED
006356  001023                          BNE     PARTI
006360  104404                  120$:   INSTER
006362  000742                          BR      PARAM1

                                        ;TEST TO SEE IF NUMBER IS WITHIN LIMITS

006364  020567  000046          LIMITS: CMP     R5,HILIM
006370  101365                          BHI     PARERR
006372  020567  000036                  CMP     R5,LOLIM
006376  103762                          BLO     PARERR
006400  136705  000036                  BITB    LOBITS,R5
006404  001357                          BNE     PARERR

                                        ;STORE NUMBER AT SPECIFIED ADDRESS

006406  016704  000026                  MOV     DEVADR,R4
006412  010524                  1$:     MOV     R5,(R4)+
006414  062705  000002                  ADD     #2,R5
006420  105367  000017                  DECB    ADRCNT
006424  001372                          BNE     1$
006426  012604                  PARTI:  MOV     (SP)+,R4
006430  012605                          MOV     (SP)+,R5
006432  000002                          RTI
006434  000000                  LOLIM:  0
006436  000000                  HILIM:  0
```

```
006440  000000                  DEVADR: 0
006442  000000                  LOBITS: 0
        006443                  ADRCNT=LOBITS+1

                                        ;SAVE PC OF TEST THAT FAILED AND R0-R5

006444  016667  000004  172522  .SAVO5: MOV     4(SP),SAVPC

                                        ;SAVE R0-R5

006452  010567  172512          SVO5:   MOV     R5,SAVR5
006456  010467  172504                  MOV     R4,SAVR4
006462  010367  172476                  MOV     R3,SAVR3
006466  010267  172470                  MOV     R2,SAVR2
006472  010167  172462                  MOV     R1,SAVR1
006476  010067  172454                  MOV     R0,SAVR0
006502  000002                          RTI

                                        ;RESTORE R0-R5

006504  016700  172446          .RESO5: MOV     SAVR0,R0
006510  016701  172444                  MOV     SAVR1,R1
006514  016702  172442                  MOV     SAVR2,R2
006520  016703  172440                  MOV     SAVR3,R3
006524  016704  172436                  MOV     SAVR4,R4
006530  016705  172434                  MOV     SAVR5,R5
006534  000002                          RTI

                                        ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER

006536  104402                  .CONVR: TYPE
006540  010336                          MCRLF
006542  010046                  .CNVRT: MOV     R0,-(SP)
006544  010146                          MOV     R1,-(SP)
006546  010346                          MOV     R3,-(SP)
006550  010446                          MOV     R4,-(SP)
006552  010546                          MOV     R5,-(SP)
006554  017601  000012                  MOV     @12(SP),R1
006560  016767  012470  172362          MOV     TEMP,TEMP3
006566  062766  000002  000012          ADD     #2,12(SP)
006574  012167  000154                  MOV     (R1)+,WRDCNT
006600  112167  000152          1$:     MOVB    (R1)+,CHRCNT
006604  112167  000147                  MOVB    (R1)+,SPACNT
006610  013167  000144                  MOV     @(R1)+,BINWRD
006614  016704  000140          2$:     MOV     BINWRD,R4
006620  116705  000132                  MOVB    CHRCNT,R5
006624  012700  011254                  MOV     #TEMP,R0
006630  010403                  3$:     MOV     R4,R3
006632  042703  177770                  BIC     #177770,R3
006636  062703  000060                  ADD     #060,R3
006642  110320                          MOVB    R3,(R0)+
006644  006204                          ASR     R4                      ;SHIFT FOR NEXT #
006646  042704  100000                  BIC     #100000,R4              ;CLUGE TO STOP BIT 15 PROPAGATING.
006652  006204                          ASR     R4                      ;DITTO
006654  006204                          ASR     R4                      ;DITTO
006656  005306                          DEC     R5
```

```
006660  001363                          BNE     3$
006662  012703  011314                  MOV     #MDATA,R3
006666  114123                  4$:     MOVB    -(R0),(R3)+
006670  105367  000062                  DECB    CHRCNT
006674  001374                          BNE     4$
006676  105767  000055                  TSTB    SPACNT
006702  001405                          BEQ     6$
006704  112723  000040          5$:     MOVB    #040,(R3)+
006710  105367  000043                  DECB    SPACNT
006714  001373                          BNE     5$
006716  105013                  6$:     CLRB    (R3)
006720  104402                          TYPE
006722  011314                          MDATA
006724  005367  000024                  DEC     WRDCNT
006730  001323                          BNE     1$
006732  016767  172212  002314          MOV     TEMP3,TEMP
006740  012605                          MOV     (SP)+,R5
006742  012604                          MOV     (SP)+,R4
006744  012603                          MOV     (SP)+,R3
006746  012601                          MOV     (SP)+,R1
006750  012600                          MOV     (SP)+,R0
006752  000002                          RTI
006754  000000                  WRDCNT: 0
006756  000000                  CHRCNT: 0
        006757                  SPACNT=CHRCNT+1
000760  000000                  BINWRD: 0

                                        ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
                                        ;BUFFER TO THE CHARACTERS "N" AND "Y".
                                        ;IF THE CHARACTER IS "N" CLEAR THE FLAG
                                        ;IF THE CHARACTER IS "Y" SET THE FLAG

006762  017605  000000          .SETFLG:MOV     @(SP),R5
006766  122767  000116  002220          CMPB    #'N,INBUF       ;IS IT "N" ?
006774  001002                          BNE     1$
006776  105015                          CLRB    (R5)    ;000
007000  000406                          BR      2$
007002  122767  000131  002204  1$:     CMPB    #'Y,INBUF               ;IS IT "Y" ?
007010  001005                          BNE     3$
007012  112715  177777                  MOVB    #-1,(R5)                ;377
007016  062716  000002          2$:     ADD     #2,(SP)
007022  000002                          RTI
007024  104404                  3$:     INSTER                          ;RETRY
007026  000755                          BR      .SETFLG
                                        ;TRAP DISPATCH SERVICE
                                        ;ARGUMENT OF TRAP IS EXTRACTED
                                        ;AND USED AS OFFSET TO OBTAIN POINTER
                                        ;TO SELECTED SUBROUTINE

