DUV/LSI-11

DUV11 OFF-LINE RCVR TIME

MD-11-DZDUS-A

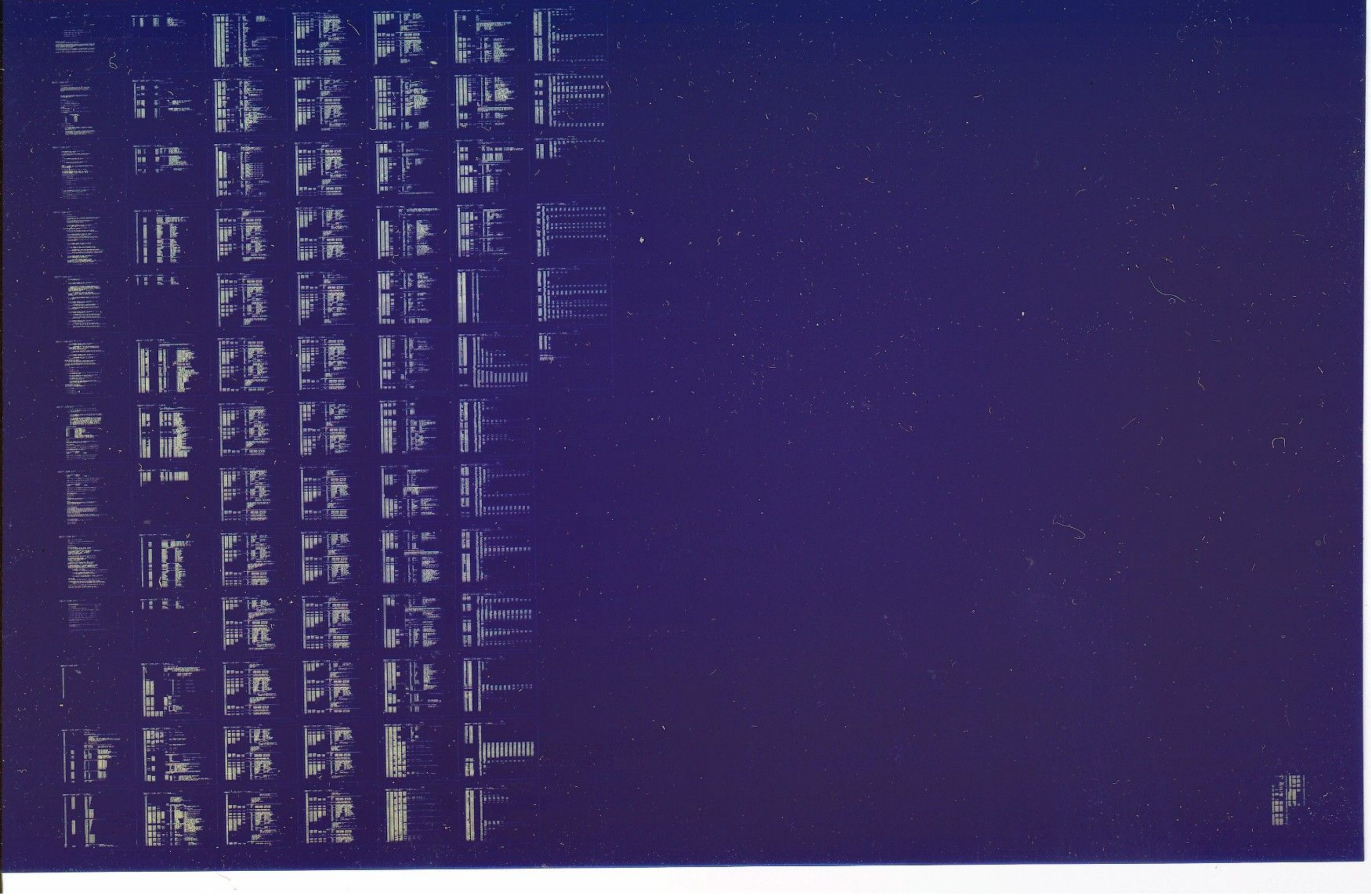
EP-DZDUS-A-DL-A

FICHE 1 OF 1

APR 197777

digital

MADE IN USA



.REM *

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

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

PRODUCT NAME: DUV11 OFFLINE RECEIVER TIMING TESTS

RELEASE DATE: FEB. 1977

MAINTAINER : DIAGNOSTICS

*
.REM *

*

.REM *

GENERAL DESCRIPTION

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

*
.REM *

1. THE DUV11 OFFLINE RECEIVER TIMING TESTS VERIFY THAT THE
RECEIVER LOGIC AND ASSOCIATED ERROR FLAGS ASSERT
AT THE PROPPER TIME

*
.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.

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

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

SW00=1

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

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

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

4.2 STARTING ADDRESS

THE STARTING ADDRESS FOR ALL TESTS IS 000200

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

4.3 PROGRAM AND/OR OPERATOR ACTION

4.3.1 INITIAL PROGRAM START

4.3.1.1 LOAD PROGRAM INTO MEMORY WITH ABSOLUTE LOADER

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

4.3.1.3 TYPE 200G.

4.3.1.4 PROGRAM WILL START.

*
.REM *

4.3.1.5 THE PROGRAM WILL TYPE "DUV11 DZDUS-A TAPE C" (ONCE ONLY)

*
.REM *

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

4.3.2 PROGRAM RESTART WITH ALL SWITCHES DOWN

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

4.3.3 PROGRAM RESTART WITH SW00=1

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

4.3.3.2 TYPE 200G.

4.3.3.3 PROGRAM WILL START.

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

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

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

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

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

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

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

4.3.3.9 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A ‹CARRIAGE RETURN›

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

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

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

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

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

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

4.3.3.11.2 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL

REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED
BY A <CARRIAGE RETURN>

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4.3.4.2 TYPE 200G.

4.3.4.3 PROGRAM WILL START.

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

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

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

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

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

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

4.3.5.2 TYPE 200G.

4.3.5.3 PROGRAM WILL START.

4.3.5.4 THE PROGRAM WILL TYPE "LOCK ON SELECTED TEST ? (Y OR N)-"

AND WAIT FOR AN INPUT FRCM 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 SWD1 =1 IT
WILL PERFORM AS IN SECTION 4.3.4 ALLOWING ONE TO FREEZE
ON A SELECTED TEST RATHER THAN DEFAULTING TO TEST 1

5. OPERATING PROCEDURE

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

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

6. ERRORS

6.1 ERROR HALTS (UNDER LSI ALL HALT ERRORS RETURN CONTROL TO O.D.T.)

THERE ARE FOUR DISTINCT ERROR TYPEOUTS

6.1.1 PC+2 = ERROR PC

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

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

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

6.1.2 PC +2 = REGISTER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING DEVICE REGISTER

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.3 PC +2 = RECEIVER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

