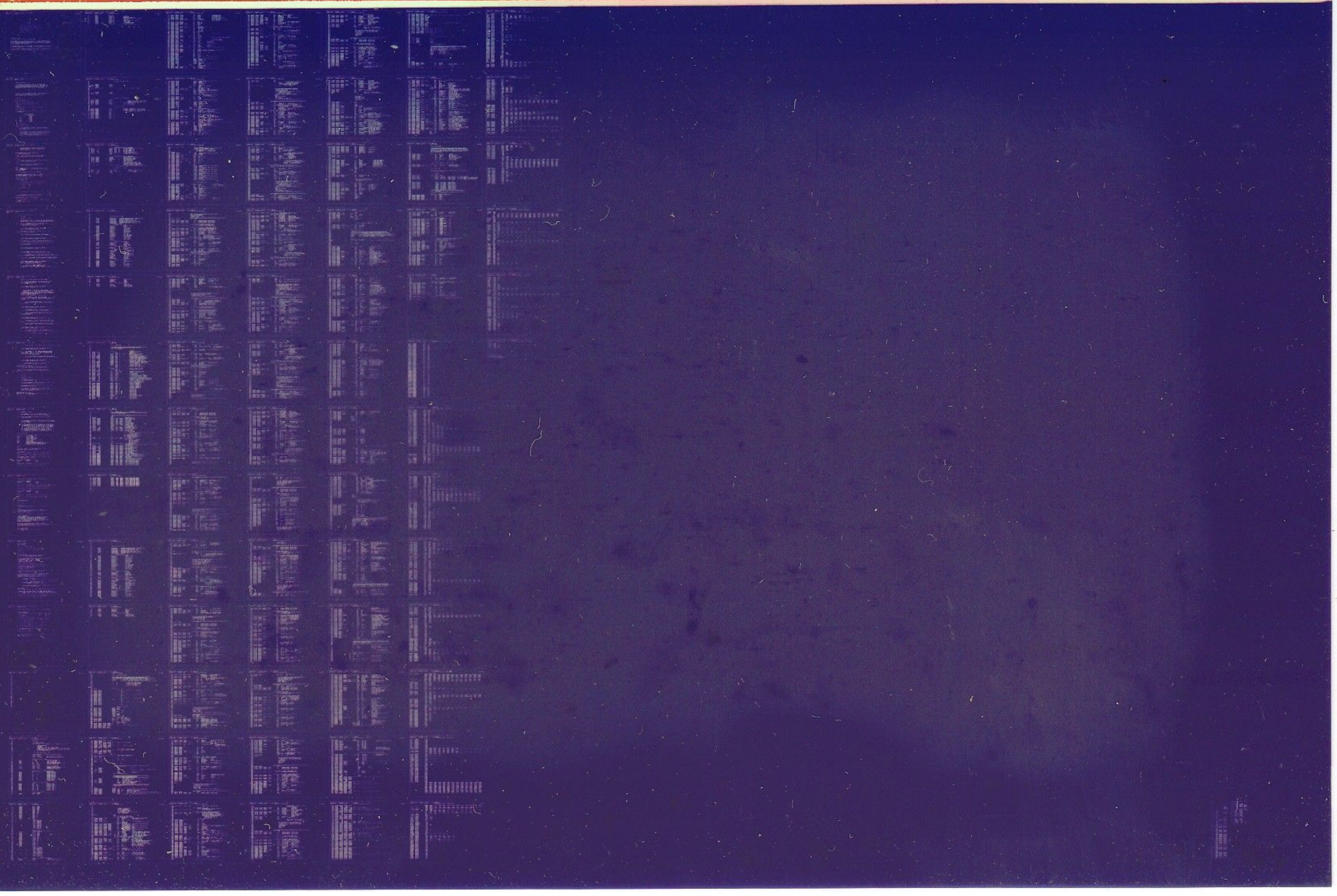
DUV-11

OFFLINE COMBINED TESTS
MD-11-DZDUV-B

EP-DZDUV-B-DL-B

FICHE 1 OF 1

DEC 1977
digital
MADE IN USA



```
.REM    *
```

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

PRODUCT CODE: MAINDEC-11-DZDUV-B-D

PRODUCT NAME: DUV11 OFFLINE COMBINED TESTS

RELEASE DATE: NOV. 1977

MAINTAINER : DIAGNOSTICS

```
*
.REM    *
```



```
*
```

.REM *

GENERAL DESCRIPTION

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

*
.REM *

1. THE DUV11 OFFLINE COMBINED TESTS VERIFY THAT THE TRANSMITTER
AND RECEIVER CAN TALK THRU THE EXTERNAL MODEM CABLE
PROVIDING THAT THE H315 CONNECTOR IS ON

*
.REM *

2. REQUIREMENTS

PDP-11/03 COMPUTER (LSI)

DUV11 SYNCRONOUS/ISOCRONOUS OPTION

ONE CONSOLE TELETYPE OR EQUIVALENT

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

3. LOADING PROCEDURE

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

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

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

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

NOTE: RUNNING UNDER APT-11, THERE IS A USER SWITCH REGISTER
CALLED "$USWR". IN ORDER TO BE FLEXIBLE ON THE AVAILIBILITY OF THE
H315 CONNECTOR, ONE BIT PASSES STATUS TO APT-11.
BIT 0 IN $USWR REFLECTS THIS STATUS, A 0 = CONNECTOR
PRESENT, A 1 = CONNECTOR NOT AVAILIBLE.

THE USER CHANGES THE CONTENTS OF THIS LOCATION
WHEN BUILDING THE E TABLE, BY ANSWERING THE
PROMPT "SWITCH 2".

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)

SW14=1
NOTE1: IN GENERAL SW01 WILL BE USED WHEN SW14=1 IS USED
NOTE2: WITHOUT SW01=1 "LOCK ON TEST" WILL DEFAULT TO TEST 1

4.2 STARTING ADDRESS

THE STARTING ADDRESS FOR ALL TESTS IS 000200

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

4.3 PROGRAM AND/OR OPERATOR ACTION

4.3.1 INITIAL PROGRAM START

4.3.1.1 LOAD PROGRAM INTO MEMORY WITH ABSOLUTE LOADER

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

4.3.1.3 TYPE 200G.

4.3.1.4 PROGRAM WILL START.

*
.REM *

4.3.1.5 THE PROGRAM WILL TYPE "DUV11 DZDUV-B TAPE F" (ONCE ONLY)

*
.REM *

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

4.3.2 PROGRAM RESTART WITH ALL SWITCHES DOWN

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

4.3.3 PROGRAM RESTART WITH SW00=1

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

4.3.3.2 TYPE 200G.

4.3.3.3 PROGRAM WILL START.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

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

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

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

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

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

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

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

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

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

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

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

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

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

4.3.3.19 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED

BY A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

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

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

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

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

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

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

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

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

4.3.4.2 TYPE 200G.

4.3.4.3 PROGRAM WILL START.

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

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

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

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

4.3.5 PROGRAM RESTART WITH SW14 =1
NOTE: THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED
SEE NOTE IN 4.3.4 FOR MORE DETAILS

4.3.5.1 SET SW14=1 IN SWITCH REG. (LOC. 176)

4.3.5.2 TYPE 200G.

4.3.5.3 PROGRAM WILL START.

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

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

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

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

5. OPERATING PROCEDURE

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

```
SW15 =1          HALT ON ERROR
SW14 =1          LOOP ON CURRENT TEST
SW13 =1          INHIBIT ERROR TYPEOUT
SW11 =1          INHIBIT ITERATIONS
SW10 =1          ESCAPE TO NEXT TEST ON ERROR
SW09 =1          LOOP ON ERROR
SW01 =1          RESTART PROGRAM AT SELECTED TEST
SW00 =1          RESELECT VECTOR AND CONTROL REGISTER ADRESSES
                 &PARAMETERS AFTER A PROGRAM RESTART
TO INHIBIT "END OF PASS" TYPEOUT - TURN TELETYPE OFF
```

6. ERRORS

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

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

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

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

6.1.2 PC +2 = REGISTER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING DEVICE REGISTER

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.3 PC +2 = RECEIVER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

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

WHERE YYYYYY IS THE EXPECTED DATA CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL DATA CONTENTS OF THAT REGISTER

6.1.4 PC +2 = TRANSMITTER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

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

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.5 ERROR DESCRIPTIONS
SEE LISTINGS FOR DETAILS OF ERRORS

6.2 ERROR RECOVERY

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

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

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

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

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

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

```
                END OF PASS TAPE Y
                16XXXX = DEVICE

                WHERE Y IS THE TAPE LOADED

                WHERE 16XXXX IS THE DEVICE'S BASE REGISTER ADDRESS

                TO INHIBIT THIS TYPEOUT - TURN TELETYPE OFF

        7.      RESTRICTIONS

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

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

        7.2     DISQUALIFYING DEVICES WHEN RUNNING MULTIPLE DEVICES

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

        7.2.1   IF DEVICE 0 IS TO BE DISQUALIFIED ,SIMPLY RESTART
                PROGRAM WITH SW00 =1 AND OMIT THE FIRST DEVICE.

        7.2.2   IF HOWEVER, DEVICES 1 THRU 15 OR ANY COMBINATION THEREOF
                ARE TO BE DISQUALIFIED....LOAD THE LOCATION OF ACTREG
                OBSERVE THE ACTIVE BITS (ACTIVE =1, NONACTIVE = 0)
                AND DEPOSIT 0 WHERE THOSE DEVICES ARE TO BE DISQUALIFIED

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

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

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

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

        7.3.1   TO PROVIDE SUFFICIENT DELAY FOR CLOCK SIGNAL OVER THE CABLE,
                LOCATION "HOLD:" MUST BE MODIFIED TO ACCOMODATE FOR FASTER MACHINES.
                        PRESENTLY "HOLD:" =20  IS SUFFICIENT TIME ON AN 11/03 MACHINE.

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

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

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


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

                VECTOR ADDRESS-                                 DURIV:  770

                ARE YOU RUNNING MULTIPLE DEVICES ?- NO          MULTD:  0

                LAST DEVICE:  RECEIVER CONTROL REGISTER ADDRESS-  LASTADD:  0

                # OF SYNC CHARS SELECTED -  2          SYNCNO:  377

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

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

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

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

        9.      PROGRAM DESCRIPTION
                                                                        *
                                                                .REM    *
        9.1     THIS PROGRAM PERFORMS THE OFFLINE COMBINED (TRANSMITTER & RECEIVER)
                CABLE TESTING OF THE DEVICE
                SEE LISTING FOR DETAILS
                                                                *
                                                                .REM    *
        10.     FLOW CHARTS:  RECEIVER FLOW, TRANSMITTER FLOW, TRANSMITTER & RECEIVER FLOW

        11.     LISTINGS
                                                                *
```

```
524
525
526
527
528
529
530
531
532
533
534
535
536             000001                  $TN=1
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
```

```
                        .ENABLE ABS

                        ;DUV11 DZDUV-B TAPE F
                        ;COPYRIGHT 1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

                                ;STARTING PROCEDURE
                                ;TYPE 200G
                                ;PROGRAM WILL TYPE "DUV11 DZDUV-B 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 F"
                                ;AND THEN RESUME TESTING

                        .SBTTL  BASIC DEFINITIONS

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

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

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

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

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

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

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

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

```
   670          000014                  TBITVEC=14              ;;"T" BIT
   671          000014                  TRTVEC= 14              ;;TRACE TRAP
   672          000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
   673          000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
   674          000024                  PWRVEC= 24              ;;POWER FAIL
   675          000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
   676          000034                  TRAPVEC=34              ;;"TRAP" TRAP
   677          000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
   678          000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
   679          000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
```

```
                                        ;STANDARD INTERRUPT VECTORS

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


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


                                        ;PROGRAM CONTROL PARAMETERS

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

                                        ;PROGRAM VARIABLES

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

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

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

                                        ;PROGRAM CONTROL FLAGS

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

```
                        ;INSTRUCTION DEFINITIONS

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

```
806          000040                  DNAINTE=BIT5            ;DNA INTR ENAB
807          000020                  SEND=BIT4               ;SEND
808          000010                  HDXEN=BIT3              ;HDX/FDX
809          000001                  BREAK=BIT0              ;BREAK
810                                          ;TXCSR WRD DEFINITIONS
811          000000                  USER=0                  ;USER MODE
812          004000                  MINT=4000               ;MAINT INT MODE
813          010000                  MEXT=10000              ;MAINT EXT MODE
814          014000                  SYSTST=14000            ;SYSTEM TEST MODE
```

```
                        .SBTTL  COMMON TAGS

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

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

```
                        ;SLF:    .ASCIZ  <12>            ;;LINE FEED
                        ;;*****************************************************************
                        .SBTTL  APT MAILBOX-ETABLE

                        ;;*****************************************************************
                        .EVEN
                        $MAIL:                          ;;APT MAILBOX
                        $MSGTY: .WORD   AMSGTY  ;;MESSAGE TYPE CODE
                        $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
                        $TESTN: .WORD   ATESTN  ;;TEST NUMBER
                        $PASS:  .WORD   APASS   ;;PASS COUNT
                        $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
                        $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
                        $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
                        $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
                        $ETABLE:                        ;;APT ENVIRONMENT TABLE
                        $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
                        $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
                        $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
                        $USWR:  .WORD   AUSWR   ;;USER SWITCHES
                        $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
                        ;*                      BITS 15-11=CPU TYPE
                        ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                        ;*                              11/70=06,PDQ=07,Q=10
                        ;*                      BIT 10=REAL TIME CLOCK
                        ;*                      BIT  9=FLOATING POINT PROCESSOR
                        ;*                      BIT  8=MEMORY MANAGEMENT
                        $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S.  BYTE
                        $MTYP1: .BYTE   AMTYP1  ;;MEM.  TYPE,BLK#1
                        ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
                        ;*                              900 NSEC CORE=001
                        ;*                              300 NSEC BIPOLAR=002
                        ;*                              500 NSEC MOS=003
                        $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                        ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
                        $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S.  BYTE
                        $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
                        $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
                        $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
                        $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
                        $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
                        $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
                        $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
                        $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
                        $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
                        $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
                        $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
                        $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
                        $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
                        $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
                        $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
                        $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
                        $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
                        $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
                        $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
                        $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
```

```
927  001626  000000                  $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
928  001630  000000                  $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
929  001632  000000                  $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
930  001634  000000                  $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
931  001636  000000                  $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
932  001640  000000                  $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
933  001642  000000                  $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
934  001644  000000                  $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
935  001646  000000                  $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
936  001650  000000                  $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
937
938
939  001652                          $ETEND:
940
941
942
```

```
                        ;INSTRUCTION DEFINITIONS

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

```
  999          000040                  DNAINTE=BIT5            ;DNA INTR ENAB
 1000          000020                  SEND=BIT4               ;SEND
 1001          000010                  HDXEN=BIT3              ;HDX/FDX
 1002          000001                  BREAK=BIT0              ;BREAK
 1003                                          ;TXCSR WRD DEFINITIONS
 1004          000000                  USER=0                  ;USER MODE
 1005          004000                  MINT=4000               ;MAINT INT MODE
 1006          010000                  MEXT=10000              ;MAINT EXT MODE
 1007          014000                  SYSTST=14000            ;SYSTEM TEST MODE
```

```
1008                                    .SBTTL  ERROR POINTER TABLE
1009
1010                                    ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
1011                                    ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
1012                                    ;*LOCATION $ITEMB.  THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
1013                                    ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
1014                                    ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
1015
1016                                    ;*      EM              ;;POINTS TO THE ERROR MESSAGE
1017                                    ;*      DH              ;;POINTS TO THE DATA HEADER
1018                                    ;*      DT              ;;POINTS TO THE DATA
1019                                    ;*      DF              ;;POINTS TO THE DATA FORMAT
1020
1021
1022  001652                            $ERRTB:
1023                                            ;ERROR TABLE
1024  001652  001762                            EM1     ;ERROR 1        REGISTER ERROR
1025  001654  002067                            DH1
1026  001656  002116                            DT1
1027  001660  002132                            DF1
1028  001662  002022                            EM2     ;ERROR 2        RECEIVER ERROR
1029  001664  002067                            DH1
1030  001666  002116                            DT1
1031  001670  002132                            DF1
1032  001672  002043                            EM3     ;ERROR 3        TRANSMITTER ERROR
1033  001674  002067                            DH1
1034  001676  002116                            DT1
1035  001700  002132                            DF1
1036  001702  001746                            EM4     ;ERROR 4        BIT ERROR (GENERAL)
1037  001704  000000                            0
1038  001706  002126                            DT4
1039  001710  002132                            DF1
1040
1041                                            ;DEFAULT DU ADDRESSES
1042  001712  160010                    RXCSR:  160010
1043  001714  160011                    HRXCSR: 160011
1044  001716  160012                    RXDBUF: 160012
1045  001720  160013                    HRXDBUF:        160013
1046  001722  160012                    PARCSR: 160012
1047  001724  160013                    HPARCSR:        160013
1048  001726  160014                    TXCSR:  160014
1049  001730  160015                    HTXCSR: 160015
1050  001732  160016                    TXDBUF: 160016
1051  001734  160017                    HTXDBUF:        160017
1052                                            ;DEFAULT DU VECTORS
1053  001736  000770                    DURIV:  770     ;REC INTR VECTOR
1054  001740  000772                    DURIS:  772     ;REC INTR STATUS
1055  001742  000774                    DUTIV:  774     ;XMIT INTR VECTOR
1056  001744  000776                    DUTIS:  776     ;XMIT INTR STATUS
1057                                            ;ERROR MESSAGES
1058  001746  020040  051105  047522    EM4:    .ASCIZ  /  ERROR PC /
1059  001754  020122  041520  000040
1060  001762  020040  047503  050115    EM1:    .ASCIZ /  COMPARISON ERROR ON REGISTERS/
1061  001770  051101  051511  047117
1062  001776  042440  051122  051117
1063  002004  047440  020116  042522
```

```
1064  002012  044507  052123  051105
1065  002020  000123
1066  002022  020040  042522  042503  EM2:    .ASCIZ  /  RECEIVER ERROR/
1067  002030  053111  051105  042440
1068  002036  051122  051117     000
1069  002043     040  052040  040522  EM3:    .ASCIZ  /  TRANSMITTER ERROR/
1070  002050  051516  044515  052124
1071  002056  051105  042440  051122
1072  002064  051117     000
1073                                  ;DATA HEADERS FOR ERROR MESSAGES
1074  002067     105  051122  041520  DH1:    .ASCIZ  /ERRPC  WANTED  ACTUAL/
1075  002074  020040  040527  052116
1076  002102  042105  020040  041501
1077  002110  052524  046101     000
1078          002116                  .EVEN
1079                                  ;DATA TABLES FOR ERROR MESSAGES
1080  002116  001416  001130  001132  DT1:    .WORD   $ERRPC,HLD0,HLD1,0
1081  002124  000000
1082
1083  002126  001416  000000          DT4:    .WORD   $ERRPC,0
1084
1085  002132     000     000     000  DF1:    .BYTE   0,0,0,0
1086  002135     000
1087                                  .EVEN
1088                                  .SBTTL  ACT11 HOOKS
1089
1090                                  ;;**********************************************************************
1091                                  ;HOOKS REQUIRED BY ACT11
1092          002136                          $SVPC=.                 ;SAVE PC
1093          000046                          .=46
1094  000046  012674                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1095          000052                          .=52
1096  000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
1097          002136                          .=$SVPC                 ;; RESTORE PC
1098                                  .SBTTL  APT PARAMETER BLOCK
1099
1100                                  ;;**********************************************************************
1101                                  ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1102                                  ;;**********************************************************************
1103          002136                          .$X=.   ;;SAVE CURRENT LOCATION
1104          000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1105  000024  000200                          200     ;;FOR APT START UP
1106          000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
1107  000044  002136                          $APTHDR ;;POINT TO APT HEADER BLOCK
1108          002136                          .=.$X   ;;RESET LOCATION COUNTER
1109                                  ;;**********************************************************************
1110                                  ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1111                                  ;INTERFACE SPEC.
1112
1113  002136                          $APTHD:
1114  002136  000000                 $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1115  002140  001526                 $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
1116  002142  000010                 $TSTM:  .WORD   10      ;;RUN TIM OF LONGEST TEST
1117  002144  000010                 $PASTM: .WORD   10      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1118  002146  000000                 $UNITM: .WORD           ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1119  002150  000052                         .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
1120
1121
1122                                        ;PROGRAM INITIALIZATION
1123                                        ;LOCK OUT INTERRUPTS
1124                                        ;SET UP PROCESSOR STACK
1125                                        ;SET UP POWER FAIL VECTOR
1126                                        ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1127                                        ;TYPE TITLE MESSAGE
1128
1129  002152                          .START:
1130                                  .SBTTL  INITIALIZE THE COMMON TAGS
1131                                  ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1132  002152  012706  001400                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1133  002156  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1134  002160  022706  001440                  CMP     #SWR,R6 ;;DONE?
1135  002164  001374                          BNE     .-6             ;;LOOP BACK IF NO
1136  002166  012706  001100                  MOV     ##STACK,SP              ;;SETUP THE STACK POINTER
1137                                  ;;INITIALIZE A FEW VECTORS
1138  002172  012737  016320  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1139  002200  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1140  002206  012737  014210  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1141  002214  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1142  002222  012737  016654  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1143  002230  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
1144  002236  012737  015012  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1145  002244  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1146  002252  005067  177234                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1147  002256  005067  177232                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1148  002262  112767  000001  177125          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1149  002270  012767  002270  177110          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1150  002276  012767  002276  177104          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1151                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER.  IF NOT FOUND OR IT IS
1152                                  ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1153  002304  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1154  002310  012737  002344  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1155  002316  012767  177570  177114          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1156  002324  012767  177570  177110          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1157  002332  022777  177777  177100          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1158  002340  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1159                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
1160  002342  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
1161  002344  012716  002352          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1162  002350  000002                          RTI
1163  002352  012767  000176  177060  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1164  002360  012767  000174  177054          MOV     #DISPREG,DISPLAY
1165  002366  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1166
1167  002372  005067  177136                  CLR     $PASS           ;;CLEAR PASS COUNT
1168  002376  132767  000200  177143          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
1169  002404  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
1170  002406  012767  001550  177024          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1171  002414                          67$:
1172  002414  012706  001100                  MOV     #STACK,SP               ;SET STACK
1173  002420  106427  000340                  MTPS    #340                    ;LOCK INTERRUPTS
1174  002424  012737  015012  000024          MOV     #.PFAIL,@#24            ;SET UP POWER FAIL VECTOR
1175  002432  105067  176535                  CLRB    STFLG                   ;CLEAR START FLAG
```

```
1176  002436  005067  176450                 CLR     PASCNT                  ;CLEAR PASS COUNT
1177  002442  105067  176735                 CLRB    $ERFLG                  ;CLEAR ERROR FLAG
1178  002446  005067  176740                 CLR     $ERTTL                  ;CLEAR ERROR COUNT
1179  002452  005067  176740                 CLR     $ERRPC                  ;CLEAR LAST EPROR POINTER
1180  002456  012767  000001  176716         MOV     #1,$TSTNM               ;SET UP FOR TEST 1
1181  002464  012767  002152  176412         MOV     #.START,RETURN          ;SET UP FOR POWER FAIL BEFORE
1182                                                                         ;TESTING STARTS
1183  002472  013746  000006                 MOV     @#6,-(SP)
1184  002476  013746  000004                 MOV     @#4,-(SP)
1185  002502  012737  002516  000004         MOV     #1$,@#4
1186  002510  005777  176724                 TST     @SWR
1187  002514  000407                         BR      2$
1188  002516  012767  000176  176714  1$:    MOV     #SWREG,SWR
1189  002524  012767  000174  176710         MOV     #DISPREG,DISPLAY
1190  002532  022626                         CMP     (SP)+,(SP)+
1191  002534  012637  000004          2$:    MOV     (SP)+,@#4
1192  002540  012637  000006                 MOV     (SP)+,@#6
1193  002544  022767  000176  176666         CMP     #SWREG,SWR
1194  002552  001007                         BNE     3$
1195  002554  005737  000042                 TST     @#42                    ;CHECK FOR CHAIN
1196  002560  001402                         BEQ     33$
1197  002562  000167  000522                 JMP     .BEGIN
1198  002566  004767  010204          33$:   JSR     PC,CNTLU
1199  002572  105767  176374          3$:    TSTB    INIFLG                  ;HAS INITIALIZATION BEEN PERFORMED
1200  002576  001004                         BNE     ONCE
1201  002600  104401  015152                 TYPE    ,MTITLE                 ;TYPE TITLE MESSAGE
1202  002604  105167  176362                 COMB    INIFLG                  ;IF NOT SET FLAG AND DO
1203  002610  105767  176732          ONCE:  TSTB    $ENV            ;APT CONTROL?
1204  002614  001410                         BEQ     11$             ;BR IF NO
1205  002616  032767  000001  176726         BIT     #1,$USWR        ;EXTENAL JUMPER ON?
1206  002624  001002                         BNE     12$             ;NO
1207  002626  105067  176321                 CLRB    JMRBY           ;CLEAR FLAG
1208  002632  000167  000452          12$:   JMP     .BEGIN          ;GO DO IT
1209  002636  032777  000001  176574  11$:   BIT     #SW00,@SWR      ;RESELECT VECTOR & CONTROL REG?
1210  002644  001002                         BNE     1$
1211  002646  000167  000436                 JMP     .BEGIN
1212  002652  012700  000300          1$:    MOV     #300,R0 ;RESTORE VECTOR AREA TO TRAPCATCHER
1213  002656  012701  000302                 MOV     #302,R1 ;START AT LOCATION 300
1214  002662  012702  000004                 MOV     #4,R2
1215  002666  010110                  2$:    MOV     R1,(R0)
1216  002670  005011                         CLR     (R1)
1217  002672  060200                         ADD     R2,R0
1218  002674  060201                         ADD     R2,R1
1219  002676  022701  001000                 CMP     #1000,R1        ;END AT LOCATION 776
1220  002702  002771                         BLT     2$
1221  002704  104406                         INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1222  002706  015220                         MREGAD  ;MESSAGE
1223  002710  104410                         PARAM   ;CONVERT STRING
1224  002712  160000                         160000  ;LOW LIMIT
1225  002714  167776                         167776  ;HIGH LIMIT
1226  002716  017150                         DUBASE  ;STORE AT THIS LOCATION
1227  002720     001                  .BYTE   1       ;MASK
1228  002721     001                  .BYTE   1       ;HOW MANY TIMES + 2
1229  002722  016767  014222  176226         MOV     DUBASE,KEEPADD  ;SAVE
1230  002730  004767  014062                 JSR     PC,DUADDR
1231  002734  016767  176216  176212         MOV     KEEPADD,BASEADD ;RESTORE FOR ROTATION
```