007030  011646                  .TRPSR: MOV     (SP),-(SP)              ;GET PC OF RETURN
007032  162716  000002                  SUB     #2,(SP)                 ;=PC OF TRAP
007036  017616  000000                  MOV     @(SP),(SP)              ;GET TRP
007042  006316                  TRPOK:  ASL     (SP)                    ;MULTIPLY TRAP ARG BY 2
007044  042716  177001                  BIC     #177001,(SP)            ;CLEAR UNWANTED BITS
007050  062716  001226                  ADD     #.TRPTAB,(SP)           ;POINTER TO SUBROUTINE ADDRESS
007054  017616  000000                  MOV     @(SP),(SP)              ;SUBROUTINE ADDRESS
```

```
007060  000136                          JMP     @(SP)+                  ;GO TO SUBROUTINE

                                        ;ERROR HANDLER

007062  104413                  .HLT:   CKSWR                           ;CHECK FOR ↑G
007064  032777  020000  172006          BIT     #SW13,@SWR              ;INHIBIT ERROR TYPE OUT ?
007072  001061                          BNE     HALTS
007074  021567  172034                  CMP     (SP),LSTERR
007100  001404                          BEQ     1$
007102  011667  172026                  MOV     (SP),LSTERR
007106  105767  172112                  CLRB    ERRFLG
007112  104406                  1$:     SAVOS
007114  011605                          MOV     (SP),R5
007116  162705  000002                  SUB     #2,R5
007122  011504                          MOV     (R5),R4
007124  006304                          ASL     R4
007126  061504                          ADD     (R5),R4
007130  006304                          ASL     R4
007132  042704  177001                  BIC     #177001,R4
007136  062704  012030                  ADD     #.ERRTAB,R4
007142  012467  000040                  MOV     (R4)+,ERRMSG
007146  012467  000046                  MOV     (R4)+,DATAHD
007152  011467  000054                  MOV     (R4),DATABP
007156  105767  172042                  TSTB    ERRFLG
007162  001403                          BEQ     TYPMSG
007164  005767  000042                  TST     DATABP
007170  001014                          BNE     TYPDAT
007172  104410                  TYPMSG: CONVRT
007174  007324                          ERTABO
007176  112767  177777  172020          MOVB    #-1,ERRFLG
007204  104402                          TYPE
007206  000000                  ERRMSG: 0
007210  005767  000004                  TST     DATAHD
007214  001402                          BEQ     TYPDAT
007216  104402                          TYPE
007220  000000                  DATAHD: 0
007222  005767  000004          TYPDAT: TST     DATABP
007226  001402                          BEQ     RESREG
007230  104410                          CONVRT
007232  000000                  DATABP: 0
007234  104407                  RESREG: RESOS
007236  005777  171636          HALTS:  TST     @SWR
007242  100005                          BPL     EXITER
007244  010046                          PUSHRO
007246  016600  000002                  MOV     2(SP),R0
007252  000000                          HALT
007254  012600                          POPRO
007256  104413                  EXITER: CKSWR                           ;CHECK FOR ↑G
007260  005267  171646                  INC     ERRCNT
007264  032777  000400  171606          BIT     #SW08,@SWR              ;LOOP ON ERROR ?
007272  001007                          BNE     1$
007274  032777  002000  171576          BIT     #SW10,@SWR              ;ESCAPE TO NEXT ON ERROR ?
007302  001407                          BEQ     2$
007304  016767  171606  171602          MOV     NEXT,RTRN               ;SET UP FOR NEXT TEST
007312  012706  001100          1$:     MOV     #STACK,SP               ;REINITIALIZE SP
007316  000177  171572                  JMP     @RTRN
```

```
007322  000002                  26:     RTI
007324  000001                  ERTABO: 1
007326     006     002                  .BYTE   6,2
007330  001174                          SAVPC
                                        ;ENTER HERE ON POWER FAILURE


007332  010046                  .PFAIL: MOV     R0,-(SP)                ;SAVE R0-R5 ON PROCESSOR STACK
007334  010146                          MOV     R1,-(SP)
007336  010246                          MOV     R2,-(SP)
007340  010346                          MOV     R3,-(SP)
007342  010446                          MOV     R4,-(SP)
007344  010546                          MOV     R5,-(SP)
007346  016746  170452                  MOV     24,-(SP)
007352  010667  171614                  MOV     SP,SAVSP                ;SAVE STACK POINTER
007356  012767  007370  170440          MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
007364  000000                          HALT                            ;HALT ON POWER DOWN NORMAL
007366  000777                  1$:     BR      1$

                                        ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED

007370  016706  171576          RESTAR: MOV     SAVSP,SP                ;RESTORE STACK POINTER
007374  012605                          MOV     (SP)+,R5                ;RESTORE R0-R5
007376  012604                          MOV     (SP)+,R4
007400  012603                          MOV     (SP)+,R3
007402  012602                          MOV     (SP)+,R2
007404  012601                          MOV     (SP)+,R1
007406  012600                          MOV     (SP)+,R0
007410  012767  007332  170406          MOV     #.PFAIL,24              ;SET UP FOR POWER FAILURE
007416  012767  000340  170352          MOV     #340,PS
007424  012706  001100                  MOV     #STACK,SP
007430  005067  001620                  CLR     TEMP
007434  005267  001614          1$:     INC     TEMP
007440  001375                          BNE     1$
007442  104410                          CONVRT
007444  007466                          PFTAB
007446  104402                          TYPE
007450  010341                          MPFAIL
007452  005067  171546                  CLR     ERRFLG
007456  005067  171542                  CLR     LSTERR
007462  000177  171426                  JMP     @RTRN
007466  000001                  PFTAB:  1
007470     006     002                  .BYTE   6,2
007472  001114                          RTRN


                                        ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR 1G TO ALLOW CHANGING
                                        ;OF LOC.176.
                                        ;LOCATIONS USED:
007474  000000                          RDSW:   .WORD   0


007476  005737  000042          .CKSWR: TST     @#42
007502  001042                          BNE     OUT
007504  022767  000176  171366          CMP     #SWREG,SWR              ;SOFTWARE SWITCH REGISTER PRESENT
007512  001036                          BNE     OUT                     ;NO, GET OUT
```