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

WHERE YYYYYY IS THE EXPECTED DATA CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL DATA CONTENTS OF THAT REGISTER

6.1.4 PC +2 = TRANSMITTER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

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

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.5 ERROR DESCRIPTIONS
SEE LISTINGS FOR DETAILS OF ERRORS

6.2 ERROR RECOVERY

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

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

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

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

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

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

END OF PASS TAPE Y
16XXXX = DEVICE

WHERE Y IS THE TAPE LOADED

WHERE 16XXXX IS THE DEVICE'S BASE REGISTER ADDRESS

TO INHIBIT THIS TYPEOUT - TURN TELETYPE OFF

7. RESTRICTIONS

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

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

7.2 DISQUALIFYING DEVICES WHEN RUNNING MULTIPLE DEVICES

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

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

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

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

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

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

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

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

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

7.4 TO USE THE "XOR" TESTER ,THE BRANCH AROUND THE "XOR"
CODE MUST BE PATCHED TO A "NOP". (SEE LISTINGS FOR DETAILS)

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

VECTOR ADDRESS- DURIV: 770

ARE YOU RUNNING MULTIPLE DEVICES ?- NO MULTD: 0

LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- LASTADD: 0

OF SYNC CHARS SELECTED - 2 SYNCNO: 377

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

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

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

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

9. PROGRAM DESCRIPTION

*
.REM *

9.1 THIS PROGRAM PERFOFMS THE OFFLINE RECEIVER TIMING TESTING
OF THE DEVICE
SEE LISTING FOR DETAILS

*
.REM *

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

11. LISTINGS

*

```
                000001                  $TN=1
```

```
                                .ENABLE ABS

                                ;DUV11 DZDUS-A TAPE C
                                ;COPYRIGHT 1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

                                        ;STARTING PROCEDURE
                                        ;TYPE 200G
                                        ;PROGRAM WILL TYPE "DUV11 DZDUS-A TAPE C "
                                        ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
                                        ;AT THE END OF A PASS, PROGRAM WILL TYPE "END OF PASS TAPE C"
                                        ;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
```

```
                                BASIC DEFINITIONS

                        SW13=   20000
                        SW12=   10000
                        SW11=   4000
                        SW10=   2000
                        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
```

```
   668          000014                  TBITVEC=14              ;;"T" BIT
   669          000014                  TRTVEC= 14              ;;TRACE TRAP
   670          000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
   671          000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
   672          000024                  PWRVEC= 24              ;;POWER FAIL
   673          000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
   674          000034                  TRAPVEC=34              ;;"TRAP" TRAP
   675          000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
   676          000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
   677          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
```

BASIC DEFINITIONS

```
804          000040                  DNAINTE=BIT5                ;DNA INTR ENAB
805          000020                  SEND=BIT4                   ;SEND
806          000010                  HDXEN=BIT3                  ;HDX/FDX
807          000001                  BREAK=BIT0                  ;BREAK
808                                          ;TXCSR WRD DEFINITIONS
809          000000                  USER=0                      ;USER MODE
810          004000                  MINT=4000                   ;MAINT INT MODE
811          010000                  MEXT=10000                  ;MAINT EXT MODE
812          014000                  SYSTST=14000                ;SYSTEM TEST MODE
```

```
                                .SBTTL  COMMON TAGS

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

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

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

                                ;;*****************************************************************
                                .EVEN
                                $MAIL:
                                $MSGTY: .WORD   AMSGTY  ;;APT MAILBOX
                                $FATAL: .WORD   AFATAL  ;;MESSAGE TYPE CODE
                                $TESTN: .WORD   ATESTN  ;;FATAL ERROR NUMBER
                                $PASS:  .WORD   APASS   ;;TEST NUMBER
                                $DEVCT: .WORD   ADEVCT  ;;PASS COUNT
                                $UNIT:  .WORD   AUNIT   ;;DEVICE COUNT
                                $MSGAD: .WORD   AMSGAD  ;;I/O UNIT NUMBER
                                $MSGLG: .WORD   AMSGLG  ;;MESSAGE ADDRESS
                                $ETABLE:                ;;MESSAGE LENGTH
                                $ENV:   .BYTE   AENV    ;;APT ENVIRONMENT TABLE
                                $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT BYTE
                                $SWREG: .WORD   ASWREG  ;;ENVIRONMENT MODE BITS
                                $USWR:  .WORD   AUSWR   ;;APT SWITCH REGISTER
                                $CPUOP: .WORD   ACPUOP  ;;USER SWITCHES
                                                        ;;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
001556      000     $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
001557      000     $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
                                ;*                     MEM.TYPE BYTE   --  (HIGH BYTE)
                                ;*                             900 NSEC CORE=001
                                ;*                             300 NSEC BIPOLAR=002
                                ;*                             500 NSEC MOS=003
001560  000000  $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                                ;*                      MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
001562      000     $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
001563      000     $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
001564  000000  $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
001566      000     $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
001567      000     $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
001570  000000  $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
001572      000     $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
001573      000     $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
001574  000000  $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
001576  000000  $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001600  000000  $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001602  000000  $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001604  000000  $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
001606  000000  $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
001610  000000  $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
001612  000000  $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
001614  000000  $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
001616  000000  $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
001620  000000  $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
001622  000000  $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
001624  000000  $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
```

APT MAILBOX-ETABLE

```
925  001626  000000                  $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
926  001630  000000                  $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
927  001632  000000                  $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
928  001634  000000                  $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
929  001636  000000                  $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
930  001640  000000                  $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
931  001642  000000                  $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
932  001644  000000                  $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
933  001646  000000                  $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
934  001650  000000                  $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
935
936
937  001652                          $ETEND:
938
939
940
```

```
                                ;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
```

APT MAILBOX-ETABLE

```
  997          000040                  DNAINTE=BIT5                ;DNA INTR ENAB
  998          000020                  SEND=BIT4                   ;SEND
  999          000010                  HDXEN=BIT3                  ;HDX/FDX
 1000          000001                  BREAK=BIT0                  ;BREAK
 1001                                          ;TXCSR WRD DEFINITIONS
 1002          000000                  USER=0                      ;USER MODE
 1003          004000                  MINT=4000                   ;MAINT INT MODE
 1004          010000                  MEXT=10000                  ;MAINT EXT MODE
 1005          014000                  SYSTST=14000                ;SYSTEM TEST MODE
```