```
1232  002742  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1233  002744  015205                          MVECTO  ;MESSAGE
1234  002746  104410                          PARAM   ;CONVERT STRING
1235  002750  000300                          300     ;LOW LIMIT
1236  002752  000776                          776     ;HIGH LIMIT
1237  002754  001736                          DURIV   ;STORE AT THIS LOCATION
1238  002756     001                  .BYTE   1       ;MASK
1239  002757     004                  .BYTE   4       ;HOW MANY TIMES + 2
1240  002760  016767  176752  176176          MOV     DURIV,KEEPIV    ;SAVE
1241  002766  016767  176744  176166          MOV     DURIV,BASEIV    ;SET UP FOR ROTATION
1242  002774  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1243  002776  015250                          MMULT   ;MESSAGE
1244  003000  104414                          SETFLG  ;SET FLAG BASED UPON INPUT STRING
1245  003002  001152                          MULTD   ;THIS FLAG
1246  003004  105767  176142                  TSTB    MULTD   ;ARE THERE MULTIPLE DEVICES
1247                                                          ;ON THE SYSTEM ?
1248  003010  100406                          BMI     BBB     ;YES, ASK NEXT QUESTION
1249  003012  005067  176150                  CLR     ACTREG
1250  003016  005067  176146                  CLR     ROTADD
1251  003022  000167  000140                  JMP     OUTMUL  ;JUMP AROUND NEXT QUESTION
1252  003026                          BBB:
1253  003026  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1254  003030  015277                          MLASTD  ;MESSAGE
1255  003032  104410                          PARAM   ;CONVERT STRING
1256  003034  160000                          160000  ;LOW LIMIT
1257  003036  167776                          167776  ;HIGH LIMIT
1258  003040  001160                          LASTADD ;STORE AT THIS LOCATION
1259  003042     001                  .BYTE   1       ;MASK
1260  003043     001                  .BYTE   1       ;HOW MANY TIMES + 2
1261                                  ;THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
1262  003044  012767  000001  176116  1$:     MOV     #1,ROTADD       ;SET UP POINTER
1263  003052  005067  176110                  CLR     ACTREG  ;CLR ACTIVE REGISTER
1264  003056  056767  176106  176102  2$:     BIS     ROTADD,ACTREG   ;MAKE THIS DEVICE ACTIVE
1265  003064  000241                          CLC
1266  003066  006167  176076                  ROL     ROTADD  ;SET UP POINTER
1267  003072  103421                          BCS     3$      ;ARE YOU OUT OF RANGE ?
1268  003074  062767  000010  176052          ADD     #10,BASEADD     ;SET UP BASE ADDRESS
1269  003102  026767  176052  176044          CMP     LASTADD,BASEADD ;IS THIS THE LAST DEVICE ?
1270  003110  101362                          BHI     2$      ;NO DO IT AGAIN
1271  003112  056767  176052  176046          BIS     ROTADD,ACTREG   ;THIS ASSUMES THAT THERE ARE AT
1272                                                          ;LEAST TWO DEVICES WHEN YOU ANSWER YES TO
1273                                                          ;MULTIPLE DEVICE QUESTION
1274  003120  012767  000001  176042  4$:     MOV     #1,ROTADD       ;SET UP FOR LATER USE IN END OF PASS ROUTINE
1275  003126  016767  176024  176020          MOV     KEEPADD,BASEADD ;DITTO
1276  003134  000414                          BR      OUTMUL  ;CONTINUE QUESTIONS
1277  003136  016767  176014  176010  3$:     MOV     KEEPADD,BASEADD ;RESTORE
1278  003144  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1279  003146  015373                          MRANGE  ;MESSAGE
1280  003150  104410                          PARAM   ;CONVERT STRING
1281  003152  160000                          160000  ;LOW LIMIT
1282  003154  167776                          167776  ;HIGH LIMIT
1283  003156  001160                          LASTADD ;STORE AT THIS LOCATION
1284  003160     001                  .BYTE   1       ;MASK
1285  003161     001                  .BYTE   1       ;HOW MANY TIMES + 2
1286  003162  000167  177656                  JMP     1$      ;DO IT AGAIN
1287  003166  012767  000340  013616  OUTMUL: MOV     #340,DUPRT
```

```
1288  003174  004767  013542                    JSR     PC,DULEV
1289                                            ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1290                                            ;BUFFER TO THE CHARACTERS "1" AND "2".
1291                                            ;IF THE CHARACTER IS "1" CLEAR THE FLAG
1292                                            ;IF THE CHARACTER IS "2" SET THE FLAG
1293  003200                            AAA:
1294  003200  104406                           INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1295  003202  015611                           MSYNC   ;MESSAGE
1296  003204  122767  000061  012740  3$:      CMPB    #'1,INBUF       ;IS IT "1" ?
1297  003212  001003                           BNE     1$
1298  003214  105067  175726                  CLRB    SYNCNO  ;000
1299  003220  000412                           BR      4$
1300  003222  122767  000062  012722  1$:      CMPB    #'2,INBUF       ;IS IT "2" ?
1301  003230  001004                           BNE     2$
1302  003232  112767  177777  175706          MOVB    #-1,SYNCNO      ;377
1303  003240  000402                           BR      4$
1304  003242  104407                    2$:    INSTER  ;RETRY
1305  003244  000757                           BR      3$
1306  003246  000240                    4$:    NOP
1307  003250  104406                           INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1308  003252  015657                           MWIRE6  ;MESSAGE
1309  003254  104414                           SETFLG  ;SET FLAG BASED UPON INPUT STRING
1310  003256  001147                           SEXMIT  ;THIS FLAG
1311  003260  104406                           INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1312  003262  015730                           MWIRE5  ;MESSAGE
1313  003264  104414                           SETFLG  ;SET FLAG BASED UPON INPUT STRING
1314  003266  001150                           SEREC   ;THIS FLAG
1315  003270  104406                           INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1316  003272  016000                           MWIRE4  ;MESSAGE
1317  003274  104414                           SETFLG  ;SET FLAG BASED UPON INPUT STRING
1318  003276  001151                           OPTCLR  ;THIS FLAG
1319  003300  104406                           INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1320  003302  016057                           MEXTJ   ;MESSAGE
1321  003304  104414                           SETFLG  ;SET FLAG BASED UPON INPUT STRING
1322  003306  001153                           JMRBY   ;THIS FLAG
1323
1324                                            ;TEST START AND RESTART
1325
1326  003310  012706  001100          .BEGIN:  MOV     #STACK,SP               ;SET UP STACK
1327  003314  106427  000340                   MTPS    #340                    ;LOCK OUT INTERRUPTS
1328  003320  032777  000002  176112           BIT     #SW01,@SWR              ;IF SW01=1, GET STARTING PC
1329  003326  001413                           BEQ     3$
1330  003330  104406                           INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1331  003332  015543                           MTSTPC  ;MESSAGE
1332  003334  104410                           PARAM   ;CONVERT STRING
1333  003336  003374                           TST1    ;LOW LIMIT
1334  003340  017500                           17500   ;HIGH LIMIT
1335  003342  001402                           $TSTNM  ;STORE AT THIS LOCATION
1336  003344     001                   .BYTE   1       ;MASK
1337  003345     001                   .BYTE   1       ;HOW MANY TIMES + 2
1338  003346  016767  176030  175530           MOV     $TSTNM,RETURN
1339  003354  000403                           BR      4$
1340  003356  012767  003374  175520  3$:      MOV     #TST1,RETURN            ;START AT TEST 1
1341  003364  104401  015537          4$:      TYPE    ,MR                     ;TYPE R
1342  003370  000177  175510                   JMP     @RETURN                 ;START TESTING
1343
```

```
                                        ;;THIS TEST VERIFYS THAT RXDONE CAUSES AN INTERRUPT
                                        ;;MODE:  SYNC EXTERNAL
                                        ;;INTERRUPT VECTOR:  DURIV
                                        ;;LENGTH:  EIGHT
                                        ;;*****************************************************************
003374  000004                        TST1:   SCOPE

003376  052777  000400  176322                BIS     #MRESET,@TXCSR  ;MASTER RESET
003404  012777  020000  176310                MOV     #SYNEXT,@PARCSR ;SET THE MODE
003412  052777  000400  176306                BIS     #MRESET,@TXCSR  ;MASTER RESET

                                        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
003420  012777  064001  176300                MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                        ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
003426  012777  026026  176266                MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
003434  052777  000020  176250                BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
                                        ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
003442  042777  020000  176256                BIC     #CLK,@TXCSR     ;POKE CLK DOWN
003450  052777  020000  176250                BIS     #CLK,@TXCSR     ;POKE CLK UP
003456  012777  003500  176252                MOV     #1$,@DURIV      ;SET UP TRAPCATCHER
003464  016777  013322  176246                MOV     DUPRT,@DURIS    ;
003472  106427  000000                        MTPS    #0      ;ALLOW INTERRUPTS
003476  000424                                BR      2$      ;JUMP AROUND INTERRUPT SVC ROUTINE
                                        ;THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
003500  106427  000340                1$:     MTPS    #340    ;DON'T ALLOW ANYMORE INTERRUPTS
003504  042777  000100  176200                BIC     #RINTEN,@RXCSR  ;CLEAR INTERRUPT ENABLE
003512  105777  176174                        TSTB    @RXCSR          ;RXDONE=1?
003516  100401                                BMI     .+4
003520  104004                                ERROR   4                       ;FALSE INTERRUPT
003522  012716  003714                        MOV     #3$,(SP)        ;SET UP RETURN LOCATION
003526  016777  176206  176202                MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
003534  012777  000000  176176                MOV     #0,@DURIS       ;
003542  017701  176150                        MOV     @RXDBUF,R1      ;CLEAR INTERRUPT
003546  000002                                RTI

003550  052777  000100  176134        2$:     BIS     #RINTEN,@RXCSR  ;SET INTERRUPT ENABLE
003556  012767  000010  175336                MOV     #8.,SHIFT       ;# OF SHIFTS
003564  012767  000025  175706                MOV     #25,$TMP1       ;TO BE SHIFTED CHARACTER
                                        ;THE FOLLOWING POKES THE MAINT DATA BASED UPON THE
                                        ;INFORMATION CONTAINED IN $TMP1 AND IT IS
                                        ;SHIFTED IN BY THE CONTENTS OF SHIFT
003572  042777  040000  176126        5$:     BIC     #MTDATA,@TXCSR
003600  000241                                CLC
003602  006067  175672                        ROR     $TMP1   ;FORCE CARRY
003606  103003                                BCC     4$
003610  052777  040000  176110                BIS     #MTDATA,@TXCSR
003616  042777  020000  176102        4$:     BIC     #CLK,@TXCSR
003624  052777  020000  176074                BIS     #CLK,@TXCSR
003632  005367  175264                        DEC     SHIFT
003636  001355                                BNE     5$
                                        ;INTERRUPT SHOULD NOW OCCUR
003640  005000                                CLR     R0
003642  005200                                INC     R0              ;WAIT FOR INTERRUPT
003644  001376                                BNE     .-2
```

```
1400  003646  016777  176066  176062          MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
1401  003654  012777  000000  176056          MOV     #0,@DURIS       ;
1402  003662  016703  176030                  MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
1403  003666  012700  000025                  MOV     #25,R0          ;EXPECTED
1404  003672  017701  176020                  MOV     @RXDBUF,R1
1405  003676  042777  000100  176006          BIC     #RINTEN,@RXCSR  ;CLR INTR ENABLE
1406  003704  020001                          CMP     R0,R1
1407  003706  001401                          BEQ     .+4
1408  003710  104002                          ERROR   2               ;CHARACTERS SHOULD COMPARE
1409  003712  104004                          ERROR   4                       ;INTERRUPT FAILED TO OCCUR
1410
1411  003714  106427  000340          3$:     MTPS    #340
1412
1413
1414
1415                                  ;;THIS TEST VERIFYS THAT TWO INTERRUPTS THAT TRAP TO
1416                                  ;;THE SAME VECTOR ARE BOTH EXECUTED
1417                                  ;;INTERRUPT VECTOR:  DURIV
1418                                  ;;THIS TEST ONLY WORKS IN MAINT EXTERNAL MODE
1419
1420                                  ;;*****************************************************************
1421  003720  000004                  TST2:   SCOPE
1422  003722  105767  175225                  TSTB    JMRBY           ;IN MAINT. EXTERNAL?
1423  003726  100402                          BMI      +6     ;IF ANSWER WAS YES DO THIS TEST
1424  003730  000167  000402                  JMP     1$      ;IF ANSWER WAS NO JUMP AROUND TEST
1425  003734  052777  000400  175764          BIS     #MRESET,@TXCSR  ;MASTER RESET
1426  003742  012777  020000  175752          MOV     #SYNEXT,@PARCSR ;SET THE MODE
1427  003750  052777  000400  175750          BIS     #MRESET,@TXCSR  ;MASTER RESET
1428
1429                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1430  003756  012777  064001  175742          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1431
1432                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1433  003764  012777  026026  175730          MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
1434  003772  052777  000020  175712          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
1435                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1436  004000  042777  020000  175720          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1437  004006  052777  020000  175712          BIS     #CLK,@TXCSR     ;POKE CLK UP
1438  004014  012777  004036  175714          MOV     #2$,@DURIV      ;SET UP TRAPCATCHER
1439  004022  016777  012764  175710          MOV     DUPRT,@DURIS    ;
1440  004030  106427  000000                  MTPS    #0      ;ALLOW INTERRUPT
1441  004034  000454                          BR      3$      ;JUMP AROUND SVC ROUTINE
1442
1443                                  ;THE FOLLOWING IS THE 1ST INTERRUPT SVC ROUTINE
1444  004036  106427  000340          2$:     MTPS    #340    ;DON'T ALLOW ANY MORE INTERRUPTS
1445  004042  105777  175644                  TSTB    @RXCSR          ;RXDONE = 1 ?
1446  004046  100401                          BMI     .+4
1447  004050  104004                          ERROR   4                       ;FALSE INTERRUPT
1448  004052  012716  004332                  MOV     #5$,(SP)        ;SET UP RETURN LOCATION
1449  004056  012777  004136  175652          MOV     #4$,@DURIV      ;SET UP TRAPCATCHER FOR SECOND
1450                                                                  ;INTERRUPT
1451  004064  052777  000002  175620          BIS     #DTR,@RXCSR     ;TRY TO CAUSE SECOND INTERRUPT
1452  004072  017701  175620                  MOV     @RXDBUF,R1      ;JUST READ RXDBUF TO CLR RXDONE
1453                                                                  ;TO ALLOW SECOND INTERRUPT
1454  004076  106427  000000                  MTPS    #0      ;ALLOW INTERRUPT
1455  004102  005000                          CLR     R0
```

```
1456  004104  005200                          INC     R0              ;WAIT FOR INTERRUPT
1457  004106  001376                          BNE     .-2
1458  004110  042777  000140  175574          BIC     #RINTEN!DSINTE,@RXCSR   ;CLR INTR ENABLES
1459  004116  104004                          ERROR   4                       ;2ND INTERRUPT FAILED TO OCCUR
1460
1461  004120  016777  175614  175610  6$:     MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
1462  004126  012777  000000  175604          MOV     #0,@DURIS       ;
1463  004134  000002                          RTI
1464
1465                                  ;THE FOLLOWING IS THE 2ND INTERRUPT SVC ROUTINE
1466  004136  106427  000340          4$:     MTPS    #340    ;DON'T ALLOW ANYMORE INTERRUPTS
1467  004142  005777  175544                  TST     @RXCSR          ;DSC = 1 ?
1468  004146  100401                          BMI     .+4
1469  004150  104004                          ERROR   4                       ;FALSE INTERRUPT
1470  004152  042777  000140  175532          BIC     #RINTEN!DSINTE,@RXCSR   ;CLR BOTH INTR ENABLES
1471  004160  012716  004120                  MOV     #6$,(SP)        ;SET UP RETURN LOCATION
1472  004164  000002                          RTI
1473
1474  004166  052777  000140  175516  3$:     BIS     #RINTEN!DSINTE,@RXCSR ;SET INTERRUPT ENABLES
1475  004174  012767  000010  174720          MOV     #8.,SHIFT       ;# OF SHIFTS
1476  004202  012767  000025  175270          MOV     #25,$TMP1
1477                                  ;THE FOLLOWING POKES THE MAINT DATA BASED UPON THE
1478                                  ;INFORMATION CONTAINED IN $TMP1 AND IT IS
1479                                  ;SHIFTED IN BY THE CONTENTS OF SHIFT
1480  004210  042777  040000  175510  8$:     BIC     #MTDATA,@TXCSR
1481  004216  000241                          CLC
1482  004220  006067  175254                  ROR     $TMP1   ;FORCE CARRY
1483  004224  103003                          BCC     7$
1484  004226  052777  040000  175472          BIS     #MTDATA,@TXCSR
1485  004234  042777  020000  175464  7$:     BIC     #CLK,@TXCSR
1486  004242  052777  020000  175456          BIS     #CLK,@TXCSR
1487  004250  005367  174646                  DEC     SHIFT
1488  004254  001355                          BNE     8$
1489                                  ;1ST INTERRUPT SHOULD NOW OCCUR
1490  004256  005000                          CLR     R0
1491  004260  005200                          INC     R0              ;WAIT FOR INTERRUPT
1492  004262  001376                          BNE     .-2             ;
1493  004264  016777  175450  175444          MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
1494  004272  012777  000000  175440          MOV     #0,@DURIS       ;
1495  004300  016703  175412                  MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
1496  004304  012700  000025                  MOV     #25,R0          ;EXPECTED
1497  004310  017701  175402                  MOV     @RXDBUF,R1
1498  004314  042777  000140  175370          BIC     #RINTEN!DSINTE,@RXCSR   ;CLR INTERRUPT ENABLES
1499  004322  020001                          CMP     R0,R1
1500  004324  001401                          BEQ     .+4
1501  004326  104002                          ERROR   2               ;CHARACTERS SHOULD COMPARE
1502  004330  104004                          ERROR   4                       ;INTERRUPT FAILED TO OCCUR
1503
1504  004332  106427  000340          5$:     MTPS    #340    ;DON'T ALLOW ANY MORE INTERRUPTS
1505  004336                          1$:
1506
1507                                  ;;THIS TEST VERIFYS THAT DNA CAUSES AN INTERRUPT
1508                                  ;;MODE:   SYNC EXTERNAL
1509                                  ;;INTERRUPT VECTOR:   DUTIV
1510
1511                                  ;;**********************************************************************
```

```
1512  004336  000004                    TST3:   SCOPE
1513
1514  004340  052777  000400  175360            BIS     #MRESET,@TXCSR  ;MASTER RESET
1515  004346  012777  020000  175346            MOV     #SYNEXT,@PARCSR ;SET THE MODE
1516  004354  052777  000400  175344            BIS     #MRESET,@TXCSR  ;MASTER RESET
1517
1518                                    ;SET MAINTENANCE MODE & SEND
1519                                            ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1520  004362  012777  004020  175336            MOV     #MINT!SEND,@TXCSR
1521
1522                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1523  004370  012777  026026  175324            MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
1524  004376  112777  000025  175326            MOVB    #25,@TXDBUF     ;LOAD CHARACTER
1525  004404  012767  000010  174510            MOV     #8.,SHIFT
1526                                    ;POKE CLK TO GET INTO SYNCRONIZATION
1527  004412  052777  020000  175306            BIS     #CLK,@TXCSR     ;POKE CLK UP
1528  004420  042777  020000  175300            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1529
1530  004426                                    1$:
1531  004426  052777  020000  175272            BIS     #CLK,@TXCSR     ;POKE CLK UP
1532  004434  042777  020000  175264            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1533  004442  005367  174454                    DEC     SHIFT           ;LAST SHIFT?
1534  004446  001367                            BNE     1$
1535  004450  012777  004516  175264            MOV     #2$,@DUTIV      ;SET UP TRAPCATCHER
1536  004456  016777  012330  175260            MOV     DUPRT,@DUTIS    ;
1537  004464  106427  000000                    MTPS    #0      ;ALLOW INTERUPTS
1538  004470  052777  000040  175230            BIS     #DNAINTE,@TXCSR ;ENABLE INTERRUPT
1539                                            ;NOW POKE CLK TO GET DNA
1540  004476  052777  020000  175222            BIS     #CLK,@TXCSR     ;POKE CLK
1541  004504  005000                            CLR     R0
1542  004506  005200                            INC     R0              ;WAIT FOR INTERRUPT
1543  004510  001376                            BNE     .-2
1544  004512  104004                            ERROR   4                       ;INTERRUPT FAILED TO OCCUR
1545  004514  000422                            BR      3$      ;JUMP AROUND SVC ROUTINE
1546                                    ;THE FOLLOWING IS THE INTERRUPT SERVICE ROUTINE
1547  004516  106427  000340            2$:     MTPS    #340    ;DON'T ALLOW ANYMORE INTERRUPTS
1548  004522  005777  175200                    TST     @TXCSR          ;DNA?
1549  004526  100401                            BMI     .+4
1550  004530  104004                            ERROR   4                       ;FALSE INTERRUPT
1551  004532  042777  000040  175166            BIC     #DNAINTE,@TXCSR ;CLR INTR ENABLE
1552  004540  012716  004604                    MOV     #4$,(SP)        ;SET UP RETURN LOCATION
1553  004544  016777  175174  175170            MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
1554  004552  012777  000000  175164            MOV     #0,@DUTIS       ;
1555  004560  000002                            RTI
1556
1557  004562  016777  175156  175152  3$:       MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
1558  004570  012777  000000  175146            MOV     #0,@DUTIS       ;
1559
1560  004576  042777  000040  175122            BIC     #DNAINTE,@TXCSR ;CLR INTERRUPT ENABLE
1561  004604  106427  000340            4$:     MTPS    #340    ;RESTORE NO INTERRUPT STATUS
1562
1563
1564                                    ;;THIS TEST VERIFYS THAT TXDONE CAUSES AN INTERRUPT
1565                                    ;;INTERRUPT VECTOR:  DUTIV
1566                                    ;;NOTE:  TXDONE = 1         AFTER A MASTER RESET
1567                                    ;;
```

```
1568                                    ;;**********************************************************************
1569  004610  000004                    TST4:   SCOPE
1570
1571  004612  052777  000400  175106            BIS     #MRESET,@TXCSR  ;MASTER RESET
1572  004620  012777  004666  175114            MOV     #1$,@DUTIV      ;SET UP TRAPCATCHER
1573  004626  016777  012160  175110            MOV     DUPRT,@DUTIS    ;
1574  004634  106427  000000                    MTPS    #0      ;ALLOW INTERPUTS
1575  004640  052777  000100  175060            BIS     #TXINTE,@TXCSR  ;ENABLE INTERRUPT
1576  004646  005000                            CLR     R0
1577  004650  005200                            INC     R0              ;WAIT FOR INTERRUPT
1578  004652  001376                            BNE     .-2             ;
1579  004654  042777  000100  175044            BIC     #TXINTE,@TXCSR  ;CLR INTERRUPT ENABLE
1580  004662  104004                            ERROR   4                       ;INTERRUPT FAILED TO OCCUR
1581  004664  000422                            BR      2$      ;JUMP AROUND SVC ROUTINE
1582
1583                                    ;THE FOLLOWING IS THE INTERRUPT SERVICE ROUTINE
1584  004666  106427  000340            1$:     MTPS    #340    ;DON'T ALLOW ANYMORE INTERRUPTS
1585  004672  042777  000100  175026            BIC     #TXINTE,@TXCSR  ;CLR INTR ENABLE
1586  004700  105777  175022                    TSTB    @TXCSR          ;TXDONE?
1587  004704  100401                            BMI     .+4
1588  004706  104004                            ERROR   4                       ;FALSE INTERRUPT
1589  004710  012716  004746                    MOV     #3$,(SP)        ;SET UP RETURN LOCATION
1590  004714  016777  175024  175020            MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
1591  004722  012777  000000  175014            MOV     #0,@DUTIS       ;
1592  004730  000002                            RTI
1593
1594  004732  016777  175006  175002  2$:       MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
1595  004740  012777  000000  174776            MOV     #0,@DUTIS       ;
1596
1597  004746  106427  000340            3$:     MTPS    #340    ;RESTORE NO INTERRUPT STATUS
1598
1599
1600
1601                                    ;;THIS TEST VERIFYS THAT TXDONE DOES NOT CAUSE AN INTERRUPT
1602                                    ;;WHEN PROCESSOR PRIORITY LEVEL IS TOO HIGH
1603                                    ;;INTERRUPT VECTOR: DUTIV
1604                                    ;;NOTE: TXDONE = 1 AFTER A MASTER RESET
1605                                    ;;
1606                                    ;;**********************************************************************
1607  004752  000004                    TST5:   SCOPE
1608  004754  052777  000400  174744            BIS     #MRESET,@TXCSR  ;MASTER RESET
1609  004762  012777  005046  174752            MOV     #1$,@DUTIV      ;SET UP TRAPCATCHER
1610  004770  016777  012016  174746            MOV     DUPRT,@DUTIS    ;
1611  004776  106427  000340                    MTPS    #340            ;SET PS LEVEL TOO HIGH
1612  005002  052777  000100  174716            BIS     #TXINTE,@TXCSR  ;ENABLE INTERRRUPT
1613  005010  005000                            CLR     R0      ;WAIT FOR INTERRUPT
1614  005012  005200                            INC     R0      ;
1615  005014  001376                            BNE     .-2     ;
1616  005016  042777  000100  174702            BIC     #TXINTE,@TXCSR  ;CLR INTR ENABLE
1617  005024  106427  000340                    MTPS    #340    ;DON'T ALLOW INTERRUPTS
1618  005030  016777  174710  174704            MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
1619  005036  012777  000000  174700            MOV     #0,@DUTIS       ;
1620  005044  000421                            BR      2$      ;TEST IS OK...GET OUT OF TEST
1621                                    ;THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
1622  005046  106427  000340            1$:     MTPS    #340    ;DONT ALLOW ANYMORE INTERRUPTS
1623  005052  042777  000100  174646            BIC     #TXINTE,@TXCSR  ;CLR INTR ENABLE
```