```
007514  105777  171364                  TSTB    @TKCSR                  ;YES,WAIT FOR
007520  100033                          BPL     OUT                     ;READY, GET CHARACTER
007522  017767  171360  176352          MOV     @TKDBR,.MSG             ;AND STRIP OFF
007530  042767  177600  176344          BIC     #177600,.MSG            ;THE GARBAGE
007536  122767  000007  176336          CMPB    #7,.MSG                 ;IS IT A <^G>
007544  001021                          BNE     OUT
007546  104402  007624                  TYPE,$CNTG
007552  005137  007474          .CNTLU: COM     @#RDSW
007556  104402  007631                  TYPE,$MSWR
007562  104411  007616                  CNVRT,SWREGC
007566  104402  007641                  INSTR,$MNEW
007572  104405                          PARAM
007574  000000                          0
007576  177777                          177777
007600  000176                          SWREG
007602     000     001          .BYTE           0,1
007604  104402  010336                  TYPE,MCRLF
007610  005037  007474          OUT:    CLR     @#RDSW
007614  000002                          RTI
007616  000001                  SWREGC: 1
007620     006     002          .BYTE   6,2
007622  000176                          SWREG
007624  005015  043536     000  $CNTG:  .ASCIZ  <15><12>/^G/
007631     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR= /
007636  020075     000
007641     040  047040  053505  $MNEW:  .ASCIZ  /  NEW= /
007646  020075     000
                007652                  .EVEN
007652  005015  042012  030525  MTITLE: .ASCIZ  <15><12><12>/DU11 DZDUF-C TAPE F /<15><12>
007660  000061  051104  000004
007666  000036  000103  042524
007674  000020  043040  006440
007702  000012
007704  005015  042526  052103  MVECTO: .ASCIZ  <15><12>/VECTOR ADDRESS-/
007712  051117  040440  052104
007720  042522  051523  000005
007726  000015  051461  000024  MREGAD: .ASCIZ  <15><12>/1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS-/
007734  000004  000006  000003
007742  000072  042522  042503
007750  053111  051105  041440
007756  047117  051124  046117
007764  051040  041505  051511
007772  042524  020122  052101
010000  051104  051505  026523
010006     000
010007     015  040412  042522  MMULT:  .ASCIZ  <15><12>/ARE YOU RUNNING MULTIPLE DEVICES ? (Y OR N)-/
010014  054440  052517  051040
010022  047125  044516  043516
010030  046440  046125  044524
010036  046120  020106  042504
010044  044526  042503  020123
010052  020077  054450  047440
010060  020122  024516  000055
010066  005015  040514  052123  MLASTD: .ASCIZ  <15><12>/LAST DEVICE:RECEIVER CONTROL REGISTER ADDRESS-/
010074  042040  053105  041511
010102  035106  042522  042503
```

```
010110  053111  051105  041440
010116  047117  051124  046117
010124  051040  043505  051511
010132  042524  020122  042101
010140  051104  051505  026523
010146     000
010147     075  042504  044526  DEVICE: .ASCIZ  /=DEVICE  /
010154  042503  020040     000
010161     015  044012  053517  MCOW:   .ASCIZ  <15><12>/HOW NOW BROWN COW? ...SELECT SOMETHING TO RUN @ACTREG/
010166  047040  053517  041040
010174  047522  047127  041440
010202  053517  020077  027056
010210  051456  046105  041505
010216  020124  047523  042515
010224  044124  047111  020107
010232  047524  051040  047125
010240  040040  041501  051124
010246  043505     000
010251     015  047412  052125  MRANGE: .ASCIZ  <15><12>/OUT OF RANGE:RETYPE LAST DEVICE RXCSR ADDRESS-/
010256  047440  020106  040522
010264  043516  035105  042522
010272  054524  042520  046040
010300  051501  020124  042504
010306  044[illegible]6  042503  051040
010314  041530  051123  040440
010322  042104  042522  051523
010330  000055
010332  020040  000077          MOM:    .ASCIZ  /  ?/
010336  000015     000          MCRLF:  .ASCIZ  <15><12>
010341     040  050040  053517  MPFAIL: .ASCIZ  /  POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/
010346  051105  [illegible]  [illegible]01
010[illegible]4  [illegible]514  [illegible]2  [illegible]4
010[illegible]2  051120  [illegible]17  [illegible]32
010[illegible]0  020115  042522  052123
010376  051101  020124  042101
010404  020040  051505  020124
010412  047111  020040  047522
010420  051107  051505  000123
010426  020015  047105  020104  MEPASS: .ASCIZ  <15><12>/END OF PASS TAPE F/
010434  043117  020040  051501
010442  020123  [illegible]4  042520
010450  043040     000
010453     015  051012     000  MR:     .ASCIZ  <15><12>/R/
010457     015  020012  051[illegible]5  MTSTPC: .ASCIZ  <15><12>/TEST PC-/
010464  020124  [illegible]  [illegible]
010472  020015  047514  020[illegible]3  MLOCK:  .ASCIZ  <15><12>/LOCK ON SELECTED TEST? (Y OR N)-/
010[illegible]0  047440  020116  020[illegible]3
010[illegible]6  047514  020103  020[illegible]5
010514  020040  020[illegible]5  037524
010[illegible]2  020040  020131  051117
010[illegible]0  047040  020451     000
010[illegible]6     015  020012  020125  MLEVEL: .ASCIZ  <15><12>/DU PRIORITY LEVEL-/
010[illegible]2  051120  047511  044522
010[illegible]0  054524  046040  053105
010[illegible]6  046105  000055
010[illegible]2  005015  020043  043117  MSYNC:  .ASCIZ  <15><12>/# OF SYNC CHARS SELECTED ( 1 OR 2)-/
```