ERROR POINTER TABLE

```
1006                                    .SBTTL  ERROR POINTER TABLE
1007
1008                                    ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
1009                                    ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
1010                                    ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
1011                                    ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
1012                                    ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
1013
1014                                    ;*      EM              ;;POINTS TO THE ERROR MESSAGE
1015                                    ;*      DH              ;;POINTS TO THE DATA HEADER
1016                                    ;*      DT              ;;POINTS TO THE DATA
1017                                    ;*      DF              ;;POINTS TO THE DATA FORMAT
1018
1019
1020  001652                            $ERRTB:
1021                                            ;ERROR TABLE
1022  001652  001762                                    EM1     ;ERROR 1        REGISTER ERROR
1023  001654  002067                                    DH1
1024  001656  002116                                    DT1
1025  001660  002132                                    DF1
1026  001662  002022                                    EM2     ;ERROR 2        RECEIVER ERROR
1027  001664  002067                                    DH1
1028  001666  002116                                    DT1
1029  001670  002132                                    DF1
1030  001672  002043                                    EM3     ;ERROR 3        TRANSMITTER ERROR
1031  001674  002067                                    DH1
1032  001676  002116                                    DT1
1033  001700  002132                                    DF1
1034  001702  001746                                    EM4     ;ERROR 4        BIT ERROR (GENERAL)
1035  001704  000000                                    0
1036  001706  002126                                    DT4
1037  001710  002132                                    DF1
1038
1039                                            ;DEFAULT DU ADDRESSES
1040  001712  160010                    RXCSR:  160010
1041  001714  160011                    HRXCSR: 160011
1042  001716  160012                    RXDBUF: 160012
1043  001720  160013                    HRXDBUF:        160013
1044  001722  160012                    PARCSR: 160012
1045  001724  160013                    HPARCSR:        160013
1046  001726  160014                    TXCSR:  160014
1047  001730  160015                    HTXCSR: 160015
1048  001732  160016                    TXDBUF: 160016
1049  001734  160017                    HTXDBUF:        160017
1050                                            ;DEFAULT DU VECTORS
1051  001736  000770                    DURIV:  770     ;REC INTR VECTOR
1052  001740  000772                    DURIS:  772     ;REC INTR STATUS
1053  001742  000774                    DUTIV:  774     ;XMIT INTR VECTOR
1054  001744  000776                    DUTIS:  776     ;XMIT INTR STATUS
1055                                            ;ERROR MESSAGES
1056  001746  020040  051105  047522    EM4:    .ASCIZ  /  ERROR PC /
1057  001754  020122  041520  000040
1058  001762  020040  047503  050115    EM1:    .ASCIZ /  COMPARISON ERROR ON REGISTERS/
1059  001770  051101  051511  047117
1060  001776  042440  051122  051117
1061  002004  047440  020116  042522
```

```
                                        ERROR POINTER TABLE

1062  002012  044507  052123  051105
1063  002020  000123
1064  002022  020040  042522  042503  EM2:    .ASCIZ  /  RECEIVER ERROR/
1065  002030  053111  051105  042440
1066  002036  051122  051117     000
1067  002043     040  052040  040522  EM3:    .ASCIZ  /  TRANSMITTER ERROR/
1068  002050  051516  044515  052124
1069  002056  051105  042440  051122
1070  002064  051117     000
1071                                          ;DATA HEADERS FOR ERROR MESSAGES
1072  002067     105  051122  041520  DH1:    .ASCIZ  /ERRPC  WANTED  ACTUAL/
1073  002074  020040  040527  052116
1074  002102  042105  020040  041501
1075  002110  052524  046101     000
1076          002116                  .EVEN
1077                                          ;DATA TABLES FOR ERROR MESSAGES
1078  002116  001416  001130  001132  DT1:    .WORD   $ERRPC,HLD0,HLD1,0
1079  002124  000000
1080
1081  002126  001416  000000          DT4:    .WORD   $ERRPC,0
1082
1083  002132     000     000     000  DF1:    .BYTE   0,0,0,0
1084  002135     000
1085                                  .EVEN
1086                                  .SBTTL  ACT11 HOOKS
1087
1088                                  ;;******************************************************************
1089                                  ;HOOKS REQUIRED BY ACT11
1090          002136                          $SVPC=.                 ;SAVE PC
1091          000046                          .=46
1092  000046  012646                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1093          000052                          .=52
1094  000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
1095          002136                          .=$SVPC                 ;; RESTORE PC
1096                                  .SBTTL  APT PARAMETER BLOCK
1097
1098                                  ;;******************************************************************
1099                                  ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1100                                  ;;******************************************************************
1101          002136                          .$X=.   ;;SAVE CURRENT LOCATION
1102          000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1103  000024  000200                          200     ;;FOR APT START UP
1104          000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
1105  000044  002136                          $APTHDR ;;POINT TO APT HEADER BLOCK
1106          002136                          .=.$X   ;;RESET LOCATION COUNTER
1107                                  ;;******************************************************************
1108                                  ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1109                                  ;INTERFACE SPEC.
1110
1111  002136                          $APTHD:
1112  002136  000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1113  002140  001526                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
1114  002142  000010                  $TSTM:  .WORD   10      ;;RUN TIM OF LONGEST TEST
1115  002144  000010                  $PASTM: .WORD   10      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1116  002146  000000                  $UNITM: .WORD           ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1117  002150  000052                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

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

002152                          .START:
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
002152  012706  001400                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
002156  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
002160  022706  001440                  CMP     #SWR,R6 ;;DONE?
002164  001374                          BNE     .-6             ;;LOOP BACK IF NO
002166  012706  001100                  MOV     #@#STACK,SP             ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
002172  012737  016276  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
002200  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
002206  012737  014166  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
002214  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
002222  012737  016614  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
002230  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
002236  012737  014770  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
002244  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
002252  005067  177234                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
002256  005067  177232                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
002262  112767  000001  177125          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
002270  012767  002270  177110          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
002276  012767  002276  177104          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
002304  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
002310  012737  002344  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
002316  012767  177570  177114          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
002324  012767  177570  177110          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
002332  022777  177777  177100          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
002340  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
002342  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
002344  012716  002352          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
002350  000002                          RTI
002352  012767  000176  177060  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
002360  012767  000174  177054          MOV     #DISPREG,DISPLAY
002366  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

002372  005067  177136                  CLR     $PASS           ;;CLEAR PASS COUNT
002376  132767  000200  177143          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
002404  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
002406  012767  001550  177024          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
002414                          67$:
002414  012706  001100                  MOV     #STACK,SP               ;SET STACK
002420  106427  000340                  MTPS    #340                    ;LOCK INTERRUPTS