```
1624  005060  012716  005102                  MOV     #3$,(SP)        ;SET UP RETURN LOCATION
1625                                                                    ;TO REPORT ERROR
1626  005064  016777  174654  174650          MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
1627  005072  012777  000000  174644          MOV     #0,@DUTIS       ;
1628  005100  000002                          RTI
1629                                  ;END OF INTERRUPT SVC ROUTINE
1630
1631
1632
1633                                  ;YOU SHOULD NOT GET INTO THIS FOLLOWING CODE UNLESS THERE
1634                                  ;WAS AN ERROR
1635  005102  106427  000340          3$:     MTPS    #340    ;DON'T ALLOW ANYMORE INTERRUPTS
1636  005106  104004                          ERROR   4       ;INTERRUPT SHOULD NOT OF OCCURED,CHECK
1637                                                          ;THE INTERRUPT LEVEL SELECTED OR CHECK
1638                                                          ;INTERRUPT LOGIC OR BOTH
1639  005110                          2$:
1640
1641                                  ;;THIS TEST VERIFYS THAT TXDONE CAUSES ONLY ONE INTERRUPT
1642                                  ;;PROVIDING THAT TXCSR IS NOT READ
1643                                  ;;AND TXDBUF IS NOT LOADED (WRITTEN)
1644                                  ;;THIS TEST CHECKS THE ONCE ONLY FLIP/FLOP (V2)
1645                                  ;;OF THE INTERRUPT CONTROL LOGIC
1646                                  ;;INTERRUPT VECTOR: DUTIV
1647                                  ;;NOTE: TXDONE = 1    AFTER A MASTER RESET
1648                                  ;;
1649                                  ;;**********************************************************************
1650  005110  000004                  TST6:   SCOPE
1651  005112  052777  000400  174606          BIS     #MRESET,@TXCSR  ;MASTER RESET
1652  005120  012777  005160  174614          MOV     #1$,@DUTIV      ;SET UP TRAPCATCHER
1653  005126  016777  011660  174610          MOV     DUPRT,@DUTIS    ;
1654  005134  106427  000000                  MTPS    #0      ;ALLOW INTERRUPTS
1655  005140  052777  000100  174560          BIS     #TXINTE,@TXCSR  ;ENABLE INTR ENABLE
1656  005146  005000                          CLR     R0
1657  005150  005200                          INC     R0
1658  005152  001376                          BNE     .-2
1659  005154  104004                          ERROR   4       ;INTERRUPT FAILED TO OCCUR
1660  005156  000425                          BR      4$
1661                                  ;THE FOLLOWING IS THE INTR SVC ROUTINE
1662  005160  106427  000340          1$:     MTPS    #340    ;DON'T ALLOW ANYMORE INTR
1663  005164  012716  005224                  MOV     #3$,(SP)        ;SET UP RETURN LOCATION
1664  005170  012777  005200  174544          MOV     #2$,@DUTIV      ;SET UP TRAPCATCHER TO
1665                                                          ;PROVE THAT THE INTERRUPT DOES NOT OCCUR
1666                                                          ;TWICE (AFTER RTI 'ING FROM THIS
1667                                                          ;SVC ROUTINE
1668  005176  000002                          RTI
1669                                  ;THE FOLLOWING INTERRUPT SVC ROUTINE WILL CATCH THE SECOND INTR
1670  005200  106427  000340          2$:     MTPS    #340    ;DON'T ALLOW INTER
1671  005204  012716  005232                  MOV     #4$,(SP)        ;SET UP RETURN LOCATION
1672  005210  105777  174512                  TSTB    @TXCSR  ;TXDONE = 1?
1673  005214  100401                          BMI     .+4
1674  005216  104004                          ERROR   4       ;TXDONE SHOULD BE SET
1675  005220  104004                          ERROR   4       ;THE INTERRUPT WAS TAKEN TWICE.....
1676                                                          ;CHECK OUT THE V2 FLIP/FLOP LOGIC
1677                                                          ;IN THE INTERRUPT CONTROL LOGIC
1678  005222  000002                          RTI
1679  005224  005000          3$:     CLR     R0      ;ALLOW TIME TO CATCH SECOND
```

```
1680  005226  005200                          INC     R0              ;IF IT WERE TO OCCUR
1681  005230  001376                          BNE     .-2     ;
1682  005232  016777  174506  174502  4$:     MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
1683  005240  012777  000000  174476          MOV     #0,@DUTIS       ;
1684  005246  042777  000100  174452          BIC     #TXINTE,@TXCSR  ;CLR INTERRUPT ENABLE
1685  005254  106427  000340                  MTPS    #340    ;RESTORE NO INTERRUPT STATUS
1686
1687
1688                                  ;;THIS TEST VERIFYS THAT TWO INTERRUPTS THAT TRAP
1689                                  ;;TO THE SAME VECTOR ARE BOTH EXECUTED
1690                                  ;;INTERRUPT VECTOR:  DUTIV
1691                                  ;;MODE:  SYNC EXTERNAL
1692
1693                                  ;;**********************************************************************
1694  005260  000004                  TST7:   SCOPE
1695
1696  005262  052777  000400  174436          BIS     #MRESET,@TXCSR  ;MASTER RESET
1697  005270  012777  020000  174424          MOV     #SYNEXT,@PARCSR ;SET THE MODE
1698  005276  052777  000400  174422          BIS     #MRESET,@TXCSR  ;MASTER RESET
1699
1700                                  ;SET MAINTENANCE MODE & SEND
1701                                          ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1702  005304  012777  004020  174414          MOV     #MINT!SEND,@TXCSR
1703
1704                                  ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1705  005312  012777  026026  174402          MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
1706  005320  112777  000025  174404          MOVB    #25,@TXDBUF     ;LOAD CHARACTER
1707  005326  012767  000010  173566          MOV     #8.,SHIFT
1708                                  ;POKE CLK TO GET INTO SYNCRONIZATION
1709  005334  052777  020000  174364          BIS     #CLK,@TXCSR     ;POKE CLK UP
1710  005342  042777  020000  174356          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1711
1712  005350                                  1$:
1713  005350  052777  020000  174350          BIS     #CLK,@TXCSR     ;POKE CLK UP
1714  005356  042777  020000  174342          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1715  005364  005367  173532                  DEC     SHIFT           ;LAST SHIFT?
1716  005370  001367                          BNE     1$
1717  005372  012777  005432  174342          MOV     #2$,@DUTIV      ;SET UP TRAPCATCHER
1718  005400  016777  011406  174336          MOV     DUPRT,@DUTIS    ;
1719  005406  106427  000000                  MTPS    #0      ;ALLOW INTERRUPTS
1720  005412  052777  000140  174306          BIS     #TXINTE!DNAINTE,@TXCSR  ;ENABLE INTERRUPTS
1721  005420  005000                          CLR     R0
1722  005422  005200                          INC     R0              ;WAIT FOR INTERRUPT
1723  005424  001376                          BNE     .-2             ;
1724  005426  104004                          ERROR   4                       ;INTERRUPT FAILED TO OCCUR
1725  005430  000461                          BR      3$      ;JUMP AROUND SVC ROUTINES
1726
1727                                  ;THE FOLLOWING IS THE 1ST INTERRUPT SVC ROUTINE
1728  005432  106427  000340          2$:     MTPS    #340    ;DON'T ALLOW ANYMORE INTERRUPTS
1729  005436  005777  174264                  TST     @TXCSR  ;DNA=0 ?
1730  005442  100001                          BPL     .+4
1731  005444  104004                          ERROR   4       ;DNA SHOULD NOT BE ASSERTED
1732  005446  105777  174254                  TSTB    @TXCSR  ;TXDONE = 1?
1733  005452  100401                          BMI     .+4
1734  005454  104004                          ERROR   4               ;FALSE INTERRUPT
1735  005456  012716  005616                  MOV     #4$,(SP)        ;SET UP RETURN LOCATION
```

```
1736  005462  012777  005544  174252          MOV     #5$,@DUTIV      ;SET UP TRAPCATCHER
1737                                          ;NOW POKE CLK TO BRING UP DNA
1738  005470  052777  020000  174230          BIS     #CLK,@TXCSR     ;POKE CLK
1739  005476  112777  000025  174226          MOVB    #25,@TXDBUF     ;JUST LOAD ANY CHAR TO CLR
1740                                                  ;TXDONE TO ALLOW SECOND INTERRUPT
1741  005504  106427  000000                  MTPS    #0      ;ALLOW INTERRUPTS
1742  005510  005000                          CLR     R0
1743  005512  005200                          INC     R0              ;WAIT FOR INTERRUPT
1744  005514  001376                          BNE     .-2
1745  005516  042777  000140  174202          BIC     #DNAINTE!TXINTE,@TXCSR  ;CLR INTR ENABLES
1746  005524  104004                          ERROR   4                       ;2ND INTERRUPT FAILED TO OCCUR
1747
1748  005526  016777  174212  174206  6$:     MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
1749  005534  012777  000000  174202          MOV     #0,@DUTIS       ;
1750  005542  000002                          RTI
1751
1752                                  ;THE FOLLOWING IS THE 2ND INTERRUPT SVC ROUTINE
1753  005544  106427  000340          5$:     MTPS    #340
1754  005550  005777  174152                  TST     @TXCSR          ;DNA
1755  005554  100401                          BMI     .+4
1756  005556  104004                          ERROR   4                       ;FALSE INTERRUPT
1757  005560  042777  000140  174140          BIC     #DNAINTE!TXINTE,@TXCSR  ;CLR BOTH INTR ENABLES
1758  005566  012716  005526                  MOV     #6$,(SP)                ;SETUP RETURN LOCATION
1759  005572  000002                          RTI
1760
1761  005574  016777  174144  174140  3$:     MOV     DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
1762  005602  012777  000000  174134          MOV     #0,@DUTIS
1763
1764  005610  042777  000140  174110          BIC     #DNAINTE!TXINTE,@TXCSR ;CLR BOTH INTERRUPT
1765                                                                          ;ENABLES
1766  005616  106427  000340          4$:     MTPS    #340    ;RESTORE NO INTERRUPT STATUS
1767
1768
1769
1770                                  ;;THIS TEST VERIFYS CTP MODE (IE SYSTST MODE)
1771                                  ;;IT BASICALLY CHECKS THE EXISTANCE OF
1772                                  ;;THE FREE RUNNING OSCILLATOR
1773                                  ;;MODE:   SYNEXT
1774                                  ;;LENGTH:  EIGHT
1775                                  ;;THIS TEST USES BOTH THE RECEIVER & TRANSMITTER LOGIC
1776
1777                                  ;;**********************************************************************
1778  005622  000004                  TST10:  SCOPE
1779                                  ;**********************
1780  005624  000167  000262                  JMP     1$      ;NOP THIS TEST
1781                                  ;**********************
1782  005630  052777  000400  174070          BIS     #MRESET,@TXCSR  ;MASTER RESET
1783  005636  012777  020000  174056          MOV     #SYNEXT,@PARCSR ;LOAD THE MODE
1784  005644  052777  000400  174054          BIS     #MRESET,@TXCSR  ;MASTER RESET
1785  005652  012777  026026  174042          MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR  ;LOAD THE MODE,
1786                                                  ;# OF BITS PER CHAR,PARITY SENSE(NO PARITY),
1787                                                  ;&SYNC CHARACTER (26)
1788  005660  112777  000025  174044          MOVB    #25,@TXDBUF     ;LOAD THE CHAR
1789  005666  012777  005764  174042          MOV     #2$,@DUKIV      ;SET UP TRAPCATCHER
1790  005674  016777  011112  174036          MOV     DUPRT,@DURIS    ;
1791  005702  106427  000000                  MTPS    #0      ;ALLOW INTERRUPTS
```

```
1792  005706  016703  174004                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1793  005712  012700  000025                  MOV     #25,R0          ;EXPECTED
1794  005716  012777  014020  174002          MOV     #SYSTST!SEND,@TXCSR     ;OK  NOW LOAD SEND &
1795                                                                    ;MAINT.  MODE
1796  005724  052777  000120  173760          BIS     #SYNSCH!RINTEN,@RXCSR   ;SET SEARCH SYNC &
1797                                                                    ;RECEIVER INTERRUPT
1798                                                                    ;ENABLE & WAIT FOR INTERRUPT
1799  005732  005067  173552                  CLR     $TMP5
1800  005736  005002                  3$:     CLR     R2
1801  005740  005202                          INC     R2              ;WAIT FOR INTERRUPT
1802  005742  001376                          BNE     .-2
1803  005744  005267  173540                  INC     $TMP5
1804  005750  022767  000003  173532          CMP     #3,$TMP5
1805  005756  002367                          BGE     3$
1806  005760  104004                          ERROR   4                       ;INTERRUPT DID NOT OCCUR
1807  005762  000422                          BR      4$
1808
1809                                  ;THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
1810  005764  106427  000340          2$:     MTPS    #340    ;PREVENT INTERRUPTS
1811  005770  017704  173716                  MOV     @RXCSR,R4       ;SAVE
1812  005774  017701  173716                  MOV     @RXDBUF,R1      ;ACTUAL
1813  006000  016777  173734  173730          MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
1814  006006  012777  000000  173724          MOV     #0,@DURIS       ;
1815  006014  012716  006060                  MOV     #5$,(SP)        ;SET UP RETURN
1816  006020  042777  000100  173664          BIC     #RINTEN,@RXCSR  ;CLR INTERRUPT ENABLE
1817  006026  000002                          RTI
1818
1819  006030  042777  000100  173654  4$:     BIC     #RINTEN,@RXCSR  ;CLR INTERRUPT ENABLE
1820  006036  106427  000340                  MTPS    #340    ;PREVENT INTERRUPTS
1821  006042  016777  173672  173666          MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
1822  006050  012777  000000  173662          MOV     #0,@DURIS       ;
1823  006056  000415                          BR      1$
1824
1825  006060  020001                  5$:     CMP     R0,R1
1826  006062  001401                          BEQ     .+4
1827  006064  104002                          ERROR   2               ;CHARACTERS DID NOT MATCH
1828  006066  016703  173620                  MOV     RXCSR,R3        ;SETUP FOR ERROR MESSAGE
1829  006072  012700  000200                  MOV     #200,R0         ;EXPECTED
1830  006076  010401                          MOV     R4,R1           ;ACTUAL
1831  006100  042701  177577                  BIC     #177577,R1      ;SAVE ONLY RXDONE
1832  006104  020001                          CMP     R0,R1
1833  006106  001401                          BEQ     .+4
1834  006110  104001                          ERROR   1       ;FALSE INTERRUPT
1835
1836  006112                          1$:
1837                                  ;;THIS TEST VERIFYS CTP MODE (IE SYSTST MODE)
1838                                  ;;IT BASICALLY CHECKS THE EXISTANCE OF
1839                                  ;;THE FREE RUNNING OSCILLATOR
1840                                  ;;MODE:  SYNINT
1841                                  ;;LENGTH:  EIGHT
1842                                  ;;THIS TEST USES BOTH THE RECEIVER & TRANSMITTER LOGIC
1843                                  ;;
1844                                  ;;**************************************************************************
1845  006112  000004                  TST11:  SCOPE
1846  006114  052777  000400  173604          BIS     #MRESET,@TXCSR  ;MASTER RESET
1847  006122  012777  030000  173572          MOV     #SYNINT,@PARCSR ;SET THE MODE
```

```
1848  006130  052777  000400  173570          BIS     #MRESET,@TXCSR  ;MASTER RESET
1849
1850                                  ;SET MAINTENANCE MODE & SEND
1851                                          ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1852  006136  012777  014020  173562          MOV     #SYSTST!SEND,@TXCSR
1853
1854                                  ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1855  006144  012777  036026  173550          MOV     #SYNINT!EIGHT!NOPAR!26,@PARCSR
1856  006152  052777  000420  173532          BIS     #SYNSCH!STPSYN,@RXCSR   ;SET SEARCH SYNC &
1857                                          ;STRIP SYNC SO THAT RXDONE ASSERTS
1858                                          ;WHEN CHAR "25" ARRIVES AND NOT BEFORE...
1859                                          ; ,,,THEREFORE,SET STRIP SYNC
1860                                  ;.............WAIT FOR SYNSCH TO BE
1861                                  ;CLOCKED IN BY SYSTST CLK
1862  006160  005067  173324          CLR     $TMP5
1863  006164  005002                  CLR     R2
1864  006166  005202                  INC     R2      ;WAIT
1865  006170  001376                  BNE     .-2
1866  006172  005267  173312          INC     $TMP5
1867  006176  022767  000003  173304  CMP     #3,$TMP5
1868  006204  002367                  BGE     .-20    ;GO BACK TO CLR R2 AND WAIT SOME MORE
1869  006206  012777  006422  173522  MOV     #2$,@DURIV      ;SET UP TRAPCATCHER
1870  006214  016777  010572  173516  MOV     DUPRT,@DURIS    ;
1871  006222  012777  006514  173512  MOV     #3$,@DUTIV      ;
1872  006230  016777  010556  173506  MOV     DUPRT,@DUTIS    ;
1873  006236  106427  000000          MTPS    #0      ;ALLOW INTERRUPTS
1874  006242  016703  173450          MOV     RXDBUF,R3       ;SET UP FOR ERROR MSG
1875  006246  012700  000025          MOV     #25,R0  ;EXPECTED CHAR
1876  006252  012767  000002  172644  MOV     #2,COUNT        ;# OF SYNC CHARS TO GET INTO
1877                                  ;SYNCRONIZATION
1878  006260  105767  172662          TSTB    SYNCNO  ;TEST TO SEE HOW MANY SYNC CHARS NEEDED
1879  006264  100402                  BMI     9$
1880  006266  005367  172632          DEC     COUNT   ;MAKE IT ONE LESS
1881  006272  052777  000100  173412  9$:     BIS     #RINTEN,@RXCSR  ;SET INTERRUPT ENABLES
1882  006300  052777  000100  173420          BIS     #TXINTE,@TXCSR  ;
1883  006306  000167  000012          JMP     8$      ;THE FIRST XMIT INTERRUPT SHOULD COME
1884                                  ;FROM TXDONE = 1 AFTER A MASTER RESET
1885  006312  112777  000026  173412  1$:     MOVB    #26,@TXDBUF     ;LOAD SYNC CHAR
1886  006320  005067  173164          CLR     $TMP5
1887  006324  005002                  8$:     CLR     R2      ;WAIT FOR INTERRUPT
1888  006326  005202                  INC     R2
1889  006330  001376                  BNE     .-2     ;
1890  006332  005267  173152          INC     $TMP5
1891  006336  022767  000003  173144  CMP     #3,$TMP5
1892  006344  002367                  BGE     8$
1893  006346  106427  000340          MTPS    #340    ;PREVENT INTERRUPTS
1894  006352  042777  000100  173346  BIC     #TXINTE,@TXCSR  ;CLR INTR ENABLES
1895  006360  042777  000100  173324  BIC     #RINTEN,@RXCSR  ;
1896  006366  016777  173346  173342  MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
1897  006374  012777  000000  173336  MOV     #0,@DURIS       ;
1898  006402  016777  173336  173332  MOV     DUTIS,@DUTIV    ;
1899  006410  012777  000000  173326  MOV     #0,@DUTIS       ;
1900  006416  104004                  ERROR   4       ;TXDONE INTERRUPT FAILED TO OCCUR
1901  006420  000540                  BR      7$      ;GET OUT OF THE TEST
1902
1903                                  ;THE FOLLOWING IS THE RECEIVER INTERRUPT SVC ROUTINE
```

```
 1904  006422  106427  000340          2$:     MTPS    #340    ;PREVENT INTERRUPTS
 1905  006426  017704  173260                  MOV     @RXCSR,R4       ;SAVE
 1906  006432  017701  173260                  MOV     @RXDBUF,R1      ;ACTUAL
 1907  006436  016777  173276  173272          MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
 1908  006444  012777  000000  173266          MOV     #0,@DURIS       ;
 1909  006452  016777  173266  173262          MOV     DUTIS,@DUTIV    ;
 1910  006460  012777  000000  173256          MOV     #0,@DUTIS       ;
 1911  006466  012716  006646                  MOV     #4$,(SP)        ;SET UP RETURN LOCATION
 1912  006472  042777  000100  173212          BIC     #RINTEN,@RXCSR  ;CLR INTERRUPT ENABLES
 1913  006500  042777  000100  173220          BIC     #TXINTE,@TXCSR  ;
 1914  006506  016705  172412                  MOV     COUNT,R5        ;SAVE COUNT
 1915  006512  000002                          RTI
 1916                                  ;END OF RECEIVER INTERRUPT SVC ROUTINE
 1917                                  ;.... THE FOLLOWING IS THE XMITTER INTERRUPT SVC ROUTINE
 1918  006514  005367  172404          3$:     DEC     COUNT
 1919  006520  100403                          BMI     5$
 1920  006522  012716  006312                  MOV     #1$,(SP)        ;SET UP RETURN LOCATION
 1921                                                          ;(LOAD SYNC CHARACTER AGAIN)
 1922  006526  000002                          RTI
 1923  006530  012716  006536          5$:     MOV     #6$,(SP)        ;SET UP RETURN LOCATION
 1924  006534  000002                          RTI
 1925                                  ;END OF XMITTER INTERRUPT SVC ROUTINE
 1926  006536  112777  000025  173166  6$:     MOVB    #25,@TXDBUF     ;LOAD CHARACTER
 1927  006544  042777  000100  173154          BIC     #TXINTE,@TXCSR  ;CLR INTR ENABLE
 1928  006552  005067  172732                  CLR     $TMP5
 1929  006556  005002                  10$:    CLR     R2      ;WAIT FOR INTERRUPT(RECEIVER)
 1930  006560  005202                          INC     R2
 1931  006562  001376                          BNE     .-2     ;
 1932  006564  005267  172720                  INC     $TMP5
 1933  006570  022767  000003  172712          CMP     #3,$TMP5
 1934  006576  002367                          BGE     10$
 1935  006600  106427  000340                  MTPS    #340    ;PREVENT INTERRUPTS
 1936  006604  042777  000100  173100          BIC     #RINTEN,@RXCSR  ;CLR INTR ENABLE
 1937  006612  016777  173122  173116          MOV     DURIS,@DURIV    ;RESTORE TRAPCATCHER
 1938  006620  012777  000000  173112          MOV     #0,@DURIS       ;
 1939  006626  016777  173112  173106          MOV     DUTIS,@DUTIV    ;
 1940  006634  012777  000000  173102          MOV     #0,@DUTIS       ;
 1941  006642  104004                          ERROR   4       ;RECEIVER INTR FAILED TO OCCUR
 1942  006644  000426                          BR      7$      ;GET OUT OF TEST
 1943  006646  020001                  4$:     CMP     R0,R1
 1944  006650  001401                          BEQ     .+4
 1945  006652  104002                          ERROR   2       ;CHARACTERS DID NOT MATCH
 1946  006654  016703  173032                  MOV     RXCSR,R3        ;SET UP FOR ERROR MSG
 1947  006660  012700  000200                  MOV     #200,R0 ;EXPECTED RXDONE
 1948  006664  010401                          MOV     R4,R1   ;ACTUAL
 1949  006666  042701  177577                  BIC     #177577,R1      ;SAVE ONLY RXDONE
 1950  006672  020001                          CMP     R0,R1
 1951  006674  001401                          BEQ     .+4
 1952  006676  104001                          ERROR   1       ;FALSE INTERRUPT
 1953  006700  020527  177777                  CMP     R5,#-1  ;WAS COUNT =-1   WHEN RECEIVER
 1954                                                          ;INTERRUPTED ?
 1955  006704  001401                          BEQ     .+4
 1956  006706  104004                          ERROR   4       ;IF R5 IS GREATER THAN -1.....
 1957                                          ;THEN EITHER THE # OF SYNC STRAP IS WRONG
 1958                                          ;OR RXDONE IS OCCURING TOO SOON
 1959  006710  026727  172210  177777          CMP     COUNT,#-1
```

```
1960  006716  001401                          BEQ     .+4
1961  006720  104004                          ERROR   4       ;IF THIS TEST FAILS,BUT THE ABOVE TEST
1962                                                        ;DOESN'T.....IT MAY BE THAT CLEARING
1963                                                        ;TXINTE IN THE RECEIVER SVC ROUTINE
1964                                                        ;IS NOT STOPPING TXDONE INTERRUPTS
1965  006722  106427  000340          7$:     MTPS    #340    ;INHIBIT INTERRUPTS
1966
1967
1968                                        ;;THIS TEST VERIFYS MATCH DETECT & DATA RDY
1969                                        ;;FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
1970                                        ;;BY OBSERVING RECACT BIT
1971                                        ;;IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
1972                                        ;;*:    DEPENDENT ON MONITOR .............
1973                                        ;;IF ONE SYNC STRAP IS SELECTED ,IT WILL
1974                                        ;;ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
1975                                        ;;ASSERT
1976                                        ;;MODE: SYNC INTERNAL
1977                                        ;;LENGTH: FIVE
1978                                        ;;SYNC CHARACTER FOR MATCH: B/C
1979                                        ;;THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
1980                                        ;;
1981                                    ;;**********************************************************************
1982  006726  000004                    TST12:  SCOPE
1983  006730  052777  000400  172770            BIS     #MRESET,@TXCSR  ;MASTER RESET
1984  006736  016703  172754                   MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1985                                    ;SET SYNC INTERNAL,FIVE,NO PARITY,0 SYNC REGISTER
1986  006742  012704  030000                   MOV     #SYNINT!FIVE!NOPAR,R4   ;CREATE PARAMETERS
1987  006746  012777  004020  172752    6$:     MOV     #MINT!SEND,@TXCSR       ;SET SEND & MAINT INTER
1988  006754  010477  172742                   MOV     R4,@PARCSR      ;LOAD CSR
1989  006760  052777  000020  172724           BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1990                                    ;POKE CLK TO GET INTO SYNCRONIZATION
1991                                    ;BOTH THE LOGIC & RECEIVER
1992  006766  052777  020000  172732           BIS     #CLK,@TXCSR     ;POKE CLK UP
1993  006774  042777  020000  172724           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1994  007002  110477  172724                   MOVB    R4,@TXDBUF      ;LOAD DATA CHARACTER
1995                                    ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCRONIZATION
1996  007006  052777  020000  172712           BIS     #CLK,@TXCSR     ;POKE CLK UP
1997  007014  042777  020000  172704           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1998  007022  032777  004000  172662           BIT     #RECACT,@RXCSR  ;RECACT ?
1999  007030  001401                           BEQ     .+4
2000  007032  104004                           ERROR   4       ;RECACT SHOULD NOT BE SET
2001  007034  000404                           BR      4$
2002  007036  010477  172660          5$:     MOV     R4,@PARCSR      ;LOAD PARCSR WITH PARAMETERS
2003  007042  110477  172664                   MOVB    R4,@TXDBUF      ;LOAD SYNC CHAR
2004  007046  012767  000002  172050    4$:     MOV     #2,COUNT        ;# OF SYNC CHARS
2005  007054  005777  172646          2$:     TST     @TXCSR  ;DNA ?
2006  007060  100001                           BPL     .+4     ;BR IF NOT SET
2007  007062  104004                           ERROR   4       ;DNA SHOULD NOT BE SET OR....
2008                                                        ;IT SHOULD BE CLEARED FROM PREVIOUS READ
2009  007064  012767  000005  172030           MOV     #5,SHIFT        ;# OF SHIFTS
2010  007072                            1$:
2011  007072  052777  020000  172626           BIS     #CLK,@TXCSR     ;POKE CLK UP
2012  007100  042777  020000  172620           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2013  007106  005367  172010                   DEC     SHIFT   ;# OF SHIFTS
2014  007112  001367                           BNE     1$
2015  007114  005367  172004                   DEC     COUNT   ;# OF SYNC CHARS
```