```
010570  051440  047131  020103
010576  044103  051101  020123
010604  042523  042514  052103
010612  042105  024040  030440
010620  047440  020122  024462
010626  000055
010630  005015  051511  051440  MWIRE6: .ASCIZ  <15><12>/IS SEC XMIT JUMPER #6 IN? (Y OR N)-/
010636  041505  054040  044515
010644  020124  052512  050115
010652  051105  021440  020066
010660  047111  020077  054450
010666  047440  020122  024516
010674  000055
010676  005015  051511  051440  MWIRE5: .ASCIZ  <15><12>/IS SEC REC JUMPER #5 IN? (Y OR N)-/
010704  041505  051040  041505
010712  045040  046525  042520
010720  020122  032443  044440
010726  037516  024040  020131
010734  051117  047040  026451
010742     000
010743     015  044412  020123  MWIRE4: .ASCIZ  <15><12>/IS OPT CLR ENABLE JUMPER #4 IN? (Y OR N)-/
010750  050117  020124  046103
010756  020122  047105  041101
010764  042514  045040  046525
010772  042520  020122  020043
011000  044440  037516  020040
011006  020131  051117  047040
011014  026451     000
011017     015  040412  042522  MEXTJ:  .ASCII  <15><12>/ARE YOU RUNNING IN MAINT MODE EXTERNAL?/
011024  054440  020517  051040
011032  047125  044516  020016
011040  044440  020116  020015
011046  047111  020124  044515
011054  042504  042440  052130
011062  051105  044516  037514
011070  005015  044001  052116          .ASCII  <15><12><1>/AND ..... DO YOU HAVE THE EXTERNAL MODEM BYPASS/
011076  027040  027056  020056
011104  042040  020117  047531
011112  020125  044510  020006
011120  020040  042010  020040
011126  020130  051105  020016
011134  020114  047515  020004
011142  020115  054502  042020
011150  020123
011152  020015  045001  046525          .ASCIZ  <15><12><1>/JUMPER CONNECTOR ON ?(Y OR N)-/
011160  042520  020122  020003
011166  047116  041505  047524
011174  020122  047117  037440
011202  020040  047440  020122
011210  024516  000055
                                  .EVEN

                                          ;BUFFERS FOR INPUT-OUTPUT

011214  000040                    INBUF:  .BLKB   40
011294  000040                    TEMP:   .BLKB   40
```

```
011314  000040                  MDATA:  .BLKB   40
                                ;*********************************************************
                                        ;UTILITIES
                                ;*********************************************************

                                ;THIS UTILITY CALCULATES PRIORITY LEVEL
011354  006367  000044          DULEV:  ASL     DUPRT   ;SHIFT LEFT
011360  006367  000040                  ASL     DUPRT   ;
011364  006367  000034                  ASL     DUPRT   ;
011370  006367  000030                  ASL     DUPRT   ;
011374  006367  000024                  ASL     DUPRT   ;
011400  016767  000020  000020          MOV     DUPRT,LESS1     ;MOVE THIS TO LESS1
011406  162767  000001  000012          SUB     #1,LESS1        ;CREATE LESS1
011414  042767  000037  000004          BIC     #37,LESS1       ;CLEAR TNZVC
011422  000207                          RTS     PC
011424  000040                  DUPRT:  LEVEL5
011426  000200                  LESS1:  LEVEL4  ;LEVEL TO ALLOW INTERRUPTS

                                        ;NEW DU ADDRESSES
011430  016767  000126  000422  DUADDR: MOV     DUBASE,RXCSR    ;XXX0
011436  005267  000120                  INC     DUBASE
011442  016767  000114  000412          MOV     DUBASE,HRXCSR   ;XXX1
011450  005267  000106                  INC     DUBASE
011454  016767  000002  000402          MOV     DUBASE,RXDBUF   ;XXX2
011462  016767  000004  000400          MOV     DUBASE,PARCSR   ;XXX2
011470  005267  000006                  INC     DUBASE
011474  016767  000002  000364          MOV     DUBASE,HRXDBUF  ;XXX3
011502  016767  000004  000362          MOV     DUBASE,HPARCSR  ;XXX3
011510  005267  000006                  INC     DUBASE
011514  016767  000002  000352          MOV     DUBASE,TXCSR    ;XXX4
011522  005267  000034                  INC     DUBASE
011526  016767  000000  000342          MOV     DUBASE,HTXCSR   ;XXX5
011534  005267  000002                  INC     DUBASE
011540  016767  000016  000332          MOV     DUBASE,TXDBUF   ;XXX6
011546  005267  000010                  INC     DUBASE
011552  016767  000004  000322          MOV     DUBASE,HTXDBUF  ;XXX7
011560  000207                          RTS     PC
011562  000000                  DUBASE: 0

                                        ;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
                                        ;INFORMATION CONTAINED IN TEMP1 AND IT IS
                                        ;SHIFTED IN BY THE CONTENTS OF SHIFT
011564  042777  040000  000302  RPOKE:  BIC     #MTDATA,@TXCSR
011572  005067  167300                  CLR     TEMP2
011576  006067  167342                  ROR     TEMP1   ;FORCE CARRY
011602  006067  167340                  ROR     TEMP2   ;PICK UP CARRY IN BIT 15
011606  006267  167334                  ASR     TEMP2   ;SHIFT INTO BIT 14
011612  042767  100000  167326          BIC     #BIT15,TEMP2    ;CLR BIT 15
011620  053777  167300  000346          BIS     TEMP2,@TXCSR    ;POKE MAINT DATA
011626  042777  020000  000240          BIC     #CLK,@TXCSR     ;POKE CLK
011634  052777  020000  000232          BIS     #CLK,@TXCSR     ;
011642  006367  167272                  DEC     SHIFT
011646  001346                          BNE     RPOKE
011650  000207                          RTS     PC

                                        ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
```

```
011652  016767  167266  167266  0008:   MOV     TEMP1,TEMP2     ;SAVE TEMP1
011660  005067  167264                  CLR     TEMP3
011664  012727  000010                  MOV     #8.,(PC)+
011670  000000                  1$:     0
011672  006067  167250          2$:     ROR     TEMP2
011676  005567  167246                  ADC     TEMP3
011702  005367  177762                  DEC     1$
011706  001371                          BNE     2$
011710  006067  167234                  ROR     TEMP3
011714  103404                          BCS     3$
011716  052767  000400  167220          BIS     #8ITB,TEMP1     ;SET ODD PARITY
011724  000400                          BR      3$
011726  042767  000400  167210  3$:     BIC     #8ITB,TEMP1     ;CLR EVEN PARITY
                                        ;TEMP1 NOW HAS ODD PARITY CHARACTER
011734  000207                  4$:     RTS     PC