002424  012737  014770  000024          MOV     #.PFAIL,@#24            ;SET UP POWER FAIL VECTOR
002432  105067  176535                  CLRB    STFLG                   ;CLEAR START FLAG
```

```
1174  002436  005067  176450                    CLR     PASCNT                  ;CLEAR PASS COUNT
1175  002442  105067  176735                    CLRB    $ERFLG                  ;CLEAR ERROR FLAG
1176  002446  005067  176740                    CLR     $ERTTL                  ;CLEAR ERROR COUNT
1177  002452  005067  176740                    CLR     $ERRPC                  ;CLEAR LAST ERROR POINTER
1178  002456  012767  000001  176716            MOV     #1,$TSTNM               ;SET UP FOR TEST 1
1179  002464  012767  002152  176412            MOV     #.START,RETURN          ;SET UP FOR POWER FAIL BEFORE
1180                                                                            ;TESTING STARTS
1181  002472  013746  000006                    MOV     @#6,-(SP)
1182  002476  013746  000004                    MOV     @#4,-(SP)
1183  002502  012737  002516  000004            MOV     #1$,@#4
1184  002510  005777  176724                    TST     @SWR
1185  002514  000407                            BR      2$
1186  002516  012767  000176  176714  1$:       MOV     #SWREG,SWR
1187  002524  012767  000174  176710            MOV     #DISPREG,DISPLAY
1188  002532  022626                            CMP     (SP)+,(SP)+
1189  002534  012637  000004          2$:       MOV     (SP)+,@#4
1190  002540  012637  000006                    MOV     (SP)+,@#6
1191  002544  022767  000176  176666            CMP     #SWREG,SWR
1192  002552  001007                            BNE     3$
1193  002554  005737  000042                    TST     @#42                    ;CHECK FOR CHAIN
1194  002560  001402                            BEQ     33$
1195  002562  000167  000522                    JMP     .BEGIN
1196  002566  004767  010162          33$:      JSR     PC,CNTLU
1197  002572  105767  176374          3$:       TSTB    INIFLG                  ;HAS INITIALIZATION BEEN PERFORMED
1198  002576  001004                            BNE     ONCE
1199  002600  104401  015130                    TYPE    ,MTITLE                 ;TYPE TITLE MESSAGE
1200  002604  105167  176362                    COMB    INIFLG                  ;IF NOT SET FLAG AND DO
1201  002610  105767  176732          ONCE:     TSTB    $ENV            ;APT CONTROL?
1202  002614  001410                            BEQ     11$             ;BR IF NO
1203  002616  032767  000001  176726            BIT     #1,$USWR        ;EXTENAL JUMPER ON?
1204  002624  001002                            BNE     12$             ;NO
1205  002626  105067  176321                    CLRB    JMRBY           ;CLEAR FLAG
1206  002632  000167  000452          12$:      JMP     .BEGIN          ;GO DO IT
1207  002636  032777  000001  176574  11$:      BIT     #SW00,@SWR      ;RESELECT VECTOR & CONTROL REG?
1208  002644  001002                            BNE     1$
1209  002646  000167  000436                    JMP     .BEGIN
1210  002652  012700  000300          1$:       MOV     #300,R0 ;RESTORE VECTOR AREA TO TRAPCATCHER
1211  002656  012701  000302                    MOV     #302,R1 ;START AT LOCATION 300
1212  002662  012702  000004                    MOV     #4,R2
1213  002666  010110                  2$:       MOV     R1,(R0)
1214  002670  005011                            CLR     (R1)
1215  002672  060200                            ADD     R2,R0
1216  002674  060201                            ADD     R2,R1
1217  002676  022701  001000                    CMP     #1000,R1        ;END AT LOCATION 776
1218  002702  002771                            BLT     2$
1219  002704  104406                            INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1220  002706  015176                            MREGAD  ;MESSAGE
1221  002710  104410                            PARAM   ;CONVERT STRING
1222  002712  160000                            160000  ;LOW LIMIT
1223  002714  167776                            167776  ;HIGH LIMIT
1224  002716  017110                            DUBASE  ;STORE AT THIS LOCATION
1225  002720     001                  .BYTE     1       ;MASK
1226  002721     001                  .BYTE     1       ;HOW MANY TIMES + 2
1227  002722  016767  014162  176226            MOV     DUBASE,KEEPADD  ;SAVE
1228  002730  004767  014022                    JSR     PC,DUADDR
1229  002734  016767  176216  176212            MOV     KEEPADD,BASEADD ;RESTORE FOR ROTATION
```

```
1230  002742  104406                                  INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1231  002744  015163                                  MVECTO  ;MESSAGE
1232  002746  104410                                  PARAM   ;CONVERT STRING
1233  002750  000300                                  300     ;LOW LIMIT
1234  002752  000776                                  776     ;HIGH LIMIT
1235  002754  001736                                  DURIV   ;STORE AT THIS LOCATION
1236  002756     001                          .BYTE    1       ;MASK
1237  002757     004                          .BYTE    4       ;HOW MANY TIMES + 2
1238  002760  016767  176752  176176                  MOV     DURIV,KEEPIV    ;SAVE
1239  002766  016767  176744  176166                  MOV     DURIV,BASEIV    ;SET UP FOR ROTATION
1240  002774  104406                                  INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1241  002776  015226                                  MMULT   ;MESSAGE
1242  003000  104414                                  SETFLG  ;SET FLAG BASED UPON INPUT STRING
1243  003002  001152                                  MULTD   ;THIS FLAG
1244  003004  105767  176142                         TSTB    MULTD   ;ARE THERE MULTIPLE DEVICES
1245                                                           ;ON THE SYSTEM ?
1246  003010  100406                                  BMI     BBB     ;YES,ASK NEXT QUESTION
1247  003012  005067  176150                         CLR     ACTREG
1248  003016  005067  176146                         CLR     ROTADD
1249  003022  000167  000140                         JMP     OUTMUL  ;JUMP AROUND NEXT QUESTION
1250  003026                               BBB:
1251  003026  104406                                  INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1252  003030  015255                                  MLASTD  ;MESSAGE
1253  003032  104410                                  PARAM   ;CONVERT STRING
1254  003034  160000                                  160000  ;LOW LIMIT
1255  003036  167776                                  167776  ;HIGH LIMIT
1256  003040  001160                                  LASTADD ;STORE AT THIS LOCATION
1257  003042     001                          .BYTE    1       ;MASK
1258  003043     001                          .BYTE    1       ;HOW MANY TIMES + 2
1259                                          ;THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
1260  003044  012767  000001  176116  1$:     MOV     #1,ROTADD       ;SET UP POINTER
1261  003052  005067  176110                 CLR     ACTREG  ;CLR ACTIVE REGISTER
1262  003056  056767  176106  176102  2$:     BIS     ROTADD,ACTREG   ;MAKE THIS DEVICE ACTIVE
1263  003064  000241                          CLC
1264  003066  006167  176076                 ROL     ROTADD  ;SET UP POINTER
1265  003072  103421                          BCS     3$      ;ARE YOU OUT OF RANGE ?
1266  003074  062767  000010  176052         ADD     #10,BASEADD     ;SET UP BASE ADDRESS
1267  003102  026767  176052  176044         CMP     LASTADD,BASEADD ;IS THIS THE LAST DEVICE ?
1268  003110  101362                          BHI     2$      ;NO DO IT AGAIN
1269  003112  056767  176052  176046         BIS     ROTADD,ACTREG   ;THIS ASSUMES THAT THERE ARE AT
1270                                                  ;LEAST TWO DEVICES WHEN YOU ANSWER YES TO
1271                                                  ;MULTIPLE DEVICE QUESTION
1272  003120  012767  000001  176042  4$:     MOV     #1,ROTADD       ;SET UP FOR LATER USE IN END OF PASS ROUTINE
1273  003126  016767  176024  176020         MOV     KEEPADD,BASEADD ;DITTO
1274  003134  000414                          BR      OUTMUL  ;CONTINUE QUESTIONS
1275  003136  016767  176014  176010  3$:     MOV     KEEPADD,BASEADD ;RESTORE
1276  003144  104406                          INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1277  003146  015351                          MRANGE  ;MESSAGE
1278  003150  104410                          PARAM   ;CONVERT STRING
1279  003152  160000                          160000  ;LOW LIMIT
1280  003154  167776                          167776  ;HIGH LIMIT
1281  003156  001160                          LASTADD ;STORE AT THIS LOCATION
1282  003160     001                  .BYTE    1       ;MASK
1283  003161     001                  .BYTE    1       ;HOW MANY TIMES + 2
1284  003162  000167  177656                 JMP     1$      ;DO IT AGAIN
1285  003166  012767  000340  013556  OUTMUL: MOV     #340,DUPRT
```

```
1286  003174  004767  013502                        JSR     PC,DULEV
1287                                                ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1288                                                ;BUFFER TO THE CHARACTERS "1" AND "2".
1289                                                ;IF THE CHARACTER IS "1" CLEAR THE FLAG
1290                                                ;IF THE CHARACTER IS "2" SET THE FLAG
1291  003200                                AAA:
1292  003200  104406                               INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1293  003202  015567                               MSYNC   ;MESSAGE
1294  003204  122767  000061  012716       3$:     CMPB    #'1,INBUF       ;IS IT "1" ?
1295  003212  001003                               BNE     1$
1296  003214  105067  175726                       CLRB    SYNCNO  ;000
1297  003220  000412                               BR      4$