```
2016  007120  001403                          BEQ     3$
2017                                  ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
2018  007122  105767  172020                  TSTB    SYNCNO
2019  007126  100752                          BMI     2$      ;TWO SYNC CHARACTERS..
2020  007130  032777  004000  172554  3$:     BIT     #RECACT,@RXCSR  ;RECACT ?
2021  007136  001001                          BNE     .+4
2022  007140  104004                          ERROR   4               ;RECACT FAILED TO SET,POSSIBLE
2023                                                                  ;THAT THE RECEIVER FAILED TO MATCH
2024                                                                  ;THE SYNC CHARACTER
2025  007142  017701  172550                  MOV     @RXDBUF,R1      ;SAVE ACTUAL
2026  007146  010400                          MOV     R4,R0           ;SAVE EXPECTED
2027  007150  042700  177400                  BIC     #177400,R0      ;CLR UPPER BYTE
2028  007154  020001                          CMP     R0,R1   ;DO THEY COMPARE ?
2029  007156  001401                          BEQ     .+4
2030  007160  104002                          ERROR   2       ;IF RECACT FAILED ALONG WITH THIS
2031                                                                  ;...IT PROBABLY IS A TRANSMITTER ERROR
2032                                                                  ;HOWEVER,...IF ONLY THIS FAILED IT
2033                                                                  ;PROBABLY IS A RECEIVER ERROR
2034  007162  104405                          SCOP1
2035                                  ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
2036                                  ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
2037                                  ;TXDBUF
2038  007164  052777  020000  172534          BIS     #CLK,@TXCSR     ;POKE CLK UP
2039  007172  005777  172530                  TST     @TXCSR  ;DNA?
2040  007176  100401                          BMI     .+4
2041  007200  104004                          ERROR   4       ;DNA DID NOT ASSERT
2042                                          ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
2043  007202  052777  000400  172516          BIS     #MRESET,@TXCSR  ;MASTER RESET
2044  007210  032777  000020  172474          BIT     #SYNSCH,@RXCSR  ;SYNC SEARCH = 0 ?
2045  007216  001401                          BEQ     .+4
2046  007220  104004                          ERROR   4       ;SYNC SEARCH SHOULD BE NOT SET
2047  007222  005204                          INC     R4
2048  007224  122704  000040                  CMPB    #40,R4  ;IS THIS THE LAST CHARACTER ?
2049  007230  001246                          BNE     6$      ;NO
2050
2051                                          ;;THIS TEST VERIFYS MATCH DETECT & DATA RDY
2052                                          ;;FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
2053                                          ;;BY OBSERVING RECACT BIT
2054                                          ;;IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
2055                                          ;;*:    DEPENDENT ON MONITOR .....
2056                                          ;;IF ONE SYNC STRAP IS SELECTED ,IT WILL
2057                                          ;;ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
2058                                          ;;ASSERT
2059                                          ;;MODE: SYNC INTERNAL
2060                                          ;;LENGTH: SIX
2061                                          ;;SYNC CHARACTER FOR MATCH: B/C
2062                                          ;;THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
2063                                          ;;
2064                                  ;;**********************************************************************
2065  007232  000004                  TST13:  SCOPE
2066  007234  052777  000400  172464          BIS     #MRESET,@TXCSR  ;MASTER RESET
2067  007242  016703  172450                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2068                                  ;SET SYNC INTERNAL,SIX,NO PARITY,0 SYNC REGISTER
2069  007246  012704  032000                  MOV     #SYNINT!SIX!NOPAR,R4    ;CREATE PARAMETERS
2070  007252  012777  004020  172446  6$:     MOV     #MINT!SEND,@TXCSR       ;SET SEND & MAINT INTER
2071  007260  010477  172436                  MOV     R4,@PARCSR      ;LOAD CSR
```

```
2072  007264  052777  000020  172420          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2073                                  ;POKE CLK TO GET INTO SYNCRONIZATION
2074                                  ;BOTH THE LOGIC & RECEIVER
2075  007272  052777  020000  172426          BIS     #CLK,@TXCSR     ;POKE CLK UP
2076  007300  042777  020000  172420          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2077  007306  110477  172420                  MOVB    R4,@TXDBUF      ;LOAD DATA CHARACTER
2078                                  ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCRONIZATION
2079  007312  052777  020000  172406          BIS     #CLK,@TXCSR     ;POKE CLK UP
2080  007320  042777  020000  172400          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2081  007326  032777  004000  172356          BIT     #RECACT,@RXCSR  ;RECACT ?
2082  007334  001401                          BEQ     .+4
2083  007336  104004                          ERROR   4       ;RECACT SHOULD NOT BE SET
2084  007340  000404                          BR      4$
2085  007342  010477  172354          5$:     MOV     R4,@PARCSR      ;LOAD PARCSR WITH PARAMETERS
2086  007346  110477  172360                  MOVB    R4,@TXDBUF      ;LOAD SYNC CHAR
2087  007352  012767  000002  171544  4$:     MOV     #2,COUNT        ;# OF SYNC CHARS
2088  007360  005777  172342          2$:     TST     @TXCSR  ;DNA ?
2089  007364  100001                          BPL     .+4     ;BR IF NOT SET
2090  007366  104004                          ERROR   4       ;DNA SHOULD NOT BE SET OR...
2091                                          ;IT SHOULD BE CLEARED FROM PREVIOUS READ
2092  007370  012767  000006  171524          MOV     #6,SHIFT        ;# OF SHIFTS
2093  007376                          1$:
2094  007376  052777  020000  172322          BIS     #CLK,@TXCSR     ;POKE CLK UP
2095  007404  042777  020000  172314          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2096  007412  005367  171504                  DEC     SHIFT   ;# OF SHIFTS
2097  007416  001367                          BNE     1$
2098  007420  005367  171500                  DEC     COUNT   ;# OF SYNC CHARS
2099  007424  001403                          BEQ     3$
2100                                  ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
2101  007426  105767  171514                  TSTB    SYNCNO
2102  007432  100752                          BMI     2$      ;TWO SYNC CHARACTERS..
2103  007434  032777  004000  172250  3$:     BIT     #RECACT,@RXCSR  ;RECACT ?
2104  007442  001001                          BNE     .+4
2105  007444  104004                          ERROR   4               ;RECACT FAILED TO SET,POSSIBLE
2106                                                  ;THAT THE RECEIVER FAILED TO MATCH
2107                                                  ;THE SYNC CHARACTER
2108  007446  017701  172244                  MOV     @RXDBUF,R1      ;SAVE ACTUAL
2109  007452  010400                          MOV     R4,R0           ;SAVE EXPECTED
2110  007454  042700  177400                  BIC     #177400,R0      ;CLR UPPER BYTE
2111  007460  020001                          CMP     R0,R1   ;DO THEY COMPARE ?
2112  007462  001401                          BEQ     .+4
2113  007464  104002                          ERROR   2       ;IF RECACT FAILED ALONG WITH THIS
2114                                                  ;...IT PROBABLY IS A TRANSMITTER ERROR
2115                                                  ;HOWEVER,...IF ONLY THIS FAILED IT
2116                                                  ;PROBABLY IS A RECEIVER ERROR
2117  007466  104405                          SCOP1
2118                                  ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
2119                                  ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
2120                                  ;TXDBUF
2121  007470  052777  020000  172230          BIS     #CLK,@TXCSR     ;POKE CLK UP
2122  007476  005777  172224                  TST     @TXCSR  ;DNA?
2123  007502  100401                          BMI     .+4
2124  007504  104004                          ERROR   4       ;DNA DID NOT ASSERT
2125                                          ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
2126  007506  052777  000400  172212          BIS     #MRESET,@TXCSR  ;MASTER RESET
2127  007514  032777  000020  172170          BIT     #SYNSCH,@RXCSR  ;SYNC SEARCH = 0 ?
```

```
2128  007522  001401                          BEQ     .+4
2129  007524  104004                          ERROR   4       ;SYNC SEARCH SHOULD BE NOT SET
2130  007526  005204                          INC     R4
2131  007530  122704  000100                  CMPB    #100,R4 ;IS THIS THE LAST CHARACTER ?
2132  007534  001246                          BNE     6$      ;NO
2133
2134                                          ;;THIS TEST VERIFYS MATCH DETECT & DATA RDY
2135                                          ;;FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
2136                                          ;;BY OBSERVING RECACT BIT
2137                                          ;;IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
2138                                          ;;*:    DEPENDENT ON MONITOR .............
2139                                          ;;IF ONE SYNC STRAP IS SELECTED ,IT WILL
2140                                          ;;ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
2141                                          ;;ASSERT
2142                                          ;;MODE: SYNC INTERNAL
2143                                          ;;LENGTH: SEVEN
2144                                          ;;SYNC CHARACTER FOR MATCH: B/C
2145                                          ;;THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
2146                                          ;;
2147                                  ;;**************************************************************************
2148  007536  000004                  TST14:  SCOPE
2149  007540  052777  000400  172160          BIS     #MRESET,@TXCSR  ;MASTER RESET
2150  007546  016703  172144                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2151                                  ;SET SYNC INTERNAL,SEVEN,NO PARITY,0 SYNC REGISTER
2152  007552  012704  034000                  MOV     #SYNINT!SEVEN!NOPAR,R4  ;CREATE PARAMETERS
2153  007556  012777  004020  172142  6$:     MOV     #MINT!SEND,@TXCSR       ;SET SEND & MAINT INTER
2154  007564  010477  172132                  MOV     R4,@PARCSR      ;LOAD CSR
2155  007570  052777  000020  172114          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2156                                  ;POKE CLK TO GET INTO SYNCRONIZATION
2157                                  ;BOTH THE LOGIC & RECEIVER
2158  007576  052777  020000  172122          BIS     #CLK,@TXCSR     ;POKE CLK UP
2159  007604  042777  020000  172114          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2160  007612  110477  172114                  MOVB    R4,@TXDBUF      ;LOAD DATA CHARACTER
2161                                  ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCRONIZATION
2162  007616  052777  020000  172102          BIS     #CLK,@TXCSR     ;POKE CLK UP
2163  007624  042777  020000  172074          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2164  007632  032777  004000  172052          BIT     #RECACT,@RXCSR  ;RECACT ?
2165  007640  001401                          BEQ     .+4
2166  007642  104004                          ERROR   4       ;RECACT SHOULD NOT BE SET
2167  007644  000404                          BR      4$
2168  007646  010477  172050          5$:     MOV     R4,@PARCSR      ;LOAD PARCSR WITH PARAMETERS
2169  007652  110477  172054                  MOVB    R4,@TXDBUF      ;LOAD SYNC CHAR
2170  007656  012767  000002  171240  4$:     MOV     #2,COUNT        ;# OF SYNC CHARS
2171  007664  005777  172036          2$:     TST     @TXCSR  ;DNA ?
2172  007670  100001                          BPL     .+4     ;BR IF NOT SET
2173  007672  104004                          ERROR   4       ;DNA SHOULD NOT BE SET OR...
2174                                          ;IT SHOULD BE CLEARED FROM PREVIOUS READ
2175  007674  012767  000007  171220          MOV     #7,SHIFT        ;# OF SHIFTS
2176  007702                          1$:
2177  007702  052777  020000  172016          BIS     #CLK,@TXCSR     ;POKE CLK UP
2178  007710  042777  020000  172010          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2179  007716  005367  171200                  DEC     SHIFT   ;# OF SHIFTS
2180  007722  001367                          BNE     1$
2181  007724  005367  171174                  DEC     COUNT   ;# OF SYNC CHARS
2182  007730  001403                          BEQ     3$
2183                                  ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
```

```
2184  007732  105767  171210                    TSTB    SYNCNO
2185  007736  100752                            BMI     2$      ;TWO SYNC CHARACTERS..
2186  007740  032777  004000  171744  3$:       BIT     #RECACT,@RXCSR  ;RECACT ?
2187  007746  001001                            BNE     .+4
2188  007750  104004                            ERROR   4               ;RECACT FAILED TO SET,POSSIBLE
2189                                                                    ;THAT THE RECEIVER FAILED TO MATCH
2190                                                                    ;THE SYNC CHARACTER
2191  007752  017701  171740                    MOV     @RXDBUF,R1      ;SAVE ACTUAL
2192  007756  010400                            MOV     R4,R0           ;SAVE EXPECTED
2193  007760  042700  177400                    BIC     #177400,R0      ;CLR UPPER BYTE
2194  007764  020001                            CMP     R0,R1   ;DO THEY COMPARE ?
2195  007766  001401                            BEQ     .+4
2196  007770  104002                            ERROR   2       ;IF RECACT FAILED ALONG WITH THIS
2197                                                                    ;...IT PROBABLY IS A TRANSMITTER ERROR
2198                                                                    ;HOWEVER,...IF ONLY THIS FAILED IT
2199                                                                    ;PROBABLY IS A RECEIVER ERROR
2200  007772  104405                            SCOP1
2201                                    ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
2202                                    ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
2203                                    ;TXDBUF
2204  007774  052777  020000  171724            BIS     #CLK,@TXCSR     ;POKE CLK UP
2205  010002  005777  171720                    TST     @TXCSR  ;DNA?
2206  010006  100401                            BMI     .+4
2207  010010  104004                            ERROR   4       ;DNA DID NOT ASSERT
2208                                            ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
2209  010012  052777  000400  171706            BIS     #MRESET,@TXCSR  ;MASTER RESET
2210  010020  032777  000020  171664            BIT     #SYNSCH,@RXCSR  ;SYNC SEARCH = 0 ?
2211  010026  001401                            BEQ     .+4
2212  010030  104004                            ERROR   4       ;SYNC SEARCH SHOULD BE NOT SET
2213  010032  005204                            INC     R4
2214  010034  122704  000200                    CMPB    #200,R4 ;IS THIS THE LAST CHARACTER ?
2215  010040  001246                            BNE     6$      ;NO
2216
2217                                            ;;THIS TEST VERIFYS MATCH DETECT & DATA RDY
2218                                            ;;FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
2219                                            ;;BY OBSERVING RECACT BIT
2220                                            ;;IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
2221                                            ;;*:    DEPENDENT ON MONITOR .............
2222                                            ;;IF ONE SYNC STRAP IS SELECTED ,IT WILL
2223                                            ;;ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
2224                                            ;;ASSERT
2225                                            ;;MODE: SYNC INTERNAL
2226                                            ;;LENGTH: EIGHT
2227                                            ;;SYNC CHARACTER FOR MATCH: B/C
2228                                            ;;THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
2229                                            ;;
2230                                    ;;**************************************************************
2231  010042  000004                    TST15:  SCOPE
2232  010044  052777  000400  171654            BIS     #MRESET,@TXCSR  ;MASTER RESET
2233  010052  016703  171640                    MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2234                                    ;SET SYNC INTERNAL,EIGHT,NO PARITY,0 SYNC REGISTER
2235  010056  012704  036000                    MOV     #SYNINT!EIGHT!NOPAR,R4  ;CREATE PARAMETERS
2236  010062  012777  004020  171636  6$:       MOV     #MINT!SEND,@TXCSR       ;SET SEND & MAINT INTER
2237  010070  010477  171626                    MOV     R4,@PARCSR      ;LOAD CSR
2238  010074  052777  000020  171610            BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2239                                    ;POKE CLK TO GET INTO SYNCRONIZATION
```

```
2240                                    ;BOTH THE LOGIC & RECEIVER
2241  010102  052777  020000  171616            BIS     #CLK,@TXCSR     ;POKE CLK UP
2242  010110  042777  020000  171610            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2243  010116  110477  171610                    MOVB    R4,@TXDBUF      ;LOAD DATA CHARACTER
2244                                    ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCRONIZATION
2245  010122  052777  020000  171576            BIS     #CLK,@TXCSR     ;POKE CLK UP
2246  010130  042777  020000  171570            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2247  010136  032777  004000  171546            BIT     #RECACT,@RXCSR  ;RECACT ?
2248  010144  001401                            BEQ     .+4
2249  010146  104004                            ERROR   4               ;RECACT SHOULD NOT BE SET
2250  010150  000404                            BR      4$
2251  010152  010477  171544           5$:      MOV     R4,@PARCSR      ;LOAD PARCSR WITH PARAMETERS
2252  010156  110477  171550                    MOVB    R4,@TXDBUF      ;LOAD SYNC CHAR
2253  010162  012767  000002  170734   4$:      MOV     #2,COUNT        ;# OF SYNC CHARS
2254  010170  005777  171532           2$:      TST     @TXCSR  ;DNA ?
2255  010174  100001                            BPL     .+4     ;BR IF NOT SET
2256  010176  104004                            ERROR   4       ;DNA SHOULD NOT BE SET OR...
2257                                            ;IT SHOULD BE CLEARED FROM PREVIOUS READ
2258  010200  012767  000010  170714            MOV     #8.,SHIFT       ;# OF SHIFTS
2259  010206                           1$:
2260  010206  052777  020000  171512            BIS     #CLK,@TXCSR     ;POKE CLK UP
2261  010214  042777  020000  171504            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2262  010222  005367  170674                    DEC     SHIFT   ;# OF SHIFTS
2263  010226  001367                            BNE     1$
2264  010230  005367  170670                    DEC     COUNT   ;# OF SYNC CHARS
2265  010234  001403                            BEQ     3$
2266                                    ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
2267  010236  105767  170704                    TSTB    SYNCNO
2268  010242  100752                            BMI     2$      ;TWO SYNC CHARACTERS..
2269  010244  032777  004000  171440   3$:      BIT     #RECACT,@RXCSR  ;RECACT ?
2270  010252  001001                            BNE     .+4
2271  010254  104004                            ERROR   4               ;RECACT FAILED TO SET,POSSIBLE
2272                                                    ;THAT THE RECEIVER FAILED TO MATCH
2273                                                    ;THE SYNC CHARACTER
2274  010256  017701  171434                    MOV     @RXDBUF,R1      ;SAVE ACTUAL
2275  010262  010400                            MOV     R4,R0           ;SAVE EXPECTED
2276  010264  042700  177400                    BIC     #177400,R0      ;CLR UPPER BYTE
2277  010270  020001                            CMP     R0,R1   ;DO THEY COMPARE ?
2278  010272  001401                            BEQ     .+4
2279  010274  104002                            ERROR   2       ;IF RECACT FAILED ALONG WITH THIS
2280                                                    ;...IT PROBABLY IS A TRANSMITTER ERROR
2281                                                    ;HOWEVER,...IF ONLY THIS FAILED IT
2282                                                    ;PROBABLY IS A RECEIVER ERROR
2283  010276  104405                            SCOP1
2284                                    ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
2285                                    ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
2286                                    ;TXDBUF
2287  010300  052777  020000  171420            BIS     #CLK,@TXCSR     ;POKE CLK UP
2288  010306  005777  171414                    TST     @TXCSR  ;DNA?
2289  010312  100401                            BMI     .+4
2290  010314  104004                            ERROR   4       ;DNA DID NOT ASSERT
2291                                            ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
2292  010316  052777  000400  171402            BIS     #MRESET,@TXCSR  ;MASTER RESET
2293  010324  032777  000020  171360            BIT     #SYNSCH,@RXCSR  ;SYNC SEARCH =.0 ?
2294  010332  001401                            BEQ     .+4
2295  010334  104004                            ERROR   4       ;SYNC SEARCH SHOULD BE NOT SET
```

```
2296  010336  005204                          INC     R4
2297  010340  122704  000000                  CMPB    #0,R4   ;IS THIS THE LAST CHARACTER ?
2298  010344  001246                          BNE     6$      ;NO
2299
2300                                          ;;THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
2301                                          ;;BOTH THE TRANSMITTER AND RECEIVER LOGIC
2302                                          ;;MODE: SYNC EXTERNAL (SYNEXT)
2303                                          ;;LENGTH: EIGHT PLUS PARITY
2304                                          ;;PARITY: EVEPAR
2305                                          ;;MAINT. MODE: MINT
2306                                          ;;
2307                                  ;;**************************************************************************
2308  010346  000004                  TST16:  SCOPE
2309  010350  052777  000400  171350          BIS     #MRESET,@TXCSR  ;MASTER RESET
2310  010356  012777  020000  171336          MOV     #SYNEXT,@PARCSR ;SET THE MODE
2311  010364  052777  000400  171334          BIS     #MRESET,@TXCSR  ;MASTER RESET
2312
2313                                  ;SET MAINTENANCE MODE & SEND
2314                                          ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2315  010372  012777  004020  171326          MOV     #MINT!SEND,@TXCSR
2316
2317                                  ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2318  010400  012777  027426  171314          MOV     #SYNEXT!EIGHT!EVEPAR!26,@PARCSR
2319  010406  016703  171304                  MOV     RXDBUF,R3       ;SETUP FOR ERROR MSG
2320  010412  005004                          CLR     R4      ;FOR DATA CHAR CREATION
2321  010414  110477  171312                  MOVB    R4,@TXDBUF      ;LOAD CHARACTER
2322  010420  052777  000020  171264          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2323                                  ;GET INTO SYNCRONIZATION
2324  010426  052777  020000  171272          BIS     #CLK,@TXCSR     ;POKE CLK UP
2325  010434  042777  020000  171264          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2326  010442  012767  000011  170452  1$:     MOV     #9.,SHIFT       ;# OF SHIFTS
2327  010450  010400                          MOV     R4,R0   ;EXPECTED
2328  010452                          2$:
2329  010452  052777  020000  171246          BIS     #CLK,@TXCSR     ;POKE CLK UP
2330  010460  042777  020000  171240          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2331  010466  005367  170430                  DEC     SHIFT   ;# OF SHIFTS
2332  010472  022767  000003  170422          CMP     #3,SHIFT        ;TIME TO LOAD NEXT CHAR ?
2333  010500  001003                          BNE     3$      ;NO ?
2334  010502  005204                          INC     R4      ;GENERATE NEXT CHAR
2335  010504  110477  171222                  MOVB    R4,@TXDBUF      ;LOAD NEXT CHARACTER
2336  010510  005767  170406          3$:     TST     SHIFT   ;IS IT 0 ?
2337  010514  001356                          BNE     2$
2338  010516  105777  171170                  TSTB    @RXCSR  ;RXDONE = 1 ?
2339  010522  100401                          BMI     .+4
2340  010524  104004                          ERROR   4       ;RXDONE SHOULD BE SET
2341  010526  017701  171164                  MOV     @RXDBUF,R1      ;ACTUAL
2342  010532  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
2343  010534  001401                          BEQ     .+4
2344  010536  104002                          ERROR   2       ;CHARACTERS SHOULD COMPARE
2345  010540  105704                          TSTB    R4      ;LAST CHARACTER ?
2346  010542  001337                          BNE     1$      ;NO
2347  010544                          4$:
2348                                          ;;THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
2349                                          ;;BOTH THE TRANSMITTER AND RECEIVER LOGIC
2350                                          ;;MODE: SYNC EXTERNAL (SYNEXT)
2351                                          ;;LENGTH: EIGHT PLUS PARITY
```

```
                                                ;;PARITY: ODDPAR
                                                ;;MAINT. MODE: MINT
                                                ;;
                                        ;;*****************************************************************
2356  010544  000004                   TST17:  SCOPE
2357  010546  052777  000400  171152           BIS     #MRESET,@TXCSR  ;MASTER RESET
2358  010554  012777  020000  171140           MOV     #SYNEXT,@PARCSR ;SET THE MODE
2359  010562  052777  000400  171136           BIS     #MRESET,@TXCSR  ;MASTER RESET
2360
2361                                    ;SET MAINTENANCE MODE & SEND
2362                                            ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2363  010570  012777  004020  171130           MOV     #MINT!SEND,@TXCSR
2364
2365                                    ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2366  010576  012777  027026  171116           MOV     #SYNEXT!EIGHT!ODDPAR!26,@PARCSR
2367  010604  016703  171106                   MOV     RXDBUF,R3       ;SETUP FOR ERROR MSG
2368  010610  005004                           CLR     R4      ;FOR DATA CHAR CREATION
2369  010612  110477  171114                   MOVB    R4,@TXDBUF      ;LOAD CHARACTER
2370  010616  052777  000020  171066           BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2371                                    ;GET INTO SYNCRONIZATION
2372  010624  052777  020000  171074           BIS     #CLK,@TXCSR     ;POKE CLK UP
2373  010632  042777  020000  171066           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2374  010640  012767  000011  170254   1$:     MOV     #9.,SHIFT       ;# OF SHIFTS
2375  010646  010400                           MOV     R4,R0   ;EXPECTED
2376  010650                           2$:
2377  010650  052777  020000  171050           BIS     #CLK,@TXCSR     ;POKE CLK UP
2378  010656  042777  020000  171042           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2379  010664  005367  170232                   DEC     SHIFT   ;# OF SHIFTS
2380  010670  022767  000003  170224           CMP     #3,SHIFT        ;TIME TO LOAD NEXT CHAR ?
2381  010676  001003                           BNE     3$      ;NO ?
2382  010700  005204                           INC     R4      ;GENERATE NEXT CHAR
2383  010702  110477  171024                   MOVB    R4,@TXDBUF      ;LOAD NEXT CHARACTER
2384  010706  005767  170210           3$:     TST     SHIFT   ;IS IT 0 ?
2385  010712  001356                           BNE     2$
2386  010714  105777  170772                   TSTB    @RXCSR  ;RXDONE = 1 ?
2387  010720  100401                           BMI     .+4
2388  010722  104004                           ERROR   4       ;RXDONE SHOULD BE SET
2389  010724  017701  170766                   MOV     @RXDBUF,R1      ;ACTUAL
2390  010730  020001                           CMP     R0,R1   ;COMPARE EXP VS ACT
2391  010732  001401                           BEQ     .+4
2392  010734  104002                           ERROR   2       ;CHARACTERS SHOULD COMPARE
2393  010736  105704                           TSTB    R4      ;LAST CHARACTER ?
2394  010740  001337                           BNE     1$      ;NO
2395  010742                           4$:
2396                                            ;;THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
2397                                            ;;BOTH THE TRANSMITTER AND RECEIVER LOGIC
2398                                            ;;MODE: SYNC EXTERNAL (SYNEXT)
2399                                            ;;LENGTH: EIGHT PLUS PARITY
2400                                            ;;PARITY: EVEPAR
2401                                            ;;MAINT. MODE: MEXT
2402                                            ;;
2403                                    ;;*****************************************************************
2404  010742  000004                   TST20:  SCOPE
2405  010744  105767  170203                   TSTB    JMRBY   ;JUMP AROUND TEST ?
2406  010750  100116                           BPL     4$      ;YES ?
2407  010752  052777  000400  170746           BIS     #MRESET,@TXCSR  ;MASTER RESET
```