                                ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
011736  016767  167202  167202  EVEN8:  MOV     TEMP1,TEMP2     ;SAVE TEMP1
011744  005067  167200                  CLR     TEMP3
011750  012727  000010                  MOV     #8.,(PC)+
011754  000000                  1$:     0
011756  006067  167164          2$:     ROR     TEMP2
011762  005567  167162                  ADC     TEMP3
011766  005367  177762                  DEC     1$
011770  001371                          BNE     2$
011772  006067  167150                  ROR     TEMP3
012000  103004                          BCC     3$
012002  052767  000400  167134          BIS     #8ITB,TEMP1     ;SET EVEN PARITY
012010  000403                          BR      4$
012012  042767  000400  167124  3$:     BIC     #8ITB,TEMP1     ;CLR ODD PARITY
                                        ;TEMP1 NOW HAS EVEN PARITY CHARACTER
012020  000207                  4$:     RTS     PC
012022  062716  000002          TRPREG: ADD     #2,(SP) ;ALLOW IT TO "CRUNCH" INTO HLT BACK
                                                        ;IN MAIN PART OF THE PROGRAM
012026  000002                          RTI
                                        ;ERROR HLT TABLE
012030  012114                  .ERRTAB:        EM0     ;HLT 0  BIT ERROR (GENERAL)
012032  000000                                  0
012034  000000                                  0
012036  012130                                  EM1     ;HLT 1  REGISTER ERROR
012040  012301                                  DH1
012042  012322                                  DT1
012044  012172                                  EM2     ;HLT 2  RECEIVER ERROR
012046  012301                                  DH1
012050  012322                                  DT1
012052  012234                                  EM3     ;HLT 3  TRANSMITTER ERROR
012054  012301                                  DH1
012056  012322                                  DT1
                                        ;DEFAULT DU ADDRESSES
012060  160040                  RXCSR:  160040
012062  160041                  HRXCSR: 160041
012064  160042                  RXBUF:  160042
012066  160043                  HRXBUF:         160043
012070  160042                  PRXBUF: 160042
012072  160043                  HPRXBUF:        160043
012074  160044                  TXCSR:  160044
```

```
012076  160045                  HTXCSR: 160045
012100  160046                  TXDBUF: 160046
012102  160047                  HTXDBUF:        160047
                                        ;DEFAULT DU VECTORS
012104  000770                  DURIV:  770     ;REC INTR VECTOR
012106  000772                  DURIS:  772     ;REC INTR STATUS
012110  000774                  DUTIV:  774     ;XMIT INTR VECTOR
012112  000776                  DUTIS:  776     ;XMIT INTR STATUS
                                        ;ERROR MESSAGES
012114  036440  042440  051122  EM0:    .ASCIZ  / = ERROR PC/
012122  051117  050040  000103
012130  036440  051040  043505  EM1:    .ASCIZ  / = REGISTER ERROR PC/<15><12><1>/REGISTER  /
012136  051511  042524  020122
012144  051105  042522  020122
012152  041520  005015  051001
012160  043505  051511  042524
012166  020122  000040
012172  036440  051040  041505  EM2:    .ASCIZ  / = RECEIVER ERROR PC/<15><12><1>/REGISTER  /
012200  044505  042526  020122
012206  051105  047522  020122
012214  041520  005015  051001
012222  043505  051511  042524
012230  020122  000040
012234  036440  051040  040522  EM3:    .ASCIZ  / = TRANSMITTER ERROR PC/<15><12><1>/REGISTER  /
012242  051516  044515  052124
012250  051105  042440  051122
012256  051117  050040  006503
012264  000412  042522  044507
012272  052123  051105  020040
012300     000
                                        ;DATA HEADERS FOR ERROR MESSAGES
012301     105  050130  041505  DH1:    .ASCIZ  /EXPECTED  ACTUAL/
012306  042524  020104  040440
012314  052103  043525  000114
                                .EVEN
                                        ;DATA TABLES FOR ERROR MESSAGES
012322  000003                  DT1:    3
012324     006     004                  .BYTE   6,4
012326  001164                          SAVR3   ;REGISTER
012330     006     004                  .BYTE   6,4
012332  001156                          SAVR0   ;EXPECTED DATA
012334     006     002                  .BYTE   6,2
012336  001160                          SAVR1   ;ACTUAL DATA
        000001                  .END
```

```
AAA     002046          954#
ACTREG  001216          772#    902*    916*    917*    924*    1652    1655    1665
ADRCNT= 006443         1817*   1856*   1865#
BASEAD  001204          767#    884*    921*    922     928*    930*    1659*   1671*   1675
BASEIV  001212          770#    894*   1660*   1672*   1677    1678*   1679    1680*   1681    1682*   1683    1684*
BBB     001664          901     905#
BINWRD  006760         1906*   1907    1944#
BITW  = 002000          682#
BIT0  = 000001          630#    658     689
BIT1  = 000002          629#    657
BIT10 = 002000          620#    648     682
BIT11 = 004000          619#    647
BIT12 = 010000          618#    646     663
BIT13 = 020000          617#    645     662     681
BIT14 = 040000          616#    644     661     680
BIT15 = 100000          615#    643     660     679     2302
BIT2  = 000004          628#    656     997
BIT3  = 000010          627#    655     688
BIT4  = 000020          626#    654     687
BIT5  = 000040          625#    653     686
BIT6  = 000100          624#    652     685
BIT7  = 000200          623#    651     684
BIT8  = 000400          622#    650     666     683     2323    2338    2340
BIT9  = 001000          621#    649     665
BREAK = 000001          689#
BRW     005774          994     1008    1745#
BRX     005776          995     1009    1746#
CARDET= 010000          646#
CCC     005530         1651    1674    1691#
CHRCNT  006756         1904*   1908    1922*   1942#   1943
CKSWR = 104413          814#    1704    1726    1750    1979    2022
CLK   = 020000          681#    1050    1051    1054    1055    1071    1072    1100    1140    1141    1144    1145    1161
                       1162    1190    1230    1231    1234    1235    1251    1252    1280    1320    1321    1324    1335
                       1341    1342    1370    1409    1410    1414    1415    1459    1460    1464    1465    1511    1518
                       1528    1535    1588    1595    1605    1612    2304    2305
CNTLU = 104414          816#    861
CNVRT = 104411          810#    2036
CONVRT= 104410          808#    1648    2002    2013    2065
COUNT   001142          743#    1063*   1075*   1153*   1165*   1243*   1255*   1333*   1345*
CTS   = 020000          645#
DATABP  007732         1997*   2000    2011    2014#
DATAHD  007730         1996*   2007    2010#
DEVADR  006440         1815*   1853    1863#
DEVICE  010147         1649    2149#
DH1     012301         2351    2354    2357    2398#
DISPRE  000174          710#    854
DLIGHT= 177570          600#    845
DMA   = 100000          679#
DMAINT= 000040          686#
DSC   = 100000          643#
DSINTE= 000040          653#
DSR   = 001000          649#
DSWR  = 177570          599#    844
DTR   = 000002          657#
DT1     012322         2352    2355    2358    2403#
DUADDR  011430          683     1676    2274#
```