1298  003222  122767  000062  012700       1$:     CMPB    #'2,INBUF       ;IS IT "2" ?
1299  003230  001004                               BNE     2$
1300  003232  112767  :77777  175706               MOVB    #-1,SYNCNO      ;377
1301  003240  000402                               BR      4$
1302  003242  104407                       2$:     INSTER  ;RETRY
1303  003244  000757                               BR      3$
1304  003246  000240                       4$:     NOP
1305  003250  104406                               INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1306  003252  015635                               MWIRE6  ;MESSAGE
1307  003254  104414                               SETFLG  ;SET FLAG BASED UPON INPUT STRING
1308  003256  001147                               SEXMIT  ;THIS FLAG
1309  003260  104406                               INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1310  003262  015706                               MWIRE5  ;MESSAGE
1311  003264  104414                               SETFLG  ;SET FLAG BASED UPON INPUT STRING
1312  003266  001150                               SEREC   ;THIS FLAG
1313  003270  104406                               INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1314  003272  015756                               MWIRE4  ;MESSAGE
1315  003274  104414                               SETFLG  ;SET FLAG BASED UPON INPUT STRING
1316  003276  001151                               OPTCLR  ;THIS FLAG
1317  003300  104406                               INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1318  003302  016035                               MEXTJ   ;MESSAGE
1319  003304  104414                               SETFLG  ;SET FLAG BASED UPON INPUT STRING
1320  003306  001153                               JMRBY   ;THIS FLAG
1321
1322                                                ;TEST START AND RESTART
1323
1324  003310  012706  001100           .BEGIN:     MOV     #STACK,SP               ;SET UP STACK
1325  003314  106427  000340                       MTPS    #340                    ;LOCK OUT INTERRUPTS
1326  003320  032777  000002  176112               BIT     #SW01,@SWR              ;IF SW01=1, GET STARTING PC
1327  003326  001406                               BEQ     3$
1328  003330  104406                               INSTR   ;OUTPUT MESSAGE & GET INPUT STRING
1329  003332  015521                               MTSTPC  ;MESSAGE
1330  003334  104410                               PARAM   ;CONVERT STRING
1331  003336  003362                               TST1    ;LOW LIMIT
1332                                                        ;HIGH LIMIT
1333                                                        ;STORE AT THIS LOCATION
1334  003340     001                       .BYTE   1       ;MASK
1335  003341     001                       .BYTE   1       ;HOW MANY TIMES + 2
1336  003342  000403                               BR      4$
1337  003344  012767  003362  175532       3$:     MOV     #TST1,RETURN            ;START AT TEST 1
1338  003352  104401  015515               4$:     TYPE    ,MR                     ;TYPE R
1339  003356  000177  175522                       JMP     @RETURN                 ;START TESTING
1340
1341                                                ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
```

```
1342                                                    ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1343                                                    ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1344                                                    ;;(OVRRUN,RXERR)
1345                                                    ;;MODE:SYNEXT
1346                                                    ;;LENGTH:SEVEN
1347                                                    ;;CHAR:177
1348                                                    ;;
1349                                                ;;*********************************************************************
1350  003362  000004                                TST1:   SCOPE
1351  003364  052777  000400  176334                        BIS     #MRESET,@TXCSR  ;MASTER RESET
1352  003372  012777  020000  176322                        MOV     #SYNEXT,@PARCSR ;SET THE MODE
1353  003400  052777  000400  176320                        BIS     #MRESET,@TXCSR  ;MASTER RESET
1354
1355                                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1356  003406  012777  064001  176312                        MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1357
1358                                                ;SET MODE ,# OF BITS,PARITY SENSE &LOAD SYNC REG
1359  003414  012777  024000  176300                        MOV     #SYNEXT!SEVEN!NOPAR!0,@PARCSR
1360  003422  052777  000020  176262                        BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
1361                                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1362  003430  042777  020000  176270                        BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1363  003436  052777  020000  176262                        BIS     #CLK,@TXCSR     ;POKE CLK UP
1364  003444  016703  176246                              MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1365  003450  012700  000177                              MOV     #177,R0 ;EXPECTED
1366  003454  012767  000007  175440                      MOV     #7,SHIFT        ;# OF SHIFTS
1367  003462  012767  000177  176010                      MOV     #177,$TMP1      ;DATA CHAR
1368  003470  004767  013416                              JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1369  003474  105777  176212                              TSTB    @RXCSR  ;RXDONE ?
1370  003500  100401                                      BMI     .+4
1371  003502  104004                                      ERROR   4               ;RXDONE SHOULD BE SET
1372  003504  017701  176206                              MOV     @RXDBUF,R1      ;ACTUAL
1373  003510  020001                                      CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1374  003512  001401                                      BEQ     .+4
1375  003514  104002                                      ERROR   2               ;RECEIVED DATA DID NOT MATCH
1376                                                                                ;EXPECTED DATA - CHECK MAINT DATA
1377                                                                                ;OR RECEIVER LOGIC
1378  003516  012767  000007  175376                      MOV     #7,SHIFT        ;# OF SHIFTS
1379  003524  012767  000177  175746                      MOV     #177,$TMP1      ;DATA CHAR
1380  003532  004767  013354                              JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1381                                                ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1382  003536  012767  000007  175356                      MOV     #7,SHIFT        ;# OF SHIFTS
1383  003544  012767  000177  175726                      MOV     #177,$TMP1      ;DATA CHAR
1384  003552  004767  013334                              JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1385  003556  012700  140177                              MOV     #140000!177,R0  ;EXPECTED DATA PLUS
1386                                                                                ;RXERR & OVRRUN
1387  003562  017701  176130                              MOV     @RXDBUF,R1      ;ACTUAL
1388  003566  020001                                      CMP     R0,R1   ;COMPARE EXP VS. ACT
1389  003570  001401                                      BEQ     .+4
1390  003572  104002                                      ERROR   2               ;SPECIFICALLY LOOK AT RXERR &
1391                                                                                ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1392
1393                                                    ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1394                                                    ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1395                                                    ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1396                                                    ;;(OVRRUN,RXERR)
1397                                                    ;;MODE:SYNEXT
```

```
1398                                                    ;;LENGTH:SEVEN
1399                                                    ;;CHAR:0
1400                                                    ;;
1401                                            ;;**********************************************************************
1402  003574  000004                          TST2:   SCOPE
1403  003576  052777  000400  176122                    BIS     #MRESET,@TXCSR  ;MASTER RESET
1404  003604  012777  020000  176110                    MOV     #SYNEXT,@PARCSR ;SET THE MODE
1405  003612  052777  000400  176106                    BIS     #MRESET,@TXCSR  ;MASTER RESET
1406
1407                                            ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1408  003620  012777  064001  176100                    MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1409
1410                                            ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1411  003626  012777  024000  176066                    MOV     #SYNEXT!SEVEN!NOPAR!0,@PARCSR
1412  003634  052777  000020  176050                    BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
1413                                            ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1414  003642  042777  020000  176056                    BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1415  003650  052777  020000  176050                    BIS     #CLK,@TXCSR     ;POKE CLK UP
1416  003656  016703  176034                          MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1417  003662  012700  000000                          MOV     #0,R0   ;EXPECTED