```
2408  010760  012777  020000  170734          MOV     #SYNEXT,@PARCSR ;SET THE MODE
2409  010766  052777  000400  170732          BIS     #MRESET,@TXCSR  ;MASTER RESET
2410
2411                                  ;SET MAINTENANCE MODE & SEND
2412                                      ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2413  010774  012777  010020  170724          MOV     #MEXT!SEND,@TXCSR
2414
2415                                  ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2416  011002  012777  027426  170712          MOV     #SYNEXT!EIGHT!EVEPAR!26,@PARCSR
2417  011010  016703  170702                  MOV     RXDBUF,R3       ;SETUP FOR ERROR MSG
2418  011014  005004                          CLR     R4      ;FOR DATA CHAR CREATION
2419  011016  110477  170710                  MOVB    R4,@TXDBUF      ;LOAD CHARACTER
2420  011022  052777  000020  170662          BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2421                                  ;GET INTO SYNCRONIZATION
2422  011030  052777  020000  170670          BIS     #CLK,@TXCSR     ;POKE CLK UP
2423                                          ;WAIT FOR CABLE & DRIVER DELAYS
2424  011036  016702  170056                  MOV     HOLD,R2 ;WAIT THIS AMT
2425  011042  005302                          DEC     R2      ;WAIT
2426  011044  001376                          BNE     .-2
2427                                          ;EXIT...
2428  011046  042777  020000  170652          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2429                                          ;WAIT FOR CABLE & DRIVER DELAYS
2430  011054  016702  170040                  MOV     HOLD,R2 ;WAIT THIS AMT
2431  011060  005302                          DEC     R2      ;WAIT
2432  011062  001376                          BNE     .-2
2433                                          ;EXIT...
2434  011064  012767  000011  170030  1$:     MOV     #9.,SHIFT       ;# OF SHIFTS
2435  011072  010400                          MOV     R4,R0   ;EXPECTED
2436  011074                          2$:
2437  011074  052777  020000  170624          BIS     #CLK,@TXCSR     ;POKE CLK UP
2438                                          ;WAIT FOR CABLE & DRIVER DELAYS
2439  011102  016702  170012                  MOV     HOLD,R2 ;WAIT THIS AMT
2440  011106  005302                          DEC     R2      ;WAIT
2441  011110  001376                          BNE     .-2
2442                                          ;EXIT...
2443  011112  042777  020000  170606          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2444                                          ;WAIT FOR CABLE & DRIVER DELAYS
2445  011120  016702  167774                  MOV     HOLD,R2 ;WAIT THIS AMT
2446  011124  005302                          DEC     R2      ;WAIT
2447  011126  001376                          BNE     .-2
2448                                          ;EXIT...
2449  011130  005367  167766                  DEC     SHIFT   ;# OF SHIFTS
2450  011134  022767  000003  167760          CMP     #3,SHIFT        ;TIME TO LOAD NEXT CHAR ?
2451  011142  001003                          BNE     3$      ;NO ?
2452  011144  005204                          INC     R4      ;GENERATE NEXT CHAR
2453  011146  110477  170560                  MOVB    R4,@TXDBUF      ;LOAD NEXT CHARACTER
2454  011152  005767  167744          3$:     TST     SHIFT   ;IS IT 0 ?
2455  011156  001346                          BNE     2$
2456  011160  105777  170526                  TSTB    @RXCSR  ;RXDONE = 1 ?
2457  011164  100401                          BMI     .+4
2458  011166  104004                          ERROR   4       ;RXDONE SHOULD BE SET
2459  011170  017701  170522                  MOV     @RXDBUF,R1      ;ACTUAL
2460  011174  020001                          CMP     R0,R1   ;COMPARE EXP VS ACT
2461  011176  001401                          BEQ     .+4
2462  011200  104002                          ERROR   2       ;CHARACTERS SHOULD COMPARE
2463                                  ;CHECK OUT MODEM BYPASS JUMPER
```

```
2464  011202  105704                          TSTB    R4      ;LAST CHARACTER ?
2465  011204  001327                          BNE     1$      ;NO
2466  011206                          4$:
2467                                          ;;THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
2468                                          ;;BOTH THE TRANSMITTER AND RECEIVER LOGIC
2469                                          ;;MODE: SYNC EXTERNAL (SYNEXT)
2470                                          ;;LENGTH: EIGHT PLUS PARITY
2471                                          ;;PARITY: ODDPAR
2472                                          ;;MAINT. MODE: MEXT
2473                                          ;;
2474                                  ;;*****************************************************************
2475  011206  000004                  TST21:  SCOPE
2476  011210  105767  167737                 TSTB    JMRBY   ;JUMP AROUND TEST ?
2477  011214  100116                 .        BPL     4$      ;YES ?
2478  011216  052777  000400  170502         BIS     #MRESET,@TXCSR  ;MASTER RESET
2479  011224  012777  020000  170470         MOV     #SYNEXT,@PARCSR ;SET THE MODE
2480  011232  052777  000400  170466         BIS     #MRESET,@TXCSR  ;MASTER RESET
2481
2482                                  ;SET MAINTENANCE MODE & SEND
2483                                          ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2484  011240  012777  010020  170460         MOV     #MEXT!SEND,@TXCSR
2485
2486                                  ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2487  011246  012777  027026  170446         MOV     #SYNEXT!EIGHT!ODDPAR!26,@PARCSR
2488  011254  016703  170436                 MOV     RXDBUF,R3       ;SETUP FOR ERROR MSG
2489  011260  005004                         CLR     R4      ;FOR DATA CHAR CREATION
2490  011262  110477  170444                 MOVB    R4,@TXDBUF      ;LOAD CHARACTER
2491  011266  052777  000020  170416         BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2492                                  ;GET INTO SYNCRONIZATION
2493  011274  052777  020000  170424         BIS     #CLK,@TXCSR     ;POKE CLK UP
2494                                          ;WAIT FOR CABLE & DRIVER DELAYS
2495  011302  016702  167612                 MOV     HOLD,R2 ;WAIT THIS AMT
2496  011306  005302                         DEC     R2      ;WAIT
2497  011310  001376                         BNE     .-2
2498                                          ;EXIT...
2499  011312  042777  020000  170406         BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2500                                          ;WAIT FOR CABLE & DRIVER DELAYS
2501  011320  016702  167574                 MOV     HOLD,R2 ;WAIT THIS AMT
2502  011324  005302                         DEC     R2      ;WAIT
2503  011326  001376                         BNE     .-2
2504                                          ;EXIT...
2505  011330  012767  000011  167564  1$:    MOV     #9.,SHIFT       ;# OF SHIFTS
2506  011336  010400                         MOV     R4,R0   ;EXPECTED
2507  011340                          2$:
2508  011340  052777  020000  170360         BIS     #CLK,@TXCSR     ;POKE CLK UP
2509                                          ;WAIT FOR CABLE & DRIVER DELAYS
2510  011346  016702  167546                 MOV     HOLD,R2 ;WAIT THIS AMT
2511  011352  005302                         DEC     R2      ;WAIT
2512  011354  001376                         BNE     .-2
2513                                          ;EXIT...
2514  011356  042777  020000  170342         BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2515                                          ;WAIT FOR CABLE & DRIVER DELAYS
2516  011364  016702  167530                 MOV     HOLD,R2 ;WAIT THIS AMT
2517  011370  005302                         DEC     R2      ;WAIT
2518  011372  001376                         BNE     .-2
2519                                          ;EXIT...
```

```
2520  011374  005367  167522                DEC     SHIFT   ;# OF SHIFTS
2521  011400  022767  000003  167514        CMP     #3,SHIFT        ;TIME TO LOAD NEXT CHAR ?
2522  011406  001003                        BNE     3$      ;NO ?
2523  011410  005204                        INC     R4      ;GENERATE NEXT CHAR
2524  011412  110477  170314                MOVB    R4,@TXDBUF      ;LOAD NEXT CHARACTER
2525  011416  005767  167500        3$:     TST     SHIFT   ;IS IT 0 ?
2526  011422  001346                        BNE     2$
2527  011424  105777  170262                TSTB    @RXCSR  ;RXDONE = 1 ?
2528  011430  100401                        BMI     .+4
2529  011432  104004                        ERROR   4       ;RXDONE SHOULD BE SET
2530  011434  017701  170256                MOV     @RXDBUF,R1      ;ACTUAL
2531  011440  020001                        CMP     R0,R1   ;COMPARE EXP VS ACT
2532  011442  001401                        BEQ     .+4
2533  011444  104002                        ERROR   2       ;CHARACTERS SHOULD COMPARE
2534                                        ;CHECK OUT MODEM BYPASS JUMPER
2535  011446  105704                        TSTB    R4      ;LAST CHARACTER ?
2536  011450  001327                        BNE     1$      ;NO
2537  011452                        4$:
2538                                        ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2539                                        ;;RECEIVER SECTION, IT USES THE ERROR FLAGS
2540                                        ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2541                                        ;;(OVRRUN,RXERR)
2542                                        ;;MODE: ISYMOD
2543                                        ;;LENGTH: FIVE
2544                                        ;;CHAR: 12
2545                                        ;;
2546                                ;;**********************************************************************
2547  011452  000004                TST22:  SCOPE
2548  011454  052777  000400  170244        BIS     #MRESET,@TXCSR  ;MASTER RESET
2549  011462  012777  000000  170232        MOV     #ISYMOD,@PARCSR ;SET THE MODE
2550  011470  052777  000400  170230        BIS     #MRESET,@TXCSR  ;MASTER RESET
2551
2552                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2553  011476  012777  064001  170222        MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2554
2555                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2556  011504  012777  000000  170210        MOV     #ISYMOD!FIVE!NOPAR!0,@PARCSR
2557  011512  052777  000020  170172        BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2558                                        ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2559  011520  042777  020000  170200        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2560  011526  052777  020000  170172        BIS     #CLK,@TXCSR     ;POKE CLK UP
2561                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2562  011534  042777  020000  170164        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2563  011542  052777  020000  170156        BIS     #CLK,@TXCSR     ;POKE CLK UP
2564  011550  016703  170142                MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2565  011554  012700  000012                MOV     #12,R0  ;EXPECTED
2566  011560  012767  000007  167334        MOV     #7,SHIFT        ;# OF SHIFTS
2567  011566  012767  000124  167704        MOV     #124,$TMP1      ;DATA CHAR
2568  011574  004767  005352                JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2569  011600  105777  170106                TSTB    @RXCSR  ;RXDONE ?
2570  011604  100401                        BMI     .+4
2571  011606  104004                        ERROR   4       ;RXDONE SHOULD BE SET
2572  011610  017701  170102                MOV     @RXDBUF,R1      ;ACTUAL
2573  011614  020001                        CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2574  011616  001401                        BEQ     .+4
2575  011620  104002                        ERROR   2       ;RECEIVED DATA DID NOT MATCH
```

```
2576                                                    ;EXPECTED DATA - CHECK MAINT DATA
2577                                                    ;OR RECEIVER LOGIC
2578  011622  012767  000007  167272          MOV     #7,SHIFT        ;# OF SHIFTS
2579  011630  012767  000124  167642          MOV     #124,$TMP1      ;DATA CHAR
2580  011636  004767  005310                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2581                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2582  011642  012767  000007  167252          MOV     #7,SHIFT        ;# OF SHIFTS
2583  011650  012767  000124  167622          MOV     #124,$TMP1      ;DATA CHAR
2584  011656  004767  005270                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2585  011662  012700  140012                  MOV     #140000!12,R0   ;EXPECTED DATA PLUS
2586                                                                  ;RXERR & OVRRUN
2587  011666  017701  170024                  MOV     @RXDBUF,R1      ;ACTUAL
2588  011672  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
2589  011674  001401                          BEQ     .+4
2590  011676  104002                          ERROR   2               ;SPECIFICALLY LOOK AT RXERR &
2591                                                                  ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2592
2593                                                  ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2594                                                  ;;RECEIVER SECTION, IT USES THE ERROR FLAGS
2595                                                  ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2596                                                  ;;(OVRRUN,RXERR)
2597                                                  ;;MODE: ISYMOD
2598                                                  ;;LENGTH: FIVE
2599                                                  ;;CHAR: 37
2600                                                  ;;
2601                                  ;;**************************************************************
2602  011700  000004                  TST23:  SCOPE
2603  011702  052777  000400  170016          BIS     #MRESET,@TXCSR  ;MASTER RESET
2604  011710  012777  000000  170004          MOV     #ISYMOD,@PARCSR ;SET THE MODE
2605  011716  052777  000400  170002          BIS     #MRESET,@TXCSR  ;MASTER RESET
2606
2607                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2608  011724  012777  064001  167774          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2609
2610                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2611  011732  012777  000000  167762          MOV     #ISYMOD!FIVE!NOPAR!0,@PARCSR
2612  011740  052777  000020  167744          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2613                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2614  011746  042777  020000  167752          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2615  011754  052777  020000  167744          BIS     #CLK,@TXCSR     ;POKE CLK UP
2616                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2617  011762  042777  020000  167736          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2618  011770  052777  020000  167730          BIS     #CLK,@TXCSR     ;POKE CLK UP
2619  011776  016703  167714                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2620  012002  012700  000037                  MOV     #37,R0  ;EXPECTED
2621  012006  012767  000007  167106          MOV     #7,SHIFT        ;# OF SHIFTS
2622  012014  012767  000176  167456          MOV     #176,$TMP1      ;DATA CHAR
2623  012022  004767  005124                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2624  012026  105777  167660                  TSTB    @RXCSR  ;RXDONE ?
2625  012032  100401                          BMI     .+4
2626  012034  104004                          ERROR   4               ;RXDONE SHOULD BE SET
2627  012036  017701  167654                  MOV     @RXDBUF,R1      ;ACTUAL
2628  012042  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2629  012044  001401                          BEQ     .+4
2630  012046  104002                          ERROR   2               ;RECEIVED DATA DID NOT MATCH
2631                                                                  ;EXPECTED DATA - CHECK MAINT DATA
```

```
2632                                                    ;OR RECEIVER LOGIC
2633  012050  012767  000007  167044          MOV     #7,SHIFT        ;# OF SHIFTS
2634  012056  012767  000176  167414          MOV     #176,$TMP1      ;DATA CHAR
2635  012064  004767  005062                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2636                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2637  012070  012767  000007  167024          MOV     #7,SHIFT        ;# OF SHIFTS
2638  012076  012767  000176  167374          MOV     #176,$TMP1      ;DATA CHAR
2639  012104  004767  005042                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2640  012110  012700  140037                  MOV     #140000!37,R0   ;EXPECTED DATA PLUS
2641                                                                  ;RXERR & OVRRUN
2642  012114  017701  167576                  MOV     @RXDBUF,R1      ;ACTUAL
2643  012120  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
2644  012122  001401                          BEQ     .+4
2645  012124  104002                          ERROR   2               ;SPECIFICALLY LOOK AT RXERR &
2646                                                                  ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2647
2648                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2649                                          ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2650                                          ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2651                                          ;;(OVRRUN,RXERR)
2652                                          ;;MODE: ISYMOD
2653                                          ;;LENGTH: FIVE
2654                                          ;;CHAR: 0
2655                                          ;;
2656                                  ;;**************************************************************************
2657  012126  000004                  TST24:  SCOPE
2658  012130  052777  000400  167570          BIS     #MRESET,@TXCSR  ;MASTER RESET
2659  012136  012777  000000  167556          MOV     #ISYMOD,@PARCSR ;SET THE MODE
2660  012144  052777  000400  167554          BIS     #MRESET,@TXCSR  ;MASTER RESET
2661
2662                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2663  012152  012777  064001  167546          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2664
2665                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2666  012160  012777  000000  167534          MOV     #ISYMOD!FIVE!NOPAR!0,@PARCSR
2667  012166  052777  000020  167516          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2668                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2669  012174  042777  020000  167524          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2670  012202  052777  020000  167516          BIS     #CLK,@TXCSR     ;POKE CLK UP
2671                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2672  012210  042777  020000  167510          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2673  012216  052777  020000  167502          BIS     #CLK,@TXCSR     ;POKE CLK UP
2674  012224  016703  167466                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2675  012230  012700  000000                  MOV     #0,R0   ;EXPECTED
2676  012234  012767  000007  166660          MOV     #7,SHIFT        ;# OF SHIFTS
2677  012242  012767  000100  167230          MOV     #100,$TMP1      ;DATA CHAR
2678  012250  004767  004676                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2679  012254  105777  167432                  TSTB    @RXCSR  ;RXDONE ?
2680  012260  100401                          BMI     .+4
2681  012262  104004                          ERROR   4               ;RXDONE SHOULD BE SET
2682  012264  017701  167426                  MOV     @RXDBUF,R1      ;ACTUAL
2683  012270  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
2684  012272  001401                          BEQ     .+4
2685  012274  104002                          ERROR   2               ;RECEIVED DATA DID NOT MATCH
2686                                                                  ;EXPECTED DATA - CHECK MAINT DATA
2687                                                                  ;OR RECEIVER LOGIC
```

```
2688  012276  012767  000007  166616          MOV     #7,SHIFT        ;# OF SHIFTS
2689  012304  012767  000100  167166          MOV     #100,$TMP1      ;DATA CHAR
2690  012312  004767  004634                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2691                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2692  012316  012767  000007  166576          MOV     #7,SHIFT        ;# OF SHIFTS
2693  012324  012767  000100  167146          MOV     #100,$TMP1      ;DATA CHAR
2694  012332  004767  004614                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2695  012336  012700  140000                  MOV     #140000!0,R0    ;EXPECTED DATA PLUS
2696                                                                  ;RXERR & OVRRUN
2697  012342  017701  167350                  MOV     @RXDBUF,R1      ;ACTUAL
2698  012346  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
2699  012350  001401                          BEQ     .+4
2700  012352  104002                          ERROR   2       ;SPECIFICALLY LOOK AT RXERR &
2701                                                          ;OVRRUN BITS...THEY BOTH SHOULD BE SET
2702
2703
2704                                  ;END OF PASS
2705                                  ;TYPE NAME OF TEST
2706                                  ;UPDATE PASS COUNT
2707                                  ;CHECK FOR EXIT TO ACT-11
2708                                  ;RESTART TEST
2709
2710  012354  000004          .EOP:   SCOPE
2711  012356  004767  000340          JSR     PC,CKSWR
2712  012362  104401                  TYPE                            ;TYPE NAME OF TEST
2713  012364  015512                  MEPASS
2714  012366  104413  012620          CONVRT  ,OUTCRY
2715  012372  104401  015331          TYPE    ,DEVICE
2716  012376  105767  166550          TSTB    MULTD   ;ARE YOU RUNNING MULTIPLE DEVICES ?
2717  012402  001511                  BEQ     CCC     ;NO,JUMP AROUND
2718  012404  005767  166556          TST     ACTREG  ;ARE ANY DEVICES ACTIVE ?
2719  012410  001007                  BNE     RUNIT   ;YES
2720  012412  104401  015343          TYPE    ,MCOW   ;NO
2721  012416  016700  166544          MOV     ACTREG,R0       ;DISPLAY ACTREG
2722  012422  000000                  HALT                    ;SELECT SOMETHING TO RUN @ ACTREG:
2723                                  ;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
2724  012424  000167  167522          JMP     .START  ;START OVER AGAIN.....YOU DESELECTED EVERYTHING
2725  012430  062767  000010  166516  RUNIT:  ADD     #10,BASEADD     ;NEXT BLOCK (ADDRESSES)
2726  012436  062767  000010  166516  ZERO:   ADD     #10,BASEIV      ;NEXT BLOCK (VECTORS)
2727  012444  000241                  CLC
2728  012446  006167  166516          ROL     ROTADD  ;UP DATE ROTATING POINTER
2729  012452  103410                  BCS     2$      ;IS IT THE LAST DEVICE
2730                                                  ;TO BE TESTED IN THIS PASS ?
2731  012454  036767  166510  166504          BIT     ROTADD,ACTREG   ;TEST THIS DEVICE FOR ACTIVE STATUS
2732  012462  001762                  BEQ     RUNIT   ;IF NOT ACTIVE, TRY NEXT ADDRESS
2733  012464  004767  000034          JSR     PC,REPLAY       ;CALCULATE NEW PARAMETERS
2734  012470  000167  000210          JMP     RESTRT  ;YES IT WAS ACTIVE,TEST THIS DEVICE
2735  012474  012767  000001  166466  2$:     MOV     #1,ROTADD       ;OK!,NOW SET UP  ROTATING
2736                                                  ;POINTER FOR NEXT MULTIPLE PASS
2737  012502  016767  166450  166444          MOV     KEEPADD,BASEADD ;RESTORE BASE ADDRESS
2738  012510  016767  166450  166444          MOV     KEEPIV,BASEIV   ;RESTORE BASE INTERRUPT VECTORS
2739  012516  004767  000002          JSR     PC,REPLAY       ;CALC NEW PARAMETERS
2740  012522  000441                  BR      CCC     ;JUMP AROUND REPLAY
2741  012524  016767  166424  004416  REPLAY: MOV     BASEADD,DUBASE  ;SET UP FOR NEW ADDRESSES
2742  012532  004767  004260          JSR     PC,DUADDR       ;CREATE NEW ADDRESSES
2743  012536  016767  166420  167172          MOV     BASEIV,DURIV    ;CREATE DURIV
```

```
2744  012544  062767  000002  166410          ADD     #2,BASEIV
2745  012552  016767  166404  167160          MOV     BASEIV,DURIS    ;CREATE DURIS
2746  012560  062767  000002  166374          ADD     #2,BASEIV
2747  012566  016767  166370  167146          MOV     BASEIV,DUTIV    ;CREATE DUTIV
2748  012574  062767  000002  166360          ADD     #2,BASEIV
2749  012602  016767  166354  167134          MOV     BASEIV,DUTIS    ;CREATE DUTIS
2750  012610  016767  167122  166344          MOV     DURIV,BASEIV    ;RESTORE
2751  012616  000207                          RTS     PC
2752
2753  012620  000001                  OUTCRY: 1
2754  012622     006     002                  .BYTE   6,2
2755  012624  001712                          RXCSR
2756
2757  012626                          CCC:
2758  012626  005067  166550                  CLR     $TSTNM                  ;CLEAR TEST NUMBER
2759  012632  005067  166560                  CLR     $ERRPC                  ;CLEAR LAST ERROR PC
2760  012636  005067  166541                  CLR     $ERFLG                  ;CLEAR ERROR FLAG
2761  012642  005267  166244                  INC     PASCNT                  ;UPDATE PASS COUNT
2762  012646  016767  166240  166226          MOV     PASCNT,LIGHTS           ;DISPLAY PASS COUNT
2763  012654  016767  166232  166652          MOV     PASCNT,$PASS            ;PASS COUNT TO APT
2764  012662  013701  000042                  MOV     @#42,R1                 ;CHECK FOR ACT-11 OR DDP
2765  012666  001406                          BEQ     RESTRT          ;IF NO CONTINUE TESTING
2766  012670  000005                          RESET
2767  012672  000005                          RESET
2768  012674  004711                  $ENDAD: JSR     PC,(R1)
2769  012676  000240                          NOP
2770  012700  000240                          NOP
2771  012702  000240                          NOP
2772  012704                          RESTRT:
2773  012704  012767  003376  166474          MOV     #TST1+2,$LPADR  ;LOAD LAST ADDR
2774  012712  004767  000004                  JSR     PC,CKSWR
2775  012716  000167  170366                  JMP     .BEGIN
2776
2777                                          ;CHECK SWITCH REGISTER ROUTINE.
2778                                          ;CHECKS TO ALLOW FOR < G> TO ALLOW
2779                                          ;THE CHANGING OF LOCATION 176
2780
2781  012722  005737  000042          CKSWR:  TST     @#42
2782  012726  001040                          BNE     OUT
2783  012730  022767  000176  166502          CMP     #SWREG,SWR      ;SOFTWARE SWR PRESENT?
2784  012736  001034                          BNE     OUT             ;NO--LEAVE
2785  012740  105777  166500                  TSTB    @$TKS           ;CHECK TTY READY
2786  012744  100031                          BPL     OUT             ;NO--LEAVE
2787  012746  017767  166474  000422          MOV     @$TKB,..MSG     ;GET CHARACTER
2788  012754  042767  177600  000414          BIC     #177600,..MSG   ;STRIP JUNK
2789  012762  122767  000007  000406          CMPB    #7,..MSG        ;IS IT < G> ?
2790  012770  001017                          BNE     OUT             ;NO
2791  012772  104401  016117                  TYPE    ,MCNTG
2792  012776  005137  013036          CNTLU:  COM     @#RDSW
2793  013002  104401  016127                  TYPE    ,MMSWR
2794  013006  104413                          CONVRT
2795  013010  013040                          SWREGL
2796  013012  104406  016140                  INSTR,MMNEW
2797  013016  104410                          PARAM
2798  013020  000000                          0
2799  013022  177777                          177777
```