```
DUBASE  011562    879    882    1675*  2274   2275*  2276   2277*  2278   2279   2280*  2281   2282   2283*
                 2284   2285*  2286   2287*  2288   2289*  2290   2292*
DULEV   011354    949   2261*
DUPRT   011424    946   2261*  2262*  2263*  2264*  2265*  2266   2270*
DURIS   012106   1679*  2372*
DURIV   012104    890    893    894   1677*  1684   2371*
DUTIS   012112   1683*  2374*
DUTIV   012110   1681*  2373*
EIGHT = 006000    674*  1314   1403   1453   1505   1532
EM0     012114   2347   2376*
EM1     012130   2350   2378*
EM2     012172   2353   2384*
EM3     012234   2356   2390*
ERRCNT  001132    736*   835*  2023*
ERRFLG  001224    782*   834*  1693*  1739*  1985*  1998   2004*  2069*
ERRMSG  007206   1995*  2006*
ERTABO  007324   2003   2032*
EVENB   011736   2358*
EVEPAR= 001400    677*  1403   1505
EVPAR = 000400    666*
EXITER  007256   2017   2022*
FIVE  = 000000    671*  1044
FRMERR= 000000    662*
HALTS   007236   1981   2016*
HDXEN = 000010    688*
HILIM   006436   1814*  1844   1862*
HOLD    001136    741*  1513   1520   1530   1537   1590   1597   1607   1614
HPARCS  012072   2282*  2365*
HPXCSR  012062   2276*  2361*
HPXCSB  012066   2291*  2363*
HTXCSR  012076   2[illegible]6*  2367*
HTXCSB  012102   2240*  2369*
ICOUNT  001122    732*  1737   1741*
INCBF   011214    957    961   1778   1820   1952   1956   2253*
INIFLG  001222    780*   840   843*
INSTER= 104404    800*   965   1839   1961
INSTR = 104403    798*   874    885    895    906    931    941    955    968    972    976    980    999
                 1012   2097
INSTR2  006222   1789   1804*
ISYMOD= 000000    670*
JMPBY   001203    763*   983   1494   1571
KEEPAD  001206    768*   882*   884    928    930   1671
KEEPIV  001214    771*   893*  1672
LASTAD  001210    769*   911    922    936
LESS1   011426   2266*  2267*  2268*  2271*
LEVEL0= 000000    640*
LEVEL1= 000040    639*
LEVEL2= 000100    638*
LEVEL3= 000140    637*
LEVEL4= 000200    636*  2271
LEVEL5= 000240    635*  2270
LEVEL6= 000300    634*
LEVEL7= 000340    633*
LIGHTS  001102    721*   845*   854*  1695*
LIMITS  006364   1830   1844*
LOBITS  006442   1816*  1848   1864*  1865
```

```
LOCK     001120    731#   1040#  1130#  1220#  1310#  1753
LOGICA   005562   1699#
LOKFLG   001225    783#   1002   1003
LOLIM    006434   1813#   1846   1861#
LPCNT    001124    733#    831#  1736#  1737   1740#
LSTERR   001134    737#    836#  1692#  1982   1984#  2070#
MCOW     010161   1654    2151#
MCRLF    010336   1786    1802   1894   2103   2171#
MDATA    011314   1920    1931   2255#
MEPASS   010426   1647    2181#
MEXT  =  010000    693#   1502   1579
MEXTJ    011017    981    2227#
MINT  =  004000    692#   1045   1135   1225   1315   1400   1450
MLASTD   010066    907    2140#
MLEVEL   010535    942    2194#
MLOCK    010472   1000    2188#
MMULT    010007    896    2132#
MPFAIL   010341   2068    2172#
MQM      010332   1799    2170#
MR       010453   1022    2185#
MRANGE   010251    932    2161#
MREGRD   007726    875    2123#
MRESET=  000400    683#   1041   1106   1131   1196   1221   1286   1311   1376   1394   1396   1444   1416
                  1496    1498   1573   1575
MSYNC    010562    956    2198#
MTDATA=  040000    680#   2297
MTITLE   007652    842    2115#
MTSTPC   010457   1013    2186#
MULTD    001202    762#    898    899    991   1650
MVECTO   007704    886    2120#
MWIRE4   010743    977    2219#
MWIRE5   010676    973    2212#
MWIRE6   010630    969    2205#
NEXT     001116    730#   1039#  1129#  1219#  1309#  1393#  1443#  1493#  1570#  1722   1742   2028
NOPAR =  000000    675#   1044   1134   1224   1314
OCOPAR=  001000    676#   1453   1562
OCOB     011552   2311#
ONCE     001352    841    844#
OPTCLR   001201    761#    979
OUT      007610   2084    2086   2088   2092   2104#
OUTCRY   00[illegible]22   1648   1687#
OUTMUL   00[illegible]4    904    929    940#
OVRFLN=  0[illegible]0    661#
PARAM =  104[illegible]5    802#    876    887    908    933    943   1014   2098
PARAM1   006270   1819#   1840
PARCSR   012070   1046#   1061#  1136#  1151#  1226#  1241#  1316#  1331#  1395#  1403.  1445#  1453#  1497#
                  1505#   1574#  1582#  2279#  2364#
PAREN =  001000    665#
PARER =  010000    663#
PARERR   006344   1822    1824   1826   1835#  1845   1847   1849
PARTI    006426   1838    1838#
PASCNT   001130    735#    833#  1694#  1695
PFTAB    007466   2066    2072#
POPRO =  012600    609#   2021
POP1SP=  005726    607#
POP2SP=  022626    611#
```