1418  003666  012767  000007  175226                  MOV     #7,SHIFT        ;# OF SHIFTS
1419  003674  012767  000000  175576                  MOV     #0,$TMP1        ;DATA CHAR
1420  003702  004767  013204                          JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1421  003706  105777  176000                          TSTB    @RXCSR  ;RXDONE ?
1422  003712  100401                                  BMI     .+4
1423  003714  104004                                  ERROR   4       ;RXDONE SHOULD BE SET
1424  003716  017701  175774                          MOV     @RXDBUF,R1      ;ACTUAL
1425  003722  020001                                  CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1426  003724  001401                                  BEQ     .+4
1427  003726  104002                                  ERROR   2       ;RECEIVED DATA DID NOT MATCH
1428                                                                  ;EXPECTED DATA - CHECK MAINT DATA
1429                                                                  ;OR RECEIVER LOGIC
1430  003730  012767  000007  175164                  MOV     #7,SHIFT        ;# OF SHIFTS
1431  003736  012767  000000  175534                  MOV     #0,$TMP1        ;DATA CHAR
1432  003744  004767  013142                          JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1433                                          ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1434  003750  012767  000007  175144                  MOV     #7,SHIFT        ;# OF SHIFTS
1435  003756  012767  000000  175514                  MOV     #0,$TMP1        ;DATA CHAR
1436  003764  004767  013122                          JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1437  003770  012700  140000                          MOV     #140000!0,R0    ;EXPECTED DATA PLUS
1438                                                                  ;RXERR & OVRRUN
1439  003774  017701  175716                          MOV     @RXDBUF,R1      ;ACTUAL
1440  004000  020001                                  CMP     R0,R1   ;COMPARE EXP VS. ACT
1441  004002  001401                                  BEQ     .+4
1442  004004  104002                                  ERROR   2       ;SPECIFICALLY LOOK AT RXERR &
1443                                                                  ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1444
1445                                                    ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1446                                                    ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1447                                                    ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1448                                                    ;;(OVRRUN,RXERR)
1449                                                    ;;MODE:SYNEXT
1450                                                    ;;LENGTH:EIGHT
1451                                                    ;;CHAR:125
1452                                                    ;;
1453                                            ;;**********************************************************************
```

```
                                        INITIALIZE THE COMMON TAGS

 1454  004006  000004                   TST3:   SCOPE
 1455  004010  052777  000400  175710           BIS     #MRESET,@TXCSR  ;MASTER RESET
 1456  004016  012777  020000  175676           MOV     #SYNEXT,@PARCSR ;SET THE MODE
 1457  004024  052777  000400  175674           BIS     #MRESET,@TXCSR  ;MASTER RESET
 1458
 1459                                   ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1460  004032  012777  064001  175666           MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1461
 1462                                   ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1463  004040  012777  026000  175654           MOV     #SYNEXT!EIGHT!NOPAR!0,@PARCSR
 1464  004046  052777  000020  175636           BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
 1465                                   ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1466  004054  042777  020000  175644           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1467  004062  052777  020000  175636           BIS     #CLK,@TXCSR     ;POKE CLK UP
 1468  004070  016703  175622                   MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
 1469  004074  012700  000125                   MOV     #125,R0 ;EXPECTED
 1470  004100  012767  000010  175014           MOV     #8.,$SHIFT      ;# OF SHIFTS
 1471  004106  012767  000125  175364           MOV     #125,$TMP1      ;DATA CHAR
 1472  004114  004767  012772                   JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1473  004120  105777  175566                   TSTB    @RXCSR  ;RXDONE ?
 1474  004124  100401                           BMI     .+4
 1475  004126  104004                           ERROR   4       ;RXDONE SHOULD BE SET
 1476  004130  017701  175562                   MOV     @RXDBUF,R1      ;ACTUAL
 1477  004134  020001                           CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
 1478  004136  001401                           BEQ     .+4
 1479  004140  104002                           ERROR   2       ;RECEIVED DATA DID NOT MATCH
 1480                                                           ;EXPECTED DATA - CHECK MAINT DATA
 1481                                                           ;OR RECEIVER LOGIC
 1482  004142  012767  000010  174752           MOV     #8.,SHIFT       ;# OF SHIFTS
 1483  004150  012767  000125  175322           MOV     #125,$TMP1      ;DATA CHAR
 1484  004156  004767  012730                   JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1485                                   ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
 1486  004162  012767  000010  174732           MOV     #8.,SHIFT       ;# OF SHIFTS
 1487  004170  012767  000125  175302           MOV     #125,$TMP1      ;DATA CHAR
 1488  004176  004767  012710                   JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1489  004202  012700  140125                   MOV     #140000!125,R0  ;EXPECTED DATA PLUS
 1490                                                           ;RXERR & OVRRUN
 1491  004206  017701  175504                   MOV     @RXDBUF,R1      ;ACTUAL
 1492  004212  020001                           CMP     R0,R1   ;COMPARE EXP VS. ACT
 1493  004214  001401                           BEQ     .+4
 1494  004216  104002                           ERROR   2       ;SPECIFICALLY LOOK AT RXERR &
 1495                                                           ;OVRRUN BITS...THEY BOTH SHOULD BE SET
 1496
 1497                                           ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
 1498                                           ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
 1499                                           ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
 1500                                           ;;(OVRRUN,RXERR)
 1501                                           ;;MODE:SYNEXT
 1502                                           ;;LENGTH:EIGHT
 1503                                           ;;CHAR:252
 1504                                           ;;
 1505                                   ;;*****************************************************************
 1506  004220  000004                   TST4:   SCOPE
 1507  004222  052777  000400  175476           BIS     #MRESET,@TXCSR  ;MASTER RESET
 1508  004230  012777  020000  175464           MOV     #SYNEXT,@PARCSR ;SET THE MODE
 1509  004236  052777  000400  175462           BIS     #MRESET,@TXCSR  ;MASTER RESET
```

```
1510
1511                                        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1512  004244  012777  064001  175454              MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1513
1514                                        ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1515  004252  012777  026000  175442              MOV     #SYNEXT!EIGHT!NOPAR!0,@PARCSR
1516  004260  052777  000020  175424              BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
1517                                        ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1518  004266  042777  020000  175432              BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1519  004274  052777  020000  175424              BIS     #CLK,@TXCSR     ;POKE CLK UP
1520  004302  016703  175410                    MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1521  004306  012700  000252                    MOV     #252,R0 ;EXPECTED
1522  004312  012767  000010  174602           MOV     #8.,SHIFT       ;# OF SHIFTS
1523  004320  012767  000252  175152           MOV     #252,STMP1      ;DATA CHAR
1524  004326  004767  012560                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1525  004332  105777  175354                    TSTB    @RXCSR  ;RXDONE ?
1526  004336  100401                            BMI     .+4
1527  004340  104004                            ERROR   4       ;RXDONE SHOULD BE SET
1528  004342  017701  175350                    MOV     @RXDBUF,R1      ;ACTUAL
1529  004346  020001                            CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1530  004350  001401                            BEQ     .+4
1531  004352  104002                            ERROR   2       ;RECEIVED DATA DID NOT MATCH
1532                                                            ;EXPECTED DATA - CHECK MAINT DATA
1533                                                            ;OR RECEIVER LOGIC
1534  004354  012767  000010  174540           MOV     #8.,SHIFT       ;# OF SHIFTS
1535  004362  012767  000252  175110           MOV     #252,STMP1      ;DATA CHAR