```
                                                SWREG
                                .BYTE   0,1
                                OUT:    CLR     @#RDSW
                                        RTS     PC
                                RDSW:   .WORD   0
                                SWREGL: 1
                                        .BYTE   6,2
                                        SWREG

                                        5

                                        ;CHECK FOR FREEZE ON CURRENT DATA

                                .SCOP1: JSR     PC,CKSWR
                                        BIT     #SW09,@SWR
                                        BEQ     1$
                                        MOV     LOCK,(SP)
                                1$:     RTI
                                .SBTTL  TYPE ROUTINE

                                ;;**********************************************************************
                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                ;*
                                ;*CALL:
                                ;*1) USING A TRAP INSTRUCTION
                                ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;*OR
                                ;*      TYPE
                                ;*      MESADR
                                ;*

                                $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
                                        BPL     1$              ;;BR IF YES
                                        HALT                    ;;HALT HERE IF NO TERMINAL
                                        BR      3$              ;;LEAVE
                                1$:     MOV     R0,-(SP)        ;;SAVE R0
                                        MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
                                        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
                                        BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
                                        BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
                                        BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
                                        MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
                                        JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
                                61$:    .WORD   0               ;;MESSAGE ADDRESS
                                62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
                                        BNE     60$             ;;YES,SKIP TYPE OUT
                                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
                                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
                                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
                                60$:    MOV     (SP)+,R0        ;;RESTORE R0
                                3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
                                        RTI                     ;;RETURN
```

```
2856  013172  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
2857  013176  001430                          BEQ     8$
2858  013200  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
2859  013204  001006                          BNE     5$
2860  013206  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
2861  013210  104401                          TYPE                    ;;TYPE A CR AND LF
2862  013212  001523                          $CRLF
2863  013214  105067  000130                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
2864  013220  000755                          BR      2$              ;;GET NEXT CHARACTER
2865  013222  004767  000056          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
2866  013226  126726  166224          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
2867  013232  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
2868  013234  016746  166214                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
2869                                                                  ;;AND THE NULL CHAR.
2870  013240  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
2871  013244  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
2872  013246  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
2873  013252  105367  000072                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
2874  013256  000770                          BR      7$              ;;LOOP
2875
2876                                  ;HORIZONTAL TAB PROCESSOR
2877
2878  013260  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
2879  013264  004767  000014          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
2880  013270  132767  000007  000052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
2881  013276  001372                          BNE     9$              ;;TAB STOP
2882  013300  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
2883  013302  000724                          BR      2$              ;;GET NEXT CHARACTER
2884  013304  105777  166140          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
2885  013310  100375                          BPL     $TYPEC
2886  013312  116677  000002  166132          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
2887  013320  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
2888  013326  001003                          BNE     1$              ;;BRANCH IF NO
2889  013330  105067  000014                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
2890  013334  000406                          BR      $TYPEX          ;;EXIT
2891  013336  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
2892  013344  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
2893  013346  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
2894  013350  000000          $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
2895  013352  000207          $TYPEX: RTS     PC
2896
2897
2898                                          ;ASCII STRING INPUT ROUTINE
2899
2900  013354  017667  000000  000014  .INSTR: MOV     @(SP)+,.MSG     ;PICK UP MESSAGE
2901  013362  062716  000002                  ADD     #2,(SP)         ;JUMP AROUND MESSAGE FOR RTI
2902  013366  105767  166154                  TSTB    $ENV            ;APT CONTROL
2903  013372  001036                          BNE     INSTR2          ;YES NO TYPE
2904  013374  104401                  .INST1: TYPE
2905  013376  000000                  .MSG:   0
2906  013400  012704  016152                  MOV     #INBUF,R4       ;GET STARTING LOC OF INBUF
2907  013404  012703  000007                  MOV     #7,R3           ;MAX # OF CHARS
2908  013410  105777  166030          1$:     TSTB    @$TKS   ;TTY FLAG
2909  013414  100375                          BPL     1$
2910  013416  117714  166024                  MOVB    @$TKB,(R4)      ;TAKE CHAR
2911  013422  142714  000200                  .BICB   #200,(R4)       ;STRIP
```

```
2912  013426  121427  000025          CMPB    (R4),#25        ;IS IT < G>
2913  013432  001760                  BEQ     .INST1
2914  013434  122427  000015          CMPB    (R4)+,#15       ;CHECK FOR CR
2915  013440  001413                  BEQ     INSTR2
2916  013442  105777  166002  2$:     TSTB    @$TPS   ;TEST FLAG
2917  013446  100375                  BPL     2$
2918  013450  117777  165772  165774  MOVB    @$TKB,@$TPB     ;ECHO CHARACTER
2919  013456  005303                  DEC     R3      ;DID YOU TYPE TOO MANY CHARS ?
2920  013460  001353                  BNE     1$
2921  013462  104401          .INSTE: TYPE
2922  013464  015437                  MQM     ;?
2923  013466  000742                  BR      .INST1  ;RETRY
2924  013470  000002          INSTR2: RTI
2925
2926                                  ;CONVERT ASCII STRING TO OCTAL
2927
2928  013472  011605          .PARAM: MOV     (SP),R5 ;PUT CONTENTS OF SP INTO R5
2929  013474  012567  000162          MOV     (R5)+,LOLIM     ;PUT LOW LIMIT INTO LOLIM
2930  013500  012567  000160          MOV     (R5)+,HILIM     ;PUT HIGH LIMIT INTO HILIM
2931  013504  012567  000156          MOV     (R5)+,DEVADR    ;PUT STORE LOC INTO DEVADR
2932  013510  112567  000154          MOVB    (R5)+,LOBITS    ;PUT MASK INTO LOBITS
2933  013514  112567  000151          MOVB    (R5)+,ADRCNT    ;PUT COUNT INTO ADRCNT
2934  013520  010516                  MOV     R5,(SP) ;RESTORE RETURN ADDR ON STACK FOR RTI
2935  013522  005005          PARAM1: CLR     R5
2936  013524  012704  016152          MOV     #INBUF,R4
2937  013530  122714  000015          CMPB    #15,(R4)        ;CR ?
2938  013534  001420                  BEQ     PARERR  ;YOU TYPED CR TOO SOON !
2939  013536  121427  000060  1$:     CMPB    (R4),#60        ;LOW LIMIT ASCII 0
2940  013542  002415                  BLT     PARERR
2941  013544  121427  000067          CMPB    (R4),#67        ;HIGH LIMIT ASCII 7
2942  013550  003012                  BGT     PARERR
2943  013552  142714  000060          BICB    #60,(R4)        ;CONVERT TO OCTAL
2944  013556  152405                  BISB    (R4)+,R5        ;STORE AWAY ITS AN OK CHAR
2945  013560  122714  000015          CMPB    #15,(R4)        ;CR ?
2946  013564  001414                  BEQ     LIMITS  ;NOW CHECK FOR HIGH &LOW LIMIT CONDS
2947  013566  006305                  ASL     R5      ;ALLOCATE ROOM FOR NEXT CHAR
2948  013570  006305                  ASL     R5
2949  013572  006305                  ASL     R5
2950  013574  000760                  BR      1$
2951  013576  122714  000015  PARERR: CMPB    #15,(R4)        ;CR?
2952  013602  001003                  BNE     120$
2953  013604  005737  013036          TST     @#RDSW          ;CK SWR USED
2954  013610  001023                  BNE     PARTI
2955  013612  104407          120$:   INSTER  ;RETRY
2956  013614  000742                  BR      PARAM1
2957
2958                                  ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2959
2960  013616  020567  000042  LIMITS: CMP     R5,HILIM
2961  013622  101365                  BHI     PARERR  ;THE # IS TOO HIGH
2962  013624  020567  000032          CMP     R5,LOLIM
2963  013630  103762                  BLO     PARERR  ;THE # IS TOO LOW
2964  013632  136705  000032          BITB    LOBITS,R5       ;TEST BY MASKINGTHE #
2965  013636  001357                  BNE     PARERR
2966
2967                                  ;STORE NUMBER AT SPECIFIED ADDRESS
```

```
                                                        ;GET STARTING ADDR OF
                                                        ;STORE AT THIS ADDR
                                                        ;HOW MANY TIMES + 2 ?
                                                        ;SAVE PC OF TEST THAT FAILED AND R0-R5
                                                        ;SAVE R0-R5
                                                        ;RESTORE R0-R5
                                                        ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
                                                        ;CR  LF
                                                        ;PICK UP DATA POINTER
                                                        ;SET UP SP FOR RTI
                                                        ;PICK UP # OF WORDS FROM TABLE
                                                        ;PICK UP # OF CHARS FROM TABLE
                                                        ;PICK UP # OF SPACES FROM TABLE
                                                        ;PICK UP ADDRESS OF  MSG
                                                        ;FROM TABLE
                                                        ;SAVE
                                                        ;SAVE
                                                        ;STARTING ADDRESS OF TEMP BLOCK
                                                        ;SAVE
                                                        ;CLR OUT UPPER BITS .. SAVE CHAR
                                                        ;CONVERT TO ASCII
                                                        ;STORE AWAY
                                                        ;SHIFT FOR NEXT #
```

```
3024  014052  006204                        ASR     R4      ;DITTO
3025  014054  006204                        ASR     R4      ;DITTO
3026  014056  005305                        DEC     R5      ;DEC CHAR COUNT
3027  014060  001365                        BNE     3$      ;DO IT AGAIN ?
3028  014062  012703  016256                MOV     #MDATA,R3       ;STARTING ADDRESS OF MDATA BLOCK
3029  014066  114023              4$:       MOVB    -(R0),(R3)+     ;REVERSE THE ORDER OF NUMBERS
3030  014070  105367  000042                DECB    CHRCNT  ;DEC CHAR COUNT
3031  014074  001374                        BNE     4$      ;DO IT AGAIN ?
3032  014076  105767  000035                TSTB    SPACNT  ;HOW MANY SPACES ?
3033  014102  001405                        BEQ     6$      ;TYPE # IF BR =0
3034  014104  112723  000240          5$:   MOVB    #240,(R3)+      ;"SPACE" IN ASCII
3035  014110  105367  000023                DECB    SPACNT  ;DEC # OF SPACE COUNT
3036  014114  001373                        BNE     5$      ;DO IT AGAIN ?
3037  014116  105013              6$:       CLRB    (R3)    ;INSERT "0" FOR TTY OUTPUT ROUTINE
3038  014120  104401                        TYPE
3039  014122  016256                        MDATA   ;THIS MESSAGE
3040  014124  005367  000004                DEC     WRDCNT  ;HOW MANY #'S ?
3041  014130  001325                        BNE     1$      ;DO THIS ROUTINE AGAIN IF NOT EQUAL TO 0
3042  014132  000002                        RTI     ;RETURN TO PROGRAM
3043  014134  000000              WRDCNT: 0
3044  014136  000000              CHRCNT: 0
3045          014137              SPACNT=CHRCNT+1
3046  014140  000000              BINWRD: 0
3047
3048                                        ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
3049                                        ;BUFFER TO THE CHARACTERS "N" AND "Y".
3050                                        ;IF THE CHARACTER IS "N" CLEAR THE FLAG
3051                                        ;IF THE CHARACTER IS "Y" SET THE FLAG
3052
3053  014142  017605  000000        .SETFLG:MOV     @(SP),R5
3054  014146  122767  000116  001776        CMPB    #'N,INBUF       ;IS IT "N" ?
3055  014154  001002                        BNE     1$
3056  014156  105015                        CLRB    (R5)    ;000
3057  014160  000406                        BR      2$
3058  014162  122767  000131  001762  1$:   CMPB    #'Y,INBUF       ;IS IT "Y" ?
3059  014170  001005                        BNE     3$
3060  014172  112715  177777                MOVB    #-1,(R5)        ;377
3061  014176  062716  000002          2$:   ADD     #2,(SP)
3062  014202  000002                        RTI
3063  014204  104407              3$:       INSTER  ;RETRY
3064  014206  000755                        BR      .SETFLG
3065                                .SBTTL  ERROR HANDLER ROUTINE
3066
3067                                ;;**********************************************************************
3068                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
3069                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
3070                                ;*AND GO TO SAVIT ON ERROR
3071                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3072                                ;*SW15=1         HALT ON ERROR
3073                                ;*SW13=1         INHIBIT ERROR TYPEOUTS
3074                                ;*SW10=1         BELL ON ERROR
3075                                ;*SW09=1         LOOP ON ERROR
3076                                ;*CALL
3077                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
3078
3079  014210                        $ERROR:
```

```
3080  014210  105267  165167          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
3081  014214  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
3082  014216  016777  165160  165216          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
3083  014224  032777  002000  165206          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
3084  014232  001402                          BEQ     1$              ;;NO - SKIP
3085  014234  104401  001516                  TYPE    ,$BELL          ;;RING BELL
3086  014240  005267  165146          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
3087  014244  011667  165146                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
3088  014250  162767  000002  165140          SUB     #2,$ERRPC
3089  014256  117767  165134  165130          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
3090  014264  032777  020000  165146          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
3091  014272  001004                          BNE     20$             ;;SKIP TYPEOUTS
3092  014274  004767  000072                  JSR     PC,SAVIT        ;;GO TO USER ERROR ROUTINE
3093  014300  104401  001523                  TYPE    ,$CRLF
3094  014304                          20$:
3095  014304  122767  000001  165234          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
3096  014312  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
3097  014314  116767  165074  000004          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
3098  014322  004767  000016'                 JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
3099  014326     000          21$:    .BYTE   0
3100  014327     000                  .BYTE   0
3101  014330  000777          22$:    BR      22$             ;;APT ERROR LOOP
3102  014332  005777  165102    2$:   TST     @SWR            ;;HALT ON ERROR
3103  014336  100001                  BPL     3$              ;;SKIP IF CONTINUE
3104  014340  000000                  HALT                    ;;HALT ON ERROR!
3105  014342  032777  001000  165070  3$:   BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
3106  014350  001402                  BEQ     4$              ;;BR IF NO
3107  014352  016716  165032          MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
3108  014356  005767  165132    4$:   TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
3109  014362  001402                  BEQ     5$              ;;BR IF NONE
3110  014364  016716  165124          MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
3111  014370                    5$:
3112  014370  000002                  RTI                     ;;RETURN
3113  014372  010067  164532  SAVIT: MOV     R0,HLD0
3114  014376  010167  164530          MOV     R1,HLD1
3115  014402  010267  164526          MOV     R2,HLD2
3116  014406  010367  164524          MOV     R3,HLD3
3117  014412  010467  164522          MOV     R4,HLD4
3118  014416  010567  164520          MOV     R5,HLD5
3119  014422  016767  164754  164514  MOV     $TSTNM,HLD6
3120
3121                            .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
3122
3123                            ;;**********************************************************************
3124                            ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
3125                            ;*ERROR IS TO BE REPORTED.  IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
3126                            ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
3127
3128  014430                    $ERRTYP:
3129  014430  104401  001523          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3130  014434  010046                  MOV     R0,-(SP)        ;;SAVE R0
3131  014436  005000                  CLR     R0              ;;PICKUP THE ITEM INDEX
3132  014440  153700  001414          BISB    @#$ITEMB,R0
3133  014444  001004                  BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
3134                                                          ;;TYPE THE PC OF THE ERROR
3135  014446  016746  164744          MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
```

```
3136
3137  014452  104402                        TYPOC                   ;;ERROR ADDRESS
3138  014454  000426                        BR      6$              ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3139  014456  005300                        DEC     R0              ;;GET OUT
3140  014460  006300                  1$:   ASL     R0              ;;ADJUST THE INDEX SO THAT IT WILL
3141  014462  006300                        ASL     R0              ;;      WORK FOR THE ERROR TABLE
3142  014464  006300                        ASL     R0
3143  014466  062700  001652              ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
3144  014472  012067  000004              MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
3145  014476  001404                        BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
3146  014500  104401                        TYPE                    ;;TYPE THE "ERROR MESSAGE"
3147  014502  000000                  2$:   .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
3148  014504  104401  001523              TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3149  014510  012067  000004            3$:   MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
3150  014514  001404                        BEQ     5$              ;;SKIP TYPEOUT IF 0
3151  014516  104401                        TYPE                    ;;TYPE THE "DATA HEADER"
3152  014520  000000                  4$:   .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
3153  014522  104401  001523              TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3154  014526  011000                  5$:   MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
3155  014530  001004                        BNE     7$              ;;GO TYPE THE DATA
3156  014532  012600                  6$:   MOV     (SP)+,R0        ;;RESTORE R0
3157  014534  104401  001523              TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
3158  014540  000207                        RTS     PC              ;;RETURN
3159  014542                            7$:
3160  014542  013046                        MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
3161  014544  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3162  014546  005710                        TST     (R0)            ;;IS THERE ANOTHER NUMBER?
3163  014550  001770                        BEQ     6$              ;;BR IF NO
3164  014552  104401  014560              TYPE    ,8$             ;;TYPE TWO(2) SPACES
3165  014556  000771                        BR      7$              ;;LOOP
3166  014560  020040      000           8$:   .ASCIZ  / /             ;;TWO(2) SPACES
3167            014564                      .EVEN
3168                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3169
3170                                  ;;**********************************************************************
3171                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3172                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
3173                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3174                                  ;*CALL:
3175                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3176                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
3177                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3178                                  ;*      .BYTE   M               ;;M=1 OR 0
3179                                  ;*                                      ;;1=TYPE LEADING ZEROS
3180                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
3181                                  ;*
3182                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3183                                  ;*$TYPOS OR $TYPOC
3184                                  ;*CALL:
3185                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3186                                  ;*      TYPON                   ;;CALL FOR TYPEOUT
3187                                  ;*
3188                                  ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3189                                  ;*CALL:
3190                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3191                                  ;*      TYPOC                   ;;CALL FOR TYPEOUT
```

```
3192
3193  014564  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
3194  014570  116667  000001  000211          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
3195  014576  112667  000207                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
3196  014602  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
3197  014606  000406                          BR      $TYPON
3198  014610  112767  000001  000171 $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
3199  014616  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
3200  014624  112767  000005  000154 $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
3201  014632  010346                          MOV     R3,-(SP)        ;;SAVE R3
3202  014634  010446                          MOV     R4,-(SP)        ;;SAVE R4
3203  014636  010546                          MOV     R5,-(SP)        ;;SAVE R5
3204  014640  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
3205  014644  005404                          NEG     R4
3206  014646  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
3207  014652  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
3208  014656  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
3209  014662  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
3210  014666  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
3211  014670  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
3212  014672  000404                          BR      3$              ;;GO DO MSB
3213  014674  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
3214  014676  006105                          ROL     R5
3215  014700  006105                          ROL     R5
3216  014702  010503                          MOV     R5,R3
3217  014704  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
3218  014706  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
3219  014712  100016                          BPL     7$              ;;BR IF NO
3220  014714  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
3221  014720  001002                          BNE     4$              ;;TEST FOR 0
3222  014722  005704                          TST     R4              ;;SUPPRESS THIS 0?
3223  014724  001403                          BEQ     5$              ;;BR IF YES
3224  014726  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
3225  014730  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
3226  014734  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
3227  014740  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
3228  014744  104401  015004                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
3229  014750  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
3230  014754  003347                          BGT     2$              ;;BR IF MORE TO DO
3231  014756  002402                          BLT     6$              ;;BR IF DONE
3232  014760  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
3233  014762  000744                          BR      2$              ;;GO DO THE LAST DIGIT
3234  014764  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
3235  014766  012604                          MOV     (SP)+,R4        ;;RESTORE R4
3236  014770  012603                          MOV     (SP)+,R3        ;;RESTORE R3
3237  014772  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
3238  015000  012616                          MOV     (SP)+,(SP)
3239  015002  000002                          RTI                     ;;RETURN
3240  015004     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
3241  015005     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
3242  015006     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
3243  015007     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
3244  015010  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
3245                                  ;ENTER HERE ON POWER FAILURE
3246
3247
```

```
3248  015012                          $PWRDN:
3249  015012  010046                  .PFAIL: MOV     R0,-(SP)            ;SAVE R0-R5 ON PROCESSOR STACK
3250  015014  010146                          MOV     R1,-(SP)
3251  015016  010246                          MOV     R2,-(SP)
3252  015020  010346                          MOV     R3,-(SP)
3253  015022  010446                          MOV     R4,-(SP)
3254  015024  010546                          MOV     R5,-(SP)
3255  015026  016746  162772                  MOV     24,-(SP)
3256  015032  010667  164060                  MOV     SP,SAVSP            ;SAVE STACK POINTER
3257  015036  012767  015050  162760          MOV     #RESTART,24         ;SET UP FOR POWER UP TRAP
3258  015044  000000                          HALT                        ;HALT ON POWER DOWN NORMAL
3259  015046  000777                          BR      .
3260
3261                                          ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3262
3263  015050  016706  164042          RESTAR: MOV     SAVSP,SP            ;RESTORE STACK POINTER
3264  015054  012605                          MOV     (SP)+,R5            ;RESTORE R0-R5
3265  015056  012604                          MOV     (SP)+,R4
3266  015060  012603                          MOV     (SP)+,R3
3267  015062  012602                          MOV     (SP)+,R2
3268  015064  012601                          MOV     (SP)+,R1
3269  015066  012600                          MOV     (SP)+,R0
3270  015070  012767  015012  162726          MOV     #.PFAIL,24          ;SET UP FOR POWER FAILURE
3271  015076  106427  000340                  MTPS    #340
3272  015102  012706  001100                  MOV     #STACK,SP
3273  015106  005067  001102                  CLR     TEMP
3274  015112  005267  001076                  INC     TEMP
3275  015116  001375                          BNE     .-4
3276  015120  104413                          CONVRT
3277  015122  015144                          PFTAB
3278  015124  104401                          TYPE
3279  015126  015446                          MPFAIL
3280  015130  005067  164247                  CLR     $ERFLG
3281  015134  005067  164256                  CLR     $ERRPC
3282  015140  000177  163740                  JMP     @RETURN
3283  015144  000001                  PFTAB:  1
3284  015146     006     002                  .BYTE   6,2
3285  015150  000207                          RETURN
3286  015152  005015  042012  053125  MTITLE: .ASCIZ  <15><12><12>/DUV11 DZDUV-B TAPE F /<15><12>
3287  015160  030461  042040  042132
3288  015166  053125  041055  052040
3289  015174  050101  020105  020106
3290  015202  005015     000
3291  015205     015  053012  041505  MVECTO: .ASCIZ  <15><12>/VEC ADD-/
3292  015212  040440  042104  000055
3293  015220  005015  051461  020124  MREGAD: .ASCIZ  <15><12>/1ST DEV: REC CSR ADD-/
3294  015226  042504  035126  051040
3295  015234  041505  041440  051123
3296  015242  040440  042104  000055
3297  015250  005015  052515  052114  MMULT:  .ASCIZ  <15><12>/MULT DEV ? (Y OR N)-/
3298  015256  042040  053105  037440
3299  015264  024040  020131  051117
3300  015272  047040  026451     000
3301  015277     015  046012  051501  MLASTD: .ASCIZ  <15><12>/LAST DEV: REC CSR ADDR-/
3302  015304  020124  042504  035126
3303  015312  051040  041505  041440
```

```
3304  015320  051123  040440  042104
3305  015326  026522     000
3306  015331     075  042504  044526  DEVICE: .ASCIZ  /=DEVICE  /
3307  015336  042503  020040     000
3308  015343     015  051412  046105  MCOW:   .ASCIZ  <15><12>/SELECT TO RUN @ACTREG/
3309  015350  041505  020124  047524
3310  015356  051040  047125  040040
3311  015364  041501  051124  043505
3312  015372     000
3313  015373     015  047412  043126  MRANGE: .ASCIZ  <15><12>/OVFLO: RETYPE LAST DEV RXCSR ADDS-/
3314  015400  047514  051072  052105
3315  015406  050131  020105  040514
3316  015414  052123  042040  053105
3317  015422  051040  041530  051123
3318  015430  040440  042104  026523
3319  015436     000
3320  015437     040  037440     000  MQM:    .ASCIZ  /  ?/
3321  015443     015  000012          MCRLF:  .ASCIZ   <15><12>
3322  015446  043120  044501  026114  MPFAIL: .ASCIZ  /PFAIL,  RESTART AT TEST IN PROGRESS/
3323  015454  020040  042522  052123
3324  015462  051101  020124  052101
3325  015470  052040  051505  020124
3326  015476  047111  050040  047522
3327  015504  051107  051505  000123
3328  015512  005015  047105  020104  MEPASS: .ASCIZ  <15><12>/END OF PASS TAPE F/
3329  015520  043117  050040  051501
3330  015526  020123  040524  042520
3331  015534  043040     000
3332  015537     015  051012     000  MR:     .ASCIZ  <15><12>/R/
3333  015543     015  052012  051505  MTSTPC. .ASCIZ  <15><12>/TEST PC-/
3334  015550  020124  041520  000055
3335  015556  005015  047514  045503  MLOCK:  .ASCIZ  <15><12>/LOCK ON  TEST? (Y OR N)-/
3336  015564  047440  020116  052040
3337  015572  051505  037524  024040
3338  015600  020131  051117  047040
3339  015606  026451     000
3340  015611     015  021412  047440  MSYNC:  .ASCIZ  <15><12>/# OF SYNC CHARS SELECTED ( 1 OR 2)-/
3341  015616  020106  054523  041516
3342  015624  041440  040510  051522
3343  015632  051440  046105  041505
3344  015640  042524  020104  020050
3345  015646  020061  051117  031040
3346  015654  026451     000
3347  015657     015  044412  020123  MWIRE6: .ASCIZ  <15><12>/IS SEC XMIT SWITCH E55-2 IN? (Y OR N)-/
3348  015664  042523  020103  046530
3349  015672  052111  051440  044527
3350  015700  041524  020110  032505
3351  015706  026465  020062  047111
3352  015714  020077  054450  047440
3353  015722  020122  024516  000055
3354  015730  005015  051511  051440  MWIRE5: .ASCIZ  <15><12>/IS SEC REC SWITCH E55-3 IN? (Y OR N)-/
3355  015736  041505  051040  041505
3356  015744  051440  044527  041524
3357  015752  020110  032505  026465
3358  015760  020063  047111  020077
3359  015766  054450  047440  020122
```