```
PS     = 177776          601#   828*   987*  1703*  2060*
PUSHPO= 010046           603#  2018
PUSH1S= 005746           606#
PUSH2S= 024646           610#
RDSW    007474          1800   1837   2080#  2094*  2104*
RECRCI= 004000           647#  1056   1080   1146   1170   1236   1260   1326   1350
REPLAY  005426          1667   1673   1675#
RESREG  007234          2012   2015#
RESTAR  007370          2046   2052#
RESTRT  005572          1669   1697   1703#
RESOS = 104407           806#  2015
RING   = 040000          644#
RINTEN= 000100           652#
ROTADD  001220           775#   903*   915*   917    919*   924    927*  1662*  1665   1669*
RPOKE   011564          2297#  2307
RTRN    001114           729#   838*  1017   1021*  1023   1705*  1722*  1723   1742*  1743   2028*  2030   2071
        2074
RTS    = 000004          656#
RUNA   = ****** U          1     33     45    108    546
RUNB   = ****** U          1     33     45    109    546
RUNC   = ****** U          1     33     45    108    546
RUND   = ****** U          1     33     45    108    546
RUNE   = ****** U          1     33     45    108    546
RUNF   = ****** U          1     33     45    108    546
RUNIT   005332          1653   1659#  1666
RXCSR   012060          1047*  1056   1080   1107   1137*  1146   1170   1197   1227*  1236   1260   1287   1317*
        1326   1350   1377   1407*  1423   1457*  1473   1509*  1549   1586*  1626   1689   2274*
        2[illegible]0#
RXDBUF  012064          1042   1086   1132   1176   1222   1266   1312   1356   1404   1427   1454   1477   1506
        1553   1583   1630   2278*  2362#
RXDONE= 000200           651#
RX[illegible] = 100000   650#
SAVFC   001174           756#   1869*  2034
SAVR0   001156           749#   1878*  1893   2407
SAVR1   001160           750#   1877*  1894   2409
SAVR2   001162           751#   1876*  1895
SAVR3   001164           752#   1875*  1896   2405
SAVR4   001166           753#   1874*  1887
SAVR5   001170           754#   1873*  1938
SAVSP   001172           755#   2045*  2052
SAVOS = 104426           804#   1906
SCOPE = 104430           792#   1114   1204   1294   1384   1434   1484   1561   1638
SCOPI = 104401           794#   1036   1186   1276   1366
SEND  = 000020           687#   1045   1135   1225   1315   1400   1450   1502   1579
S[illegible]EC  001230   760#    975
SETFLG= 104412           812#    897    970    974    978    982   1001
SEVEN = 004000           673#   1224
SEXMIT  001177           759#    971
SHIFT   001140           742#   1069*  1073*  1159*  1163*  1249*  1253*  1339*  1343*  1411*  1416*  1417   1421
        1461*  1466*  1467   1471   1525*  1542*  1543   1547   1602*  1619*  1620   1624   2306*
SIX    = 002000          672#   1134
SPACNT= 0[illegible]57  1905*  1924   1927*  1943#
S[illegible]D  = 0[illegible]000  648#
STACK = 001130           602#    829    988   2029   2061
STD    = 000010          655#
STFLG   001223           781#    832*
```

```
STPSYN= 000400          650#
SVOS    006452         1873#
SWR     001100          720#   844#   849    853#   859    862    997    1010   1727   1734   1751   1980   2016
                       2024   2026   2005
SWREG   000176          711#   8[illegible]3   859    2085   2101   2108
SWREGC  007616         2096   2106#
SW00  = 000001          582#   862
SW01  = 000002          581#   1010
SW02  = 000004          580#
SW03  = 000010          579#
SW04  = 000020          578#
SW05  = 000040          577#
SW06  = 000100          576#
SW08  = 000400          575#   2024
SW09  = 001000          574#   1751
SW10  = 002000          573#   2026
SW11  = 004000          572#   1734
SW12  = 010000          571#
SW13  = 020000          570#   1980
SW14  = 040000          569#   1727
SW15  = 100000          568#
SYNCHO  001176          758#   959#   963#   1078   1168   1258   1348
SYNEXT= 020000          669#   1395   1403   1445   1453   1497   1505   1574   1582
SYNINT= 030000          669#   1044   1134   1224   1314
SYNSCH= 000020          654#   1047   1107   1137   1197   1227   1287   1317   1377   1407   1457   1509   1586
SYSTST= 014000          694#
TEMP    011254         1901   1909   1934#  2062#  2063#  2254#
TEMP1   001144          744#   2299#  2311   2321#  2323#  2328   2338#  2340#
TEMP2   001146          745#   2293#  2300#  2301#  2302#  2303   2311#  2315#  2328#  2332#
TEMP3   001150          746#   1901#  1934   2312#  2316#  2319#  2329#  2333#  2336#
TEMP4   001152          747#
TEMP5   001154          748#
TKCSR   001104          722#   1730   1780   2097
TKDBR   001106          723#   1732   1782   1790   2089
TLAST = 005004         1016   1640#
TPCSR   001110          724#   1763   1791
TPDBR   001112          725#   1765#  1790#
TRPOK   007042         1971#
TRPREG  012022         2343#
TSTN0   001126          734#   837#   1039#  1128#  1218#  1308#  1392#  1442#  1492#  1569#
TST1    002350         1015   1021   1036#  1705   1706
TST2    002376         1039   1128#
TST3    002224         1129   1219#
TST4    003552         1219   13[illegible]#
TST5    004100         1309   13[illegible]#
TST6    004312         1393   1442#
TST7    004524         1443   1492#
TST8    005004         1493   1569#  1640
TST9  = ###### U       1570
TTST    005676          994#   9[illegible]#  1005#  1006#  1008#  1009#  1728#
TXCSR   012074         1041#  10[illegible]#  1050#  1051#  1054#  1055#  1064   1071#  1072#  1100#  1101   1106#  1131#
                       1135#  1140#  1141#  1144#  1145#  1154   1161#  1162#  1190#  1191   1196#  1221#  1225#
                       1230#  1231#  1234#  1235#  1244   1251#  12[illegible]3#  12[illegible]0#  1281   12[illegible]6#  1311#  1315#  13[illegible]0#
                       1321#  13[illegible]4#  13[illegible]6#  1334   1341#  1342#  13[illegible]0#  1371   1376#  13[illegible]4#  1396#  1400#  14[illegible]3#
                       1410#  1414#  14[illegible]5#  1444#  1446#  14[illegible]0#  1459#  14[illegible]0#  1464#  1465#  1496#  1498#  1502#
                       1511#  1518#  1528#  1535#  1573#  1575#  1579#  15[illegible]8#  1595#  1605#  1612#  2284#  2297#
```