1536  004370  004767  012516                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1537                                        ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1538  004374  012767  000010  174520           MOV     #8.,SHIFT       ;# OF SHIFTS
1539  004402  012767  000252  175070           MOV     #252,STMP1      ;DATA CHAR
1540  004410  004767  012476                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1541  004414  012700  140252                    MOV     #140000!252,R0  ;EXPECTED DATA PLUS
1542                                                            ;RXERR & OVRRUN
1543  004420  017701  175272                    MOV     @RXDBUF,R1      ;ACTUAL
1544  004424  020001                            CMP     R0,R1   ;COMPARE EXP VS. ACT
1545  004426  001401                            BEQ     .+4
1546  004430  104002                            ERROR   2       ;SPECIFICALLY LOOK AT RXERR &
1547                                                            ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1548
1549                                                    ;;THIS TEST VERIFYS WORC LENGTH SELECT OF THE
1550                                                    ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1551                                                    ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1552                                                    ;;(OVRRUN,RXERR)
1553                                                    ;;MODE:SYNEXT
1554                                                    ;;LENGTH:EIGHT
1555                                                    ;;CHAR:377
1556                                                    ;;
1557                                        ;;**************************************************************
1558  004432  000004                        TST5:   SCOPE
1559  004434  052777  000400  175264              BIS     #MRESET,@TXCSR  ;MASTER RESET
1560  004442  012777  020000  175252              MOV     #SYNEXT,@PARCSR ;SET THE MODE
1561  004450  052777  000400  175250              BIS     #MRESET,@TXCSR  ;MASTER RESET
1562
1563                                        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1564  004456  012777  064001  175242              MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1565
```

```
DZDUS-A MACY11 27(1006)  03-FEB-77  07:52  PAGE 36
DZDUSA.M11    13-OCT-76 08:39          INITIALIZE THE COMMON TAGS

 1566
 1567  004464  012777  026000  175230          ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
                                                        MOV     #SYNEXT!EIGHT!NOPAR!0,@PARCSR
 1568  004472  052777  000020  175212                  BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
 1569                                           ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1570  004500  042777  020000  175220                  BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1571  004506  052777  020000  175212                  BIS     #CLK,@TXCSR     ;POKE CLK UP
 1572  004514  016703  175176                        MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
 1573  004520  012700  000377                        MOV     #377,R0 ;EXPECTED
 1574  004524  012767  000010  174370                MOV     #8.,SHIFT       ;# OF SHIFTS
 1575  004532  012767  000377  174740                MOV     #377,STMP1      ;DATA CHAR
 1576  004540  004767  012346                        JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1577  004544  105777  175142                        TSTB    @RXCSR  ;RXDONE ?
 1578  004550  100401                                BMI     .+4
 1579  004552  104004                                ERROR   4               ;RXDONE SHOULD BE SET
 1580  004554  017701  175136                        MOV     @RXDBUF,R1      ;ACTUAL
 1581  004560  020001                                CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
 1582  004562  001401                                BEQ     .+4
 1583  004564  104002                                ERROR   2               ;RECEIVED DATA DID NOT MATCH
 1584                                                                          ;EXPECTED DATA - CHECK MAINT DATA
 1585                                                                          ;OR RECEIVER LOGIC
 1586  004566  012767  000010  174326                MOV     #8.,SHIFT       ;# OF SHIFTS
 1587  004574  012767  000377  174676                MOV     #377,STMP1      ;DATA CHAR
 1588  004602  004767  012304                        JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1589                                           ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
 1590  004606  012767  000010  174306                MOV     #8.,SHIFT       ;# OF SHIFTS
 1591  004614  012767  000377  174656                MOV     #377,STMP1      ;DATA CHAR
 1592  004622  004767  012264                        JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1593  004626  012700  140377                        MOV     #140000!377,R0  ;EXPECTED DATA PLUS
 1594                                                                          ;RXERR & OVRRUN
 1595  004632  017701  175060                        MOV     @RXDBUF,R1      ;ACTUAL
 1596  004636  020001                                CMP     R0,R1   ;COMPARE EXP VS. ACT
 1597  004640  001401                                BEQ     .+4
 1598  004642  104002                                ERROR   2               ;SPECIFICALLY LOOK AT RXERR &
 1599                                                                          ;OVRRUN BITS...THEY BOTH SHOULD BE SET
 1600
 1601                                                   ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
 1602                                                   ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
 1603                                                   ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
 1604                                                   ;;(OVRRUN,RXERR)
 1605                                                   ;;MODE:SYNEXT
 1606                                                   ;;LENGTH:EIGHT
 1607                                                   ;;CHAR:0
 1608                                                   ;;
 1609                                           ;;*****************************************************************
 1610  004644  000004                          TST6:   SCOPE
 1611  004646  052777  000400  175052                  BIS     #MRESET,@TXCSR  ;MASTER RESET
 1612  004654  012777  020000  175040                  MOV     #SYNEXT,@PARCSR ;SET THE MODE
 1613  004662  052777  000400  175036                  BIS     #MRESET,@TXCSR  ;MASTER RESET
 1614
 1615                                           ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1616  004670  012777  064001  175030                  MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1617
 1618                                           ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1619  004676  012777  026000  175016                  MOV     #SYNEXT!EIGHT!NOPAR!0,@PARCSR
 1620  004704  052777  000020  175000                  BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
 1621                                           ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
```

```
1622  004712  042777  020000  175006          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1623  004720  052777  020000  175000          BIS     #CLK,@TXCSR     ;POKE CLK UP
1624  004726  016703  174764                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1625  004732  012700  000000                  MOV     #0,R0   ;EXPECTED
1626  004736  012767  000010  174156          MOV     #8.,SHIFT       ;# OF SHIFTS
1627  004744  012767  000000  174526          MOV     #0,$TMP1        ;DATA CHAR
1628  004752  004767  012134                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1629  004756  105777  174730                  TSTB    @RXCSR  ;RXDONE ?
1630  004762  100401                          BMI     .+4
1631  004764  104004                          ERROR   4       ;RXDONE SHOULD BE SET
1632  004766  017701  174724                  MOV     @RXDBUF,R1      ;ACTUAL
1633  004772  020001                          CMP     R0,R1   ;COMPARE EXPECTED VS. ACTUAL
1634  004774  001401                          BEQ     .+4
1635  004776  104002                          ERROR   2       ;RECEIVED DATA DID NOT MATCH
1636                                                          ;EXPECTED DATA - CHECK MAINT DATA
1637                                                          ;OR RECEIVER LOGIC
1638  005000  012767  000010  174114          MOV     #8.,SHIFT       ;# OF SHIFTS
1639  005006  012767  000000  174464          MOV     #0,$TMP1        ;DATA CHAR
1640  005014  004767  012072                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1641                                  ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1642  005020  012767  000010  174074          MOV     #8.,SHIFT       ;# OF SHIFTS
1643  005026  012767  000000  174444          MOV     #0,$TMP1        ;DATA CHAR
1644  005034  004767  012052                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1645  005040  012700  140000                  MOV     #140000!0,R0    ;EXPECTED DATA PLUS
1646                                                          ;RXERR & OVRRUN
1647  005044  017701  174646                  MOV     @RXDBUF,R1      ;ACTUAL
1648  005050  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
1649  005052  001401                          BEQ     .+4
1650  005054  104002                          ERROR   2       ;SPECIFICALLY LOOK AT RXERR &
1651                                                          ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1652
1653                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1654                                          ;;SECTION ,IT USES THE ERROR FLAGS TO DETERMINE