```
3360  015774  024516  000055
3361  016000  005015  051511  047440  MWIRE4: .ASCIZ  <15><12>/IS OPT CLR ENABLE SWITCH E55-1 IN? (Y OR N)-/
3362  016006  052120  041440  051114
3363  016014  042440  040516  046102
3364  016022  020105  053523  052111
3365  016030  044103  042440  032465
3366  016036  030455  044440  037516
3367  016044  024040  020131  051117
3368  016052  047040  026451     000
3369  016057     015  005012  031510  MEXTJ:  .ASCIZ  <15><12><12>/H315 CONNECTOR ON ?(Y OR N)-/
3370  016064  032461  041440  047117
3371  016072  042516  052103  051117
3372  016100  047440  020116  024077
3373  016106  020131  051117  047040
3374  016114  026451     000
3375  016117     015  020012  043536  MCNTG:  .ASCIZ  <15><12>/  G  /
3376  016124  020040     000
3377  016127     040  053523  036522  MMSWR:  .ASCIZ  / SWR=   /
3378  016134  020040  000040
3379  016140  020040  047040  053505  MMNEW:  .ASCIZ  /   NEW=  /
3380  016146  020075  000040
3381                                  .EVEN
3382
3383                                          ;BUFFERS FOR INPUT-OUTPUT
3384
3385  016152  000000                  INBUF:  0
3386          016214                  .=.+40
3387  016214  000000                  TEMP:   0
3388          016256                  .=.+40
3389  016256  000000                  MDATA:  0
3390          016320                  .=.+40
3391                                  .SBTTL  SCOPE HANDLER ROUTINE
3392
3393                                  ;;**********************************************************************
3394                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS.  IT WILL INCREMENT
3395                                  ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG. (DISPLAY<7:0>)
3396                                  ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
3397                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3398                                  ;*SW14=1         LOOP ON TEST
3399                                  ;*SW11=1         INHIBIT ITERATIONS
3400                                  ;*SW09=1         LOOP ON ERROR
3401                                  ;*SW08=1         LOOP ON TEST IN SWR<7:0>
3402                                  ;*CALL
3403                                  ;*      SCOPE           ;;SCOPE=IOT
3404
3405  016320                          $SCOPE:
3406
3407                                          ;SCOPE LOOP AND INTERATION HANDLER
3408
3409  016320                          .SCOPE:
3410  016320  004767  174376                  JSR     PC,CKSWR
3411  016324  005067  163066                  CLR     $ERRPC          ;CLEAR LAST ERROR PC
3412  016330  022716  003376                  CMP     #TST1+2,(SP)    ;IS SCOPE AT BEGINING OF TEST 1?
3413  016334  001422                          BEQ     $XTSTR          ;YES NO LOOP.
3414
3415  016336  032777  040000  163074  TTST:   BIT     #BIT14,@SWR     ;THIS CODE IS FOR TESTING FOR BIT 14
```

```
3416  016344  001412                          BEQ     1$                ;ON LSI WHICH SYSMAC CANNOT HANDLE
3417  016346  016767  163030  163032          MOV     $TSTNM,$LPADR
3418  016354  000406                          BR      1$
3419  016356  105777  163062                  TSTB    @$TKS             ;KEYBOARD DONE?
3420  016362  100123                          BPL     $OVER             ;BR IF NO
3421  016364  017766  163056  177776          MOV     @$TKB,-2(SP)      ;CLEAR DONE BIT
3422  016372  032777  040000  163040  1$:     BIT     #BIT14,@SWR       ;;LOOP ON PRESENT TEST?
3423  016400  001114                          BNE     $OVER             ;;YES IF SW14=1
3424                                  ;#####START OF CODE FOR THE XOR TESTER#####
3425  016402  000416                  $XTSTR: BR      6$                ;;IF RUNNING ON THE "XOR" TESTER CHANGE
3426                                                                    ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
3427  016404  013746  000004                  MOV     @#ERRVEC,-(SP)    ;;SAVE THE CONTENTS OF THE ERROR VECTOR
3428  016410  012737  016430  000004          MOV     #5$,@#ERRVEC      ;;SET FOR TIMEOUT
3429  016416  005737  177060                  TST     @#177060          ;;TIME OUT ON XOR?
3430  016422  012637  000004                  MOV     (SP)+,@#ERRVEC    ;;RESTORE THE ERROR VECTOR
3431  016426  000463                          BR      $SVLAD            ;;GO TO THE NEXT TEST
3432  016430  022626                  5$:     CMP     (SP)+,(SP)+       ;;CLEAR THE STACK AFTER A TIME OUT
3433  016432  012637  000004                  MOV     (SP)+,@#ERRVEC    ;;RESTORE THE ERROR VECTOR
3434  016436  000423                          BR      7$                ;;LOOP ON THE PRESENT TEST
3435  016440                          6$: ;#####END OF CODE FOR THE XOR TESTER#####
3436  016440  032777  000400  162772          BIT     #BIT08,@SWR       ;;LOOP ON SPEC. TEST?
3437  016446  001404                          BEQ     2$                ;;BR IF NO
3438  016450  127767  162764  162724          CMPB    @SWR,$TSTNM       ;;ON THE RIGHT TEST?    SWR<7:0>
3439  016456  001465                          BEQ     $OVER             ;;BR IF YES
3440  016460  105767  162717          2$:     TSTB    $ERFLG            ;;HAS AN ERROR OCCURRED?
3441  016464  001421                          BEQ     3$                ;;BR IF NO
3442  016466  126767  162723  162707          CMPB    $ERMAX,$ERFLG     ;;MAX. ERRORS FOR THIS TEST OCCURRED?
3443  016474  101015                          BHI     3$                ;;BR IF NO
3444  016476  032777  001000  162734          BIT     #BIT09,@SWR       ;;LOOP ON ERROR?
3445  016504  001404                          BEQ     4$                ;;BR IF NO
3446  016506  016767  162676  162672  7$:     MOV     $LPERR,$LPADR     ;;SET LOOP ADDRESS TO LAST SCOPE
3447  016514  000446                          BR      $OVER
3448  016516  105067  162661          4$:     CLRB    $ERFLG            ;;ZERO THE ERROR FLAG
3449  016522  005067  162764                  CLR     $TIMES            ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
3450  016526  000415                          BR      1$                ;;ESCAPE TO THE NEXT TEST
3451  016530  032777  004000  162702  3$:     BIT     #BIT11,@SWR       ;;INHIBIT ITERATIONS?
3452  016536  001011                          BNE     1$                ;;BR IF YES
3453  016540  005767  162770                  TST     $PASS             ;;IF FIRST PASS OF PROGRAM
3454  016544  001406                          BEQ     1$                ;;      INHIBIT ITERATIONS
3455  016546  005267  162632                  INC     $ICNT             ;;INCREMENT ITERATION COUNT
3456  016552  026767  162734  162624          CMP     $TIMES,$ICNT      ;;CHECK THE NUMBER OF ITERATIONS MADE
3457  016560  002024                          BGE     $OVER             ;;BR IF MORE ITERATION REQUIRED
3458  016562  012767  000001  162614  1$:     MOV     #1,$ICNT          ;;REINITIALIZE THE ITERATION COUNTER
3459  016570  016767  000056  162714          MOV     $MXCNT,$TIMES     ;;SET NUMBER OF ITERATIONS TO DO
3460  016576  105267  162600          $SVLAD: INCB    $TSTNM            ;;COUNT TEST NUMBERS
3461  016602  116767  162574  162722          MOVB    $TSTNM,$TESTN     ;;SET TEST NUMBER IN APT MAILBOX
3462  016610  011667  162572                  MOV     (SP),$LPADR       ;;SAVE SCOPE LOOP ADDRESS
3463  016614  011667  162570                  MOV     (SP),$LPERR       ;;SAVE ERROR LOOP ADDRESS
3464  016620  005067  162670                  CLR     $ESCAPE           ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
3465  016624  112767  000001  162563          MOVB    #1,$ERMAX         ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
3466  016632  016777  162544  162602  $OVER:  MOV     $TSTNM,@DISPLAY   ;;DISPLAY TEST NUMBER
3467  016640  016716  162542                  MOV     $LPADR,(SP)       ;;FUDGE RETURN ADDRESS
3468  016644  000002                  4$:     RTI
3469  016646  001407                  BRW:    1407
3470  016650  000432                  BRX:    432
3471  016652  000005                  $MXCNT: 5                         ;;MAX. NUMBER OF ITERATIONS
```

```
                                .SBTTL  TRAP DECODER

                                ;;**********************************************************************
                                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                ;*GO TO THAT ROUTINE.

016654  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
016656  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
016662  005740                          TST     -(R0)           ;;BACKUP BY 2
016664  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
016666  006300                          ASL     R0              ;;POSITION FOR INDEXING
016670  016000  016710                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
016674  000200                          RTS     R0              ;;GO TO ROUTINE

                                ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

016676  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
016700  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
016706  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                ;*BY THE "TRAP" INSTRUCTION.

                                ;       ROUTINE
                                ;       -------
016710  016676                  $TRPAD: .WORD   $TRAP2
016712  013072                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
016714  014610                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
016716  014564                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
016720  014624                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)

016722  013050                          .SCOP1  ;;CALL=SCOP1    TRAP+5(104405)
016724  013354                          .INSTR  ;;CALL=INSTR    TRAP+6(104406)
016726  013462                          .INSTER ;;CALL=INSTER   TRAP+7(104407)
016730  013472                          .PARAM  ;;CALL=PARAM    TRAP+10(104410)
016732  013672                          .SAV05  ;;CALL=SAV05    TRAP+11(104411)
016734  013732                          .RES05  ;;CALL=RES05    TRAP+12(104412)
016736  013764                          .CONVRT ;;CALL=CONVRT   TRAP+13(104413)
016740  014142                          .SETFLG ;;CALL=SETFLG   TRAP+14(104414)
                                ;**********************************************************************
                                ;UTILITIES
                                ;**********************************************************************

                                ;THIS UTILITY CALCULATES PRIORITY LEVEL
016742  006367  000044          DULEV:  ASL     DUPRT   ;SHIFT LEFT
016746  006367  000040                  ASL     DUPRT   ;
016752  006367  000034                  ASL     DUPRT   ;
016756  006367  000030                  ASL     DUPRT   ;
016762  006367  000024                  ASL     DUPRT   ;
016766  016767  000020  000020          MOV     DUPRT,LESS1     ;MOVE THIS TO LESS1
```

```
3528  016774  162767  000001  000012          SUB     #1,LESS1        ;CREATE LESS1
3529  017002  042767  000037  000004          BIC     #37,LESS1       ;CLEAR TNZVC
3530  017010  000207                          RTS     PC
3531  017012  000240                  DUPRT:  PR5
3532  017014  000200                  LESS1:  PR4     ;LEVEL TO ALLOW INTERRUPTS
3533
3534                                          ;NEW DU ADDRESSES
3535  017016  016767  000126  162666  DUADDR: MOV     DUBASE,RXCSR    ;XXX0
3536  017024  005267  000120                  INC     DUBASE
3537  017030  016767  000114  162656          MOV     DUBASE,HRXCSR   ;XXX1
3538  017036  005267  000106                  INC     DUBASE
3539  017042  016767  000102  162646          MOV     DUBASE,RXDBUF   ;XXX2
3540  017050  016767  000074  162644          MOV     DUBASE,PARCSR   ;XXX2
3541  017056  005267  000066                  INC     DUBASE
3542  017062  016767  000062  162630          MOV     DUBASE,HRXDBUF  ;XXX3
3543  017070  016767  000054  162626          MOV     DUBASE,HPARCSR  ;XXX3
3544  017076  005267  000046                  INC     DUBASE
3545  017102  016767  000042  162616          MOV     DUBASE,TXCSR    ;XXX4
3546  017110  005267  000034                  INC     DUBASE
3547  017114  016767  000030  162606          MOV     DUBASE,HTXCSR   ;XXX5
3548  017122  005267  000022                  INC     DUBASE
3549  017126  016767  000016  162576          MOV     DUBASE,TXDBUF   ;XXX6
3550  017134  005267  000010                  INC     DUBASE
3551  017140  016767  000004  162566          MOV     DUBASE,HTXDBUF  ;XXX7
3552  017146  000207                          RTS     PC
3553  017150  000000                  DUBASE: 0
3554
3555                                          ;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3556                                          ;INFORMATION CONTAINED IN $TMP1 AND IT IS
3557                                          ;SHIFTED IN BY THE CONTENTS OF SHIFT
3558  017152  042777  040000  162546  RPOKE:  BIC     #MTDATA,@TXCSR
3559  017160  005067  162316                  CLR     $TMP2
3560  017164  006067  162310                  ROR     $TMP1   ;FORCE CARRY
3561  017170  006067  162306                  ROR     $TMP2   ;PICK UP CARRY IN BIT 15
3562  017174  006267  162302                  ASR     $TMP2   ;SHIFT INTO BIT 14
3563  017200  042767  100000  162274          BIC     #BIT15,$TMP2    ;CLR BIT 15
3564  017206  056777  162270  162512          BIS     $TMP2,@TXCSR    ;POKE MAINT DATA
3565  017214  042777  020000  162504          BIC     #CLK,@TXCSR     ;POKE CLK
3566  017222  052777  020000  162476          BIS     #CLK,@TXCSR     ;
3567  017230  005367  161666                  DEC     SHIFT
3568  017234  001346                          BNE     RPOKE
3569  017236  000207                          RTS     PC
3570
3571                                          ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3572  017240  016767  162234  162234  ODD8:   MOV     $TMP1,$TMP2     ;SAVE TEMP1
3573  017246  005067  162232                  CLR     $TMP3
3574  017252  012727  000010                  MOV     #8.,(PC)+
3575  017256  000000                  4$:     0
3576  017260  006067  162216          1$:     ROR     $TMP2
3577  017264  005567  162214                  ADC     $TMP3
3578  017270  005367  177762                  DEC     4$
3579  017274  001371                          BNE     1$
3580  017276  006067  162202                  ROR     $TMP3
3581  017302  103404                          BCS     2$
3582  017304  052767  000400  162166          BIS     #BIT8,$TMP1     ;SET ODD PARITY
3583  017312  000403                          BR      3$
```

```
3584  017314  042767  000400  162156  2$:     BIC     #BIT8,$TMP1     ;CLR EVEN PARITY
3585                                          ;$TMP1 NOW HAS ODD PARITY CHARACTER
3586  017322  000207                  3$:     RTS     PC
3587
3588                                  ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3589  017324  016767  162150  162150  EVEN8:  MOV     $TMP1,$TMP2     ;SAVE TEMP1
3590  017332  005067  162146                  CLR     $TMP3
3591  017336  012727  000010                  MOV     #8.,(PC)+
3592  017342  000000                  4$:     0
3593  017344  006067  162132          1$:     ROR     $TMP2
3594  017350  005567  162130                  ADC     $TMP3
3595  017354  005367  177762                  DEC     4$
3596  017360  001371                          BNE     1$
3597  017362  006067  162116                  ROR     $TMP3
3598  017366  103004                          BCC     2$
3599  017370  052767  000400  162102          BIS     #BIT8,$TMP1     ;SET EVEN PARITY
3600  017376  000403                          BR      3$
3601  017400  042767  000400  162072  2$:     BIC     #BIT8,$TMP1     ;CLR ODD PARITY
3602                                          ;$TMP1 NOW HAS EVEN PARITY CHARACTER
3603  017406  000207                  3$:     RTS     PC
3604  017410  062716  000002          TRPREG: ADD     #2,(SP) ;ALLOW IT TO "CRUNCH" INTO HLT BACK
3605                                                          ;IN MAIN PART OF THE PROGRAM
3606  017414  000002                          RTI
3607          000001                  .END
```

```
AAA     003200           1293#
ABASE = 000000            876     917
ACDW1 = 000000            876     919
ACDW2 = 000000            876     920
ACPUOP= 000000            876     891
ACTREG  001166            735#   1249*   1263*   1264*   1271*   2718    2721    2731
ADDW0 = 000000            876     921
ADDW1 = 000000            876     922
ADDW10= 000000            876     931
ADDW11= 000000            876     932
ADDW12= 000000            876     933
ADDW13= 000000            876     934
ADDW14= 000000            876     935
ADDW15= 000000            876     936
ADDW2 = 000000            876     923
ADDW3 = 000000            876     924
ADDW4 = 000000            876     925
ADDW5 = 000000            876     926
ADDW6 = 000000            876     927
ADDW7 = 000000            876     928
ADDW8 = 000000            876     929
ADDW9 = 000000            876     930
ADEVCT= 000000            876     882
ADEVM = 000000            876     918
ADRCNT= 013671           2933*   2972*   2979#
AENV  = 000000            876     887
AENVM = 000000            876     888
AFATAL= 000000            876     879
AMADR1= 000000            876     904
AMADR2= 000000            876     908
AMADR3= 000000            876     911
AMADR4= 000000            876     914
AMAMS1= 000000            876     898
AMAMS2= 000000            876     906
AMAMS3= 000000            876     909
AMAMS4= 000000            876     912
AMSGAD= 000000            876     884
AMSGLG= 000000            876     885
AMSGTY= 000000            876     878
AMTYP1= 000000            876     899
AMTYP2= 000000            876     907
AMTYP3= 000000            876     910
AMTYP4= 000000            876     913
APASS = 000000            876     881
APRIOR= 000000            876
APTCSU= 000040            536#    2848
APTENV= 000001            536#    2841    3095
APTSIZ= 000200            536#    1168
APTSPO= 000100            536#    2843
ASWREG= 000000            876     889
ATESTN= 000000            876     880
AUNIT = 000000            876     883
AUSWR = 000000            876     890
AVECT1= 000000            876     915
AVECT2= 000000            876     916
BASEAD  001154            730#   1231*   1268*   1269    1275*   1277*   2725*   2737*   2741
```

```
BASEIV  001162            733#   1241*  2726*  2738*  2743   2744*  2745   2746*  2747   2748*  2749   2750*
BBB     003026           1248    1252#
BINWRD  014140           3014*   3016   3046#
BITW  = 002000            802#    995#
BIT0  = 000001            665#    778    809    971   1002
BIT00 = 000001            655#    665
BIT01 = 000002            654#    664
BIT02 = 000004            653#    663
BIT03 = 000010            652#    662
BIT04 = 000020            651#    661
BIT05 = 000040            650#    660
BIT06 = 000100            649#    659
BIT07 = 000200            648#    658
BIT08 = 000400            647#    657   3436
BIT09 = 001000            646#    656   3105   3444
BIT1  = 000002            664#    777    970
BIT10 = 002000            645#    768    802    961    995   3083
BIT11 = 004000            644#    767    960   3451
BIT12 = 010000            643#    766    783    959    976
BIT13 = 020000            642#    765    782    801    958    975    994   3090
BIT14 = 040000            641#    764    781    800    957    974    993   3415   3422
BIT15 = 100000            640#    763    780    799    956    973    992   3563
BIT2  = 000004            663#    776    969
BIT3  = 000010            662#    775    808    968   1001
BIT4  = 000020            661#    774    807    967   1000
BIT5  = 000040            660#    773    806    966    999
BIT6  = 000100            659#    772    805    965    998
BIT7  = 000200            658#    771    804    964    997
BIT8  = 000400            657#    770    786    803    963    979    996   3582   3584   3599   3601
BIT9  = 001000            656#    769    785    962    978
BPTVEC= 000014            672#
BREAK = 000001            809#   1002#   1357   1430   2553   2608   2663
BRW     016646           3469#
BRX     016650           3470#
CARDET= 010000            766#    959#
CCC     012626           2717    2740   2757#
CHRCNT  014136           3012*   3017   3030*  3044#  3045
CKSWR   012722           2711    2774   2781#  2813   3410
CLK   = 020000            801#    994#   1357   1363   1364   1392   1393   1430   1436   1437   1485   1486   1527
                         1528    1531   1532   1540   1709   1710   1713   1714   1738   1992   1993   1996   1997
                         2011    2012   2038   2075   2076   2079   2080   2094   2095   2121   2158   2159   2162
                         2163    2177   2178   2204   2241   2242   2245   2246   2260   2261   2287   2324   2325
                         2329    2330   2372   2373   2377   2378   2422   2428   2437   2443   2493   2499   2508
                         2514    2553   2559   2560   2562   2563   2608   2614   2615   2617   2618   2663   2669
                         2670    2672   2673   3565   3566
CNTLU   012776           1198    2792#
CONVRT= 104413           2714    2794   3276   3515#
COUNT   001124            710#   1876*  1880*  1914   1918*  1959   2004*  2015*  2087*  2098*  2170*  2181*  2253*
                         2264*
CR    = 000015            580#    2887   2897
CRLF  = 000200            581#    2858   2897
CTS   = 020000            765#    958#
DDISP = 177570            587#    843    1156
DEVADR  013666           2931*   2969   2977#
DEVICE  015331           2715    3306#
DF1     002132           1027    1031   1035   1039   1085#
```

```
DH1     002067          1025    1029    1033    1074#
DISPLA  001442           843#   1156*   1164*   1189*   3082*   3466*
DISPRE  000174           684#   1164    1189
DNA    = 100000          799#    992#
DNAINT= 000040           806#    999#   1538    1551    1560    1720    1745    1757    1764
DSC    = 100000          763#    956#
DSINTE= 000040           773#    966#   1458    1470    1474    1498
DSR    = 001000          769#    962#
DSWR   = 177570           586#    842    1155
DTR    = 000002          777#    970#   1451
DT1     002116          1026    1030    1034    1080#
DT4     002126          1038    1083#
DUADDR  017016          1230    2742    3535#
DUBASE  017150          1226    1229    2741*   3535    3536*   3537    3538*   3539    3540    3541*   3542    3543    3544*
                        3545    3546*   3547    3548*   3549    3550*   3551    3553#
DULEV   016742          1288    3522#
DUPRT   017012          1287*   1366    1439    1536    1573    1610    1653    1718    1790    1870    1872    3522*   3523*
                        3524*   3525*   3526*   3527    3531#
DURIS   001740          1054#   1366*   1376    1377*   1400    1401*   1439*   1461    1462*   1493    1494*   1790*   1813
                        1814*   1821    1822*   1870*   1896    1897*   1907    1908*   1937    1938*   2745*
DURIV   001736          1053#   1237    1240    1241    1365*   1376*   1400*   1438*   1449*   1461*   1493*   1789*   1813*
                        1821*   1869*   1896*   1907*   1937*   2743*   2750
DUTIS   001744          1056#   1536*   1553    1554*   1557    1558*   1573*   1590    1591*   1594    1595*   1610*   1618
                        1619*   1626    1627*   1653*   1682    1683*   1718*   1748    1749*   1761    1762*   1872*   1898
                        1899*   1909    1910*   1939    1940*   2749*
DUTIV   001742          1055#   1535*   1553*   1557*   1572*   1590*   1594*   1609*   1618*   1626*   1652*   1664*   1682*
                        1717*   1736*   1748*   1761*   1871*   1898*   1909*   1939*   2747*
EIGHT  = 006000           794#    987#   1360    1433    1523    1705    1785    1855    2235    2318    2366    2416    2487
EMTVEC= 000030           675#   1140*   1141*
EM1     001762          1024    1060#
EM2     002022          1028    1066#
EM3     002043          1032    1069#
EM4     001746          1036    1058#
ERRCNT  001114           703#
ERRVEC= 000004           668#   1153    1154*   1165*   3427    3428*   3430*   3433*
EVEN8   017324          3589#
EVEPAR= 001400           797#    990#   2318    2416
EVPAR  = 000400           786#    979#
FIVE   = 000000           791#    984#   1986    2556    2611    2666
FRMERR= 020000           782#    975#
GNS    = ****** U       3503    3504    3505    3506    3509    3510    3511    3512    3513    3514    3515    3516
HDXEN  = 000010           808#   1001#
HILIM   013664          2930*   2960    2976#
HLD0    001130           712#   1080    3113*
HLD1    001132           713#   1080    3114*
HLD2    001134           714#   3115*
HLD3    001136           715#   3116*
HLD4    001140           716#   3117*
HLD5    001142           717#   3118*
HLD6    001144           718#   3119*
HOLD    001120           708#   2424    2430    2439    2445    2495    2501    2510    2516
HPARCS  001724          1047#   3543*
HRXCSR  001714          1043#   3537*
HRXDBU  001720          1045#   3542*
HT     = 000011           578#   2856    2897
HTXCSR  001730          1049#   3547*
```

```
HTXDBU  001734           1051#   3551*
INBUF   016152           1296    1300    2906    2936    3054    3058    3385#
INIFLG  001172            743#   1199    1202*
INSTER= 104407           1304    2955    3063    3511#
INSTR = 104406           1221    1232    1242    1253    1278    1294    1307    1311    1315    1319    1330    2796    3510#
INSTR2  013470           2903    2915    2924#
IOTVEC= 000020            673#   1138*   1139*
ISYMOD= 000000            790#    983#   2549    2556    2604    2611    2659    2666
JMRBY   001153            726#   1207*   1322    1422    2405    2476
KEEPAD  001156            731#   1229*   1231    1275    1277    2737
KEEPIV  001164            734#   1240*   2738
LASTAD  001160            732#   1258    1269    1283
LESS1   017014           3527*   3528*   3529*   3532#
LF    = 000012            579#   2891    2897
LIGHTS  001102            693#   2762*
LIMITS  013616           2946    2960#
LOBITS  013670           2932*   2964    2978#   2979
LOCK    001110            701#   2816
LOKFLG  001174            745#
LOLIM   013662           2929*   2962    2975#
MCNTG   016117           2791    3375#
MCOW    015343           2720    3308#
MCRLF   015443           3008    3321#
MDATA   016256           3028    3039    3389#
MEPASS  015512           2713    3328#
MEXT  = 010000            813#   1006#   2413    2484
MEXTJ   016057           1320    3369#
MINT  = 004000            812#   1005#   1357    1430    1520    1702    1987    2070    2153    2236    2315    2363    2553
                         2608    2663
MLASTD  015277           1254    3301#
MLOCK   015556           3335#
MMNEW   016140           2796    3379#
MMSWR   016127           2793    3377#
MMULT   015250           1243    3297#
MPFAIL  015446           3279    3322#
MQM     015437           2922    3320#
MR      015537           1341    3332#
MRANGE  015373           1279    3313#
MREGAD  015220           1222    3293#
MRESET= 000400            803#    996#   1352    1354    1425    1427    1514    1516    1571    1608    1651    1696    1698
                         1782    1784    1846    1848    1983    2043    2066    2126    2149    2209    2232    2292    2309
                         2311    2357    2359    2407    2409    2478    2480    2548    2550    2603    2605    2658    2660
MSYNC   015611           1295    3340#
MTDATA= 040000            800#    993#   1357    1387    1391    1430    1480    1484    2553    2608    2663    3558
MTITLE  015152           1201    3286#
MTSTPC  015543           1331    3333#
MULTD   001152            725#   1245    1246    2716
MVECTO  015205           1233    3291#
MWIRE4  016000           1316    3361#
MWIRE5  015730           1312    3354#
MWIRE6  015657           1308    3347#
NEXT    001106            700#
NOPAR = 000000            795#    988#   1360    1433    1523    1705    1785    1855    1986    2069    2152    2235    2556
                         2611    2666
ODDPAR= 001000            796#    989#   2366    2487
ODD8    017240           3572#
```