```
                        2303#   2304#   2305#   2366#
TXDBUF  012100          1052#   1[illegible]2#  1142#   1152#   1232#   1242#   1322#   1332#   1406#   1420#   1456#   1470#   1508#
                        1546#   1565#   1623#   2268#   2368#
TX[illegible]ME= 000200  694#
TXINIT= 000100           685#
TYPDAT  00[illegible]2  2001    2008    2011#
TYPE  = 1[illegible]402  796#    842    1022    1646    1649    1654    1776    1786    1798    1802    1893    1930    2005
                        2009    2067    2093    2095    2103
TYPMSG  007172          1999    2002#
USER  = 000000           691#
W[illegible]D  = 000001  658#
[illegible]NT   00[illegible]754 1903#   1932#   1941#
[illegible]     00[illegible]340 1650#
[illegible]G    00[illegible]624 2093    2109#
SE    = 000012           552#    1039    1041#   1129    1131#   1219    1221#   1309    1311#   1393    1394#   1443    1444#
                        1493    1494#   1570    1571#
[illegible]W    007641  2097    2112#
[illegible]R    007631  2095    2110#
SN    = 000010           552#    1038    1041#   1128    1131#   1218    1221#   1308    1311#   1392    1394#   1442    1444#
                        1492    1494#   1569    1571#   1640#
SY    = 000015           784#    [illegible]92    794#    796#    798#    800#    802#    804#    806#    808#    810#    812#    814#
                        816#    816[illegible]
.     = 012340           696#    699#    705#    716#    2114#   2253#   2254#   2255#
.BEGIN  00[illegible]156 864     987#
.C[illegible]R  007476   815     2083#
.CNTLU  007552           817     2[illegible]4#
.CNVRT  006542           811     18[illegible]5#
.CONVR  006536           809     1893#
.EOP    00[illegible]364 1570    16[illegible]#
.ERRTA  012230           1994    2347#
.HLT    007062           702     1979#
.INSTE  006176           801     1796#
.INSTG  006202           1795    17[illegible]#
.INSTR  006060           799     177[illegible]#
.INSTI  006100           1776#   1787    1803
.MSG    006102           1774#   1777#   2089#   2090#   2091
.PHBOM  00[illegible]220 803     1810#
.PFAIL  007732           700     8[illegible]0   2038#   2059
.RESUB  00[illegible]604 807     1[illegible]#
.[illegible]    00[illegible]114 805     1[illegible]#
.[illegible]    00[illegible]174 793     1[illegible]#
.[illegible]    00[illegible]000 795     1[illegible]#
.[illegible]    00[illegible]72  813     1[illegible]#   1962
.[illegible]    012[illegible]0  712     8[illegible]#   838     1658
.TMPSR  007[illegible]0  704     19[illegible]#
.TMPTR  0012[illegible]  789#    1973
.TYPE   006020           797     1758#
```

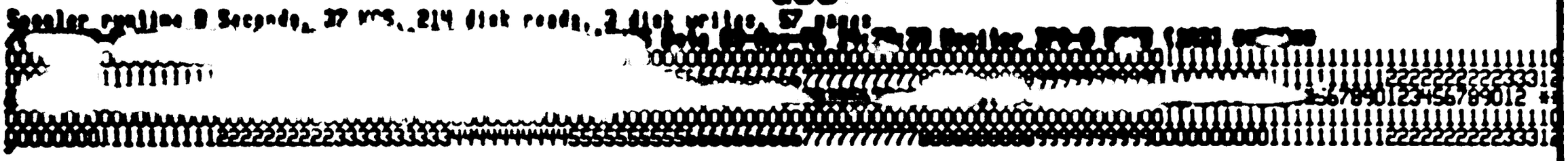
HLT	6128	1058	1076	1092	1091	1103	1109	1148	1156	1172	1181	1193	1199	1238	1246	
	1202	1271	1283	1289	1328	1336	1352	1361	1373	1379	1425	1430	1475	1480	1551	
	1528	1628	1633													
PCEND	128	1639														
PCRAT	128	553														
PCSYF	128	1511	1527	1588	1604											
PCTUP	128															
TCTUP	128	1394	1444	1496	1573											
SCIN	528	984														
SCOR	528	1395	1435	1485	1562											
[illegible]	128	2250														
[illegible]	128															
SCH	128	695														
SCLRVE	128	844														
SCONVR	128	1890														
SCP	128															
SCP	128	1639														
SCETFL	528	895	968	972	976	980	999									
SCETPA	128	874	885	905	931	940	1012									
SCTSY	128	950														
SCADE	128	553														
SCLT	128	1976														
SCINSTR	128	1769														
SCB	128															
SCTCH	128	1025	1115	1205	1295											
SCR	128															
SCRW	128															
[illegible]	528															
[illegible]	128	2115														
[illegible]	128	1807														
SCPHIL	128	2035														
SCPOKE	128	1050	1054	1070	1140	1144	1160	1230	1234	1250	1320	1324	1340	1409	1413	
	1429	1463														
SCPOWER	128															
SCRCT	128															
SCRC	128															
[illegible]	128	1866														
SCSET	128	1041	1106	1131	1196	1221	1286	1311	1376	1394	1396	1444	1446	1496	1498	
	1573	1575														
SCXACT	528															
SCOPE	528	1707														
SCOP1	128	1747														
SCTFL	128	1945														
SCTVE	128	[illegible]														
SCTART	128	820														
SCTEMP	128															
[illegible]	128	565														
[illegible]	128															
[illegible]	128	784														
[illegible]	128															
[illegible]	128	792	794	796	798	800	802	804	806	808	810	812	814	816		
[illegible]	128	1963														
STSTNO	128	1[illegible]8	1128	1218	1308	1392	1442	1492	1569							
STYPE	528	1755														
SUNTRU	128															
SVMMIA	528	715														

```
$WORDF    552#
$WORDO    552#
$WORDP    552#


. ABS.  012340     000


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DZDUFC,DZDUFC/CRF/SOL=HELLO.P11,PARA.P11,KEET.P11,DZDUFC.P11
 RUN-TIME: 16 23 2 SECONDS
 RUN-TIME RATIO: 117/42=2.7
 CORE USED:  18K  (35 PAGES)
```

GO5

Spooler runtime 0 Seconds, 37 ms, 214 disk reads, 2 disk writes, 57 pages