```
ONCE    002610          1200    1203#
OPTCLR  001151           724#   1318
OUT     013030          2782    2784    2786    2790    2802#
OUTCRY  012620          2714    2753#
OUTMUL  003166          1251    1276    1287#
OVRRUN= 040000           781#    974#
PARAM = 104410          1223    1234    1255    1280    1332    2797    3512#
PARAM1  013522          2935#   2956
PARCSR  001722          1046#   1353*   1360*   1426*   1433*   1515*   1523*   1697*   1705*   1783*   1785*   1847*   1855*
                        1988*   2002*   2071*   2085*   2154*   2168*   2237*   2251*   2310*   2318*   2358*   2366*   2408*
                        2416*   2479*   2487*   2549*   2556*   2604*   2611*   2659*   2666*   3540*
PAREN = 001000           785#    978#
PARER = 010000           783#    976#
PARERR  013576          2938    2940    2942    2951#   2961    2963    2965
PARTI   013660          2954    2974#
PASCNT  001112           702#   1176*   2761*   2762    2763
PFTAB   015144          3277    3283#
PIRQ  = 177772           585#
PIRQVE= 000240           679#
POPRO = 012600           758#    951#
POP1SP= 005726           756#    949#
POP2SP= 022626           760#    953#
PR0   = 000000           602#
PR1   = 000040           603#
PR2   = 000100           604#
PR3   = 000140           605#
PR4   = 000200           606#   3532
PR5   = 000240           607#   3531
PR6   = 000300           608#
PR7   = 000340           609#
PS    = 177776           582#    583
PSW   = 177776           583#
PUSHR0= 010046           757#    950#
PUSH1S= 005746           755#    948#
PUSH2S= 024646           759#    952#
PWRVEC= 000024           674#   1144*   1145*
RDSW    013036          2792*   2802*   2804#   2953
RECACT= 004000           767#    960#   1998    2020    2081    2103    2164    2186    2247    2269
REPLAY  012524          2733    2739    2741#
RESTAR  015050          3257    3263#
RESTRT  012704          2734    2765    2772#
RESVEC= 000010           669#
RESOS = 104412          3514#
RETURN  001104           699#   1181*   1338*   1340*   1342    3282
RING  = 040000           764#    957#
RINTEN= 000100           772#    965#   1371    1381    1405    1458    1470    1474    1498    1796    1816    1819    1881
                        1895    1912    1936
ROTADD  001170           738#   1250*   1262*   1264    1266*   1271    1274*   2728*   2731    2735*
RPOKE   017152          2568    2580    2584    2623    2635    2639    2678    2690    2694    3558#   3568
RTS   = 000004           776#    969#
RUNA  = ****** U            1      51     140     513
RUNB  = ****** U            1      51     140     513
RUNC  = ****** U            1      51     140     513
RUND  = ****** U            1      51     140     513
RUNE  = ****** U            1      51     140     513
RUNF  = 000000              1      51     140     513
```

```
RUNIT   012430          2719    2725#   2732
RXCSR   001712          1042#   1361*   1371*   1372    1381*   1405*   1434*   1445    1451*   1458*   1467    1470*   1474*
                        1498*   1796*   1811    1816*   1819*   1828    1856*   1881*   1895*   1905    1912*   1936*   1946
                        1989*   1998    2020    2044    2072*   2081    2103    2127    2155*   2164    2186    2210    2238*
                        2247    2269    2293    2322*   2338    2370*   2386    2420*   2456    2491*   2527    2557*   2569
                        2612*   2624    2667*   2679    2755    3535*
RXDBUF  001716          1044#   1378    1402    1404    1452    1495    1497    1792    1812    1874    1906    1984    2025
                        2067    2108    2150    2191    2233    2274    2319    2341    2367    2389    2417    2459    2488
                        2530    2564    2572    2587    2619    2627    2642    2674    2682    2697    3539*
RXDONE= 000200           771#    964#
RXERR = 100000           780#    973#
SAVIT   014372          3092    3113#
SAVPC   001126           711#   2983*
SAVSP   001116           704#   3256*   3263
SAV05 = 104411          3513#
SCOP1 = 104405          2034    2117    2200    2283    3509#
SEND  = 000020           807#   1000#   1520    1702    1794    1852    1987    2070    2153    2236    2315    2363    2413
                        2484
SEREC   001150           723#   1314
SETFLG= 104414          1244    1309    1313    1317    1321    3516#
SEVEN = 004000           793#    986#   2152
SEXMIT  001147           722#   1310
SHIFT   001122           709#   1382*   1394*   1475*   1487*   1525*   1533*   1707*   1715*   2009*   2013*   2092*   2096*
                        2175*   2179*   2258*   2262*   2326*   2331*   2332    2336    2374*   2379*   2380    2384    2434*
                        2449*   2450    2454    2505*   2520*   2521    2525    2566*   2578*   2582*   2621*   2633*   2637*
                        2676*   2688*   2692*   3567*
SIX   = 002000           792#    985#   2069
SPACNT= 014137          3013*   3032    3035*   3045#
SRD   = 002000           768#    961#
STACK = 001100           573#   1136    1172    1326    3272
STD   = 000010           775#    968#
STFLG   001173           744#   1175*
STKLMT= 177774           584#
STPSYN= 000400           770#    963#   1856
SV05    013700          2987#
SWR     001440           842#   1134    1155*   1157    1163*   1170*   1186    1188*   1193    1209    1328    2783    2814
                        3083    3090    3102    3105    3415    3422    3436    3438    3444    3451
SWREG   000176           685#   1163    1188    1193    2783    2800    2807
SWREGL  013040          2795    2805#
SW0   = 000001           637#
SW00  = 000001           627#    637    1209
SW01  = 000002           626#    636    1328
SW02  = 000004           625#    635
SW03  = 000010           624#    634
SW04  = 000020           623#    633
SW05  = 000040           622#    632
SW06  = 000100           621#    631
SW07  = 000200           620#    630
SW08  = 000400           619#    629
SW09  = 001000           618#    628    2814
SW1   = 000002           636#
SW10  = 002000           617#
SW11  = 004000           616#
SW12  = 010000           615#
SW13  = 020000           614#
SW14  = 040000           613#
```

```
SW15   = 100000            612#
SW2    = 000004            635#
SW3    = 000010            634#
SW4    = 000020            633#
SW5    = 000040            632#
SW6    = 000100            631#
SW7    = 000200            630#
SW8    = 000400            629#
SW9    = 001000            628#
SYNCNO  001146             721#   1298*  1302*  1878   2018   2101   2184   2267
SYNEXT= 020000             789#    982#  1353   1360   1426   1433   1515   1523   1697   1705   1783   1785   2310
                          2318    2358   2366   2408   2416   2479   2487
SYNINT= 030000             788#    981#  1847   1855   1986   2069   2152   2235
SYNSCH= 000020             774#    967#  1361   1434   1796   1856   1989   2044   2072   2127   2155   2210   2238
                          2293    2322   2370   2420   2491   2557   2612   2667
SYSTST= 014000             814#   1007#  1794   1852
TBITVE= 000014             670#
TEMP    016214            3018    3273*  3274*  3387#
TKVEC  = 000060            677#
TPVEC  = 000064            678#
TRAPVE= 000034             676#   1142*  1143*
TRPREG  017410            3604#
TRTVEC= 000014             671#
TST1    003374            1333    1340   1350#  2773   3412
TST10   005622            1778#
TST11   006112            1845#
TST12   006726            1982#
TST13   007232            2065#
TST14   007536            2148#
TST15   010042            2231#
TST16   010346            2308#
TST17   010544            2356#
TST2    003720            1421#
TST20   010742            2404#
TST21   011206            2475#
TST22   011452            2547#
TST23   011700            2602#
TST24   012126            2657#
TST3    004336            1512#
TST4    004610            1569#
TST5    004752            1607#
TST6    005110            1650#
TST7    005260            1694#
TTST    016336            3415#
TXCSR   001726            1048#  1352*  1354*  1357*  1363*  1364*  1387*  1391*  1392*  1393*  1425*  1427*  1430*
                          1436*  1437*  1480*  1484*  1485*  1486*  1514*  1516*  1520*  1527*  1528*  1531*  1532*
                          1538*  1540*  1548   1551*  1560*  1571*  1575*  1579*  1585*  1586   1608*  1612*  1616*
                          1623*  1651*  1655*  1672   1684*  1696*  1698*  1702*  1709*  1710*  1713*  1714*  1720*
                          1729   1732   1738*  1745*  1754   1757*  1764*  1782*  1784*  1794*  1846*  1848*  1852*
                          1882*  1894*  1913*  1927*  1983*  1987*  1992*  1993*  1996*  1997*  2005   2011*  2012*
                          2038*  2039   2043*  2066*  2070*  2075*  2076*  2079*  2080*  2088   2094*  2095*  2121*
                          2122   2126*  2149*  2153*  2158*  2159*  2162*  2163*  2171   2177*  2178*  2204*  2205
                          2209*  2232*  2236*  2241*  2242*  2245*  2246*  2254   2260*  2261*  2287*  2288   2292*
                          2309*  2311*  2315*  2324*  2325*  2329*  2330*  2357*  2359*  2363*  2372*  2373*  2377*
                          2378*  2407*  2409*  2413*  2422*  2428*  2437*  2443*  2478*  2480*  2484*  2493*  2499*
                          2508*  2514*  2548*  2550*  2553*  2559*  2560*  2562*  2563*  2603*  2605*  2608*  2614*
```

```
                        2615*   2617*   2618*   2658*   2660*   2663*   2669*   2670*   2672*   2673*   3545*   3558*   3564*
                        3565*   3566*
TXDBUF  001732          1050#   1524*   1706*   1739*   1788*   1885*   1926*   1994*   2003*   2077*   2086*   2160*   2169*
                        2243*   2252*   2321*   2335*   2369*   2383*   2419*   2453*   2490*   2524*   3549*
TXDONE= 000200           804#    997#
TXINTE= 000100           805#    998#   1575    1579    1585    1612    1616    1623    1655    1684    1720    1745    1757
                        1764    1882    1894    1913    1927
TYPE  = 104401          1201    1341    2712    2715    2720    2791    2793    2861    2904    2921    3007    3038    3085
                        3093    3129    3146    3148    3151    3153    3157    3164    3228    3278    3503#
TYPOC = 104402          3137    3161    3504#
TYPON = 104404          3506#
TYPOS = 104403          3505#
USER  = 000000           811#   1004#
VOID  = 000001           778#    971#
WRDCNT  014134          3011*   3040*   3043#
ZERO    012436          2726#
$APTHD  002136          1107    1113#
$ASTAT= ****** U         536
$ATYC   000024R          536#
$ATY1   000000R          536#
$ATY3   000006R          536#   2846
$ATY4   000016R          536#   3098
$AUTOB  001434           839#
$BASE   001602           917#
$BDADR  001422           834#
$BDDAT  001426           836#
$BELL   001516           868#   3085    3113
$CDW1   001606           919#
$CDW2   001610           920#
$CHARC  013350          2863*   2873*   2880    2889*   2894#
$CKSWR= ****** U        3509
$CMTAG  001400           822#   1131    1132    1140    1146    1147
$CM1  = 000006           854#    855#    856#    857#    858#    859#    860#
$CM2  = 000014           854#    855#    856#    857#    858#    859#    860#
$CM3  = 000006           852#    854
$CM4  = 000006           860#    861#    862#    863#    864#    865#    866#
$CPUOP  001554           891#
$CRLF   001523           870#   2862    2897    3093    3113    3129    3148    3153    3157
$DDW0   001612           921#
$DDW1   001614           922#
$DDW10  001636           931#
$DDW11  001640           932#
$DDW12  001642           933#
$DDW13  001644           934#
$DDW14  001646           935#
$DDW15  001650           936#
$DDW2   001616           923#
$DDW3   001620           924#
$DDW4   001622           925#
$DDW5   001624           926#
$DDW6   001626           927#
$DDW7   001630           928#
$DDW8   001632           929#
$DDW9   001634           930#
$DEVCT  001536           882#
$DEVM   001604           918#
```

```
$E     = 000002            534#
$ENDAD  012674            1094    2768#
$ENV    001546             536     887#   1203    2841    2902    3095
$ENVM   001547             536     888#   1168    2843    2848
$ERFLG  001403             825#   1177*   2760*   3080*   3113    3280*   3396    3440    3442    3448*   3472
$ERMAX  001415             831#   1148*   3442    3465*   3472
$ERROR  014210            1140    3079#
$ERRPC  001416             832#   1080    1083    1179*   2759*   3087*   3088*   3089    3113    3135    3281*   3411*
$ERRTB  001652            1022#   3143
$ERRTY  014430            3128#
$ERTTL  001412             829#   1178*   3086*   3113
$ESCAP  001514             867#   1147*   3108    3110    3113    3464*
$ETABL  001546             886#
$ETEND  001652             939#   1119
$FATAL  001530             536*    879#
$FFLG   000244R            536#*
$FILLC  001456             850#   2866    2897
$FILLS  001455             849#   2897
$GDADR  001420             833#
$GDDAT  001424             835#
$GTSWR= ****** U           3508
$HIBTS  002136            1114#
$ICNT   001404             826#   3455*   3456    3458*   3471
$INTAG  001435             840#
$ITEMB  001414             830#   3089*   3097    3113    3132
$LF     001524             871#   2897    3113
$LFLG   000243R            536#*
$LPADR  001406             827#   1149*   2773*   3417*   3446*   3462*   3467    3471
$LPERR  001410             828#   1150*   3107    3446    3463*   3471
$MADR1  001560             904#
$MADR2  001564             908#
$MADR3  001570             911#
$MADR4  001574             914#
$MAIL   001526             877#   1115    1119    1167    2841    3095    3461
$MAMS1  001556             898#
$MAMS2  001562             906#
$MAMS3  001566             909#
$MAMS4  001572             912#
$MBADR  002140            1115#
$MFLG   000242R            536#*
$MSGAD  001542             536*    884#
$MSGLG  001544             536*    885#
$MSGTY  001526             536*    878#
$MTYP1  001557             899#
$MTYP2  001563             907#
$MTYP3  001567             910#
$MTYP4  001573             913#
$MXCNT  016652            3459    3471#
$N     = 000000            534#   2704#
$NULL   001454             848#   2868    2897
$NWTST= 000000            1349#   1420#   1511#   1568#   1606#   1649#   1693#   1777#   1844#   1981#   2064#   2147#   2230#
                          2307#   2355#   2403#   2474#   2546#   2601#   2656#
$OCNT   015006            3200*   3229*   3242#
$OMODE  015010            3195*   3199*   3204    3207*   3218*   3244#
$OVER   016632            3420    3423    3439    3447    3457    3466#
$PASS   001534             831#   1167*   2763*   3453    3472
```

```
$PASTM  002144          1117#
$PWRDN  015012          1144    3248#
$QUES   001522           869#   2897    3113
$RDCHR= ****** U        3509
$RDDEC= ****** U        3509
$RDLIN= ****** U        3509
$RDOCT= ****** U        3509
$REGAD  001460           852#
$REG0   001462           854#   2992*   2997
$REG1   001464           855#   2991*   2998
$REG2   001466           856#   2990*   2999
$REG3   001470           857#   2989*   3000
$REG4   001472           858#   2988*   3001
$REG5   001474           859#   2987*   3002
$R2A  = ****** U        3509
$SAVRE= ****** U        3509
$SCOPE  016320          1138    3405#
$SETUP= 000017          1120#   1137    1138    1140    1142    1144    1146    1147    1149    3080    3105    3112    3406
$STUP = 177777          1120#
$SVLAD  016576          3431    3460#
$SVPC = 002136          1092#   1097
$SWR  = 177400           525#    866     867     868    1146    1147    1149    1150    1351    1422    1513    1570    1608
                        1651    1695    1779    1846    1983    2066    2149    2232    2309    2357    2405    2476    2548
                        2603    2658    3071    3072    3073    3074    3075    3083    3090    3102    3105    3113    3397
                        3398    3399    3400    3401    3422    3434    3436    3437    3440    3441    3442    3449    3450
                        3451    3463    3466    3471
$SWREG  001550           889#   1170
$SWRMK= 000000          3401    3402    3438
$TESTN  001532           880#   3461*
$TIMES  001512           866#   1146*   3449*   3456    3459*   3471
$TKB    001446           845#   2787    2910    2918    3421
$TKS    001444           844#   2785    2908    3419
$TMP0   001476           860#
$TMP1   001500           861#   1383*   1389*   1476*   1482*   2567*   2579*   2583*   2622*   2634*   2638*   2677*   2689*
                        2693*   3560*   3572    3582*   3584*   3589    3599*   3601*
$TMP2   001502           862#   3559*   3561*   3562*   3563*   3564    3572*   3576*   3589*   3593*
$TMP3   001504           863#   3573*   3577*   3580*   3590*   3594*   3597*
$TMP4   001506           864#
$TMP5   001510           865#   1799*   1803*   1804    1862*   1866*   1867    1886*   1890*   1891    1928*   1932*   1933
$TN   = 000025           536#   1349    1351#   1420    1422#   1511    1513#   1568    1570#   1606    1608#   1649    1651#
                        1693    1695#   1777    1779#   1844    1846#   1981    1983#   2064    2066#   2147    2149#   2230
                        2232#   2307    2309#   2355    2357#   2403    2405#   2474    2476#   2546    2548#   2601    2603#
                        2656    2658#
$TPB    001452           847#   2886*   2897    2918*
$TPFLG  001457           851#   2835    2897
$TPS    001450           846#   2884    2897    2916
$TRAP   016654          1142    3480#
$TRAP2  016676          3491#   3502
$TRP  = 000015          3495#   3504#   3505#   3506#   3507#   3509    3510#   3511#   3512#   3513#   3514#   3515#   3516#
                        3517#
$TRPAD  016710          3485    3502#
$TSTM   002142          1116#
$TSTNM  001402           824#   1180*   1335    1338    2758*   3082    3113    3119    3396    3417    3438    3460*   3461
                        3466    3472
$TYPBN= ****** U        3507
$TYPDS= ****** U        3507
```

```
$TYPE   013072            536    2835#   3495    3503
$TYPEC  013304           2865    2872    2879    2884#   2885
$TYPEX  013352           2890    2892    2895#
$TYPOC  014610           3198#   3504
$TYPON  014624           3197    3200#   3506
$TYPOS  014564           3193#   3505
$UNIT   001540            883#
$UNITM  002146           1118#
$USWR   001552            890#   1205
$VECT1  001576            915#
$VECT2  001600            916#
$XTSTR  016402           3413    3425#
$OFILL  015007           3194*   3198*   3208    3243#
$40CAT= ****** U         3092    3422
      = 017416            536#    570#    683#    686#    691#    746#    747#    821#    872    1078#   1092    1093#   1095#
                         1097#   1103    1104#   1106#   1108#   1135    1149    1150    1373    1399    1407    1423    1446
                         1457    1468    1492    1500    1543    1549    1578    1587    1615    1658    1673    1681    1723
                         1730    1733    1744    1755    1802    1826    1833    1865    1868    1889    1931    1944    1951
                         1955    1960    1999    2006    2021    2029    2040    2045    2082    2089    2104    2112    2123
                         2128    2165    2172    2187    2195    2206    2211    2248    2255    2270    2278    2289    2294
                         2339    2343    2387    2391    2426    2432    2441    2447    2457    2461    2497    2503    2512
                         2518    2528    2532    2570    2574    2589    2625    2629    2644    2680    2684    2699    2897
                         3113    3167#   3259    3275    3386#   3388#   3390#   3471    3472
.BEGIN  003310           1197    1208    1211    1326#   2775
.CONVR  013764           3007#   3515
.EOP    012354           2710#
.INSTE  013462           2921#   3511
.INSTR  013354           2900#   3510
.INST1  013374           2904#   2913    2923
.MSG    013376           2787*   2788*   2789    2900*   2905#
.PARAM  013472           2928#   3512
.PFAIL  015012           1174    3249#   3270
.RESOS  013732           2997#   3514
.SAVOS  013672           2983#   3513
.SCOPE  016320           3409#
.SCOP1  013050           2813#   3509
.SETFL  014142           3053#   3064    3516
.START  002152            687    1129#   1181    2724
.$ASTA= ****** U          536
.$X   = 002136           1103#   1108
```

```
COMMEN    680#
ENDCOM    680#
ERROR     574#   1374   1408   1409   1447   1459   1469   1501   1502   1544   1550   1580   1588   1636   1659
         1674   1675   1724   1731   1734   1746   1756   1806   1827   1834   1900   1941   1945   1952   1956
         1961   2000   2007   2022   2030   2041   2046   2083   2090   2105   2113   2124   2129   2166   2173
         2188   2196   2207   2212   2249   2256   2271   2279   2290   2295   2340   2344   2388   2392   2458
         2462   2529   2533   2571   2575   2590   2626   2630   2645   2681   2685   2700
ESCAPE    680#
GETPRI    680#
GETSWR    680#
MULT      680#
NEWTST    680#   1349   1420   1511   1568   1606   1649   1693   1777   1844   1981   2064   2147   2230   2307
         2355   2403   2474   2546   2601   2656
POP       536    680#
PRGEND    558#   2703
PRGFRT    537#    559
PUSH      536    680#
PUSSYF    555#   2422   2436   2493   2507
REPORT    525#   680#
RSETUP    542#   1352   1425   2548   2603   2658
SC        534#   3406
SCOPE     575#   1350   1421   1512   1569   1607   1650   1694   1778   1845   1982   2065   2148   2231   2308
         2356   2404   2475   2547   2602   2657   2710
SC1       534#   3468
SETPRI    680#
SETTRA    525#   3495#  3504   3505   3506   3509   3510   3511   3512   3513   3514   3515   3516
SETUP     680#   1129
SKIP      680#
SLASH     680#
SPACE     680#
STARS     536    680#   817    872    875    1090   1100   1102   1109   1349   1420   1511   1568   1606   1649
         1693   1777   1844   1981   2064   2147   2230   2307   2355   2403   2474   2546   2601   2656   2820
         3067   3123   3170   3393   3474
SWRSU     680#   1151#
TRMTRP    3495#
TSETUP    543#   1514   1696   1846   2309   2357   2407   2478
TYPBIN    680#
TYPDEC    680#
TYPNAM    680#
TYPNUM    680#
TYPOCS    680#
TYPOCT    680#   3135   3159
TYPTXT    680#
$BEGIN    534#   1323
$BINAR    554#   2300   2348   2396   2467
$BUFFE    536#   3382
$CABLE    556#
$CATCH    528#    570
$CK       535#   2776
$CLRVE    534#   1203
$CONVR    536#   3004
$DNA      552#
$EOP      535#   2703
$GETFL    534#   1242   1307   1311   1315   1319
$GETPA    534#   1221   1232   1252   1278   1330
$GETSY    534#   1289
```

```
$HEADE    525#    559
$INSTR    536#   2897
$ISOB     550#
$MATCH    547#   1968   2051   2134   2217
$MRR      541#
$MRRW     539#
$MRW      540#
$MSG      536#   3286
$NEED     557#   3113
$PARAM    536#   2925
$PFAIL    536#   3245
$POKE     545#   1527   1530   1709   1712   1992   1996   2010   2075   2079   2093   2158   2162   2176   2241
         2245    2259   2324   2328   2372   2376
$POKER    546#   1362   1435   2561   2616   2671
$RCNET    557#
$RECAC    545#
$REG      536#   2980
$RESET    538#   1352   1354   1425   1427   1514   1516   1571   1608   1651   1696   1698   1782   1784   1846
         1848    1983   2043   2066   2126   2149   2209   2232   2292   2309   2311   2357   2359   2407   2409
         2478    2480   2548   2550   2603   2605   2658   2660
$RXACT    544#
$SCOPE    534#   3391
$SCOP1    535#   2810
$SETFL    534#   3047
$SETVE    529#    680
$START    534#   1121
$STRIP    551#
$SYMBO    525#    748    941
$SYNCR    547#   2557   2612   2667
$TRPAR    553#
$TSTNO    534#   1349   1420   1511   1568   1606   1649   1693   1777   1844   1981   2064   2147   2230   2307
         2355    2403   2474   2546   2601   2656
$UNIBU    537#
$VARIA    532#    690
$WORDF    548#
$WORDO    547#   2538   2593   2648
$WORDP    549#
$$CMRE    815#    854    855    856    857    858    859
$$CMTM    815#    860    861    862    863    864    865
$$ESCA    680#
$$NEWT    680#   1349   1420   1511   1568   1606   1649   1693   1777   1844   1981   2064   2147   2230   2307
         2355    2403   2474   2546   2601   2656
$$SET    3495#   3504   3505   3506   3509   3510   3511   3512   3513   3514   3515   3516
$$SETM   1167#
$$SKIP    680#
.EQUAT    525#    570
.HEADE    525#
.SETUP    525#   1120
.$ACT1    525#   1088
.$APTB    525#    873#
.$APTH    525#   1098
.$APTY    525#    536
.$CATC    525#
.$CMTA    525#    815
.$EOP     525#
.$ERRO    525#   3065
```

```
. $ERRT    525#   3121
. $POWE    525#
. $SCOP    525#   3391
. $TRAP    525#   3472
. $TYPE    525#   2818
. $TYPO    525#   3168


  ABS.   017416     000


 ERRORS DETECTED:  0

 DZDUVB,DZDUVB/SOL/CRF=DZDUV1/EQ:RUNF,DZDUV2,DZDUVB
 RUN-TIME:  21 12 1 SECONDS
 RUN-TIME RATIO:  270/34=7.8
 CORE USED:  30K  (59 PAGES)
```