1655                                          ;;THAT IT WAS SELECTED PROPERLY
1656                                          ;;FRAME ERROR (FRMERR,RXERR)
1657                                          ;;MODE:ISOC (ISYMOD)
1658                                          ;;LENGTH:FIVE
1659                                          ;;CHAR: 25
1660                                          ;;
1661                                  ;;**************************************************************
1662  005056  000004                  TST7:   SCOPE
1663  005060  052777  000400  174640          BIS     #MRESET,@TXCSR  ;MASTER RESET
1664  005066  012777  000000  174626          MOV     #ISYMOD,@PARCSR ;SET THE MODE
1665  005074  052777  000400  174624          BIS     #MRESET,@TXCSR  ;MASTER RESET
1666
1667                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1668  005102  012777  064001  174616          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1669
1670                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1671  005110  012777  000000  174604          MOV     #ISYMOD!FIVE!NOPAR!0,@PARCSR
1672  005116  052777  000020  174566          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1673                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1674  005124  042777  020000  174574          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1675  005132  052777  020000  174566          BIS     #CLK,@TXCSR     ;POKE CLK UP
1676                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1677  005140  042777  020000  174560          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
```

```
                                        INITIALIZE THE COMMON TAGS

1678  005146  052777  020000  174552            BIS     #CLK,@TXCSR     ;POKE CLK UP
1679  005154  012767  000007  173740            MOV     #7,SHIFT        ;# OF SHIFTS
1680  005162  012767  000052  174310            MOV     #52,$TMP1       ;DATA CHAR
1681                                    ;NOTE: THE ABOVE CHARACTER IS MISSING STOP BIT
1682  005170  004767  011716                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1683  005174  016703  174516                    MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
1684  005200  012700  120025                    MOV     #RXERR!FRMERR!25,R0     ;EXPECTED
1685  005204  017701  174506                    MOV     @RXDBUF,R1      ;ACTUAL
1686  005210  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
1687  005212  001401                            BEQ     .+4
1688  005214  104000                            ERROR   ;FRAME ERROR    4 & RX ERROR SHOULD BE SET
1689                                                    ;IF LOWER BYTE DOES NOT MATCH  IT
1690                                                    ;PROBABLY IS A LENGTH SELECT PROBLEM
1691
1692                                            ;;THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1693                                            ;;SECTION ,IT USES THE ERROR FLAGS TO DETERMINE
1694                                            ;;THAT IT WAS SELECTED PROPERLY
1695                                            ;;FRAME ERROR (FRMERR,RXERR)
1696                                            ;;MODE:ISOC (ISYMOD)
1697                                            ;;LENGTH:SIX
1698                                            ;;CHAR: 25
1699                                            ;;
1700                                    ;;**************************************************************************
1701  005216  000004                    TST10:  SCOPE
1702  005220  052777  000400  174500            BIS     #MRESET,@TXCSR  ;MASTER RESET
1703  005226  012777  000000  174466            MOV     #ISYMOD,@PARCSR ;SET THE MODE
1704  005234  052777  000400  174464            BIS     #MRESET,@TXCSR  ;MASTER RESET
1705
1706                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1707  005242  012777  064001  174456            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1708
1709                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1710  005250  012777  002000  174444            MOV     #ISYMOD!SIX!NOPAR!0,@PARCSR
1711  005256  052777  000020  174426            BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1712                                            ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1713  005264  042777  020000  174434            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1714  005272  052777  020000  174426            BIS     #CLK,@TXCSR     ;POKE CLK UP
1715                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1716  005300  042777  020000  174420            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1717  005306  052777  020000  174412            BIS     #CLK,@TXCSR     ;POKE CLK UP
1718  005314  012767  000010  173600            MOV     #8.,SHIFT       ;# OF SHIFTS
1719  005322  012767  000052  174150            MOV     #52,$TMP1       ;DATA CHAR
1720                                    ;NOTE: THE ABOVE CHARACTER IS MISSING STOP BIT
1721  005330  004767  011556                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1722  005334  016703  174356                    MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
1723  005340  012700  120025                    MOV     #RXERR!FRMERR!25,R0     ;EXPECTED
1724  005344  017701  174346                    MOV     @RXDBUF,R1      ;ACTUAL
1725  005350  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
1726  005352  001401                            BEQ     .+4
1727  005354  104000                            ERROR   ;FRAME ERROR    4 & RX ERROR SHOULD BE SET
1728                                                    ;IF LOWER BYTE DOES NOT MATCH  IT
1729                                                    ;PROBABLY IS A LENGTH SELECT PROBLEM
1730
1731                                            ;;THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1732                                            ;;SECTION ,IT USES THE ERROR FLAGS TO DETERMINE
1733                                            ;;THAT IT WAS SELECTED PROPERLY
```

```
 1734                                          ;;FRAME ERROR (FRMERR,RXERR)
 1735                                          ;;MODE:ISOC (ISYMOD)
 1736                                          ;;LENGTH:SEVEN
 1737                                          ;;CHAR: 125
 1738                                          ;;
 1739                                  ;;**************************************************************************
 1740  005356  000004                 TST11:  SCOPE
 1741  005360  052777  000400  174340         BIS     #MRESET,@TXCSR  ;MASTER RESET
 1742  005366  012777  000000  174326         MOV     #ISYMOD,@PARCSR ;SET THE MODE
 1743  005374  052777  000400  174324         BIS     #MRESET,@TXCSR  ;MASTER RESET
 1744
 1745                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1746  005402  012777  064001  174316         MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1747
 1748                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1749  005410  012777  004000  174304         MOV     #ISYMOD!SEVEN!NOPAR!0,@PARCSR
 1750  005416  052777  000020  174266         BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
 1751                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
 1752  005424  042777  020000  174274         BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1753  005432  052777  020000  174266         BIS     #CLK,@TXCSR     ;POKE CLK UP
 1754                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1755  005440  042777  020000  174260         BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1756  005446  052777  020000  174252         BIS     #CLK,@TXCSR     ;POKE CLK UP
 1757  005454  012767  000011  173440         MOV     #9.,SHIFT       ;# OF SHIFTS
 1758  005462  012767  000252  174010         MOV     #252,STMP1      ;DATA CHAR
 1759                                  ;NOTE: THE ABOVE CHARACTER IS MISSING STOP BIT
 1760  005470  004767  011416                 JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1761  005474  016703  174216                 MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
 1762  005500  012700  120125                 MOV     #RXERR!FRMERR!125,R0    ;EXPECTED
 1763  005504  017701  174206                 MOV     @RXDBUF,R1      ;ACTUAL
 1764  005510  020001                         CMP     R0,R1   ;COMPARE EXP VS ACT
 1765  005512  001401                         BEQ     .+4
 1766  005514  104000                         ERROR   ;FRAME ERROR    4 & RX ERROR SHOULD BE SET
 1767                                                  ;IF LOWER BYTE DOES NOT MATCH  IT
 1768                                                  ;PROBABLY IS A LENGTH SELECT PROBLEM
 1769
 1770                                          ;;THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
 1771                                          ;;SECTION ,IT USES THE ERROR FLAGS TO DETERMINE
 1772                                          ;;THAT IT WAS SELECTED PROPERLY
 1773                                          ;;FRAME ERROR (FRMERR,RXERR)
 1774                                          ;;MODE:ISOC (ISYMOD)
 1775                                          ;;LENGTH:EIGHT
 1776                                          ;;CHAR: 125
 1777                                          ;;
 1778                                  ;;**************************************************************************
 1779  005516  000004                 TST12:  SCOPE
 1780  005520  052777  000400  174200         BIS     #MRESET,@TXCSR  ;MASTER RESET
 1781  005526  012777  000000  174166         MOV     #ISYMOD,@PARCSR ;SET THE MODE
 1782  005534  052777  000400  174164         BIS     #MRESET,@TXCSR  ;MASTER RESET
 1783
 1784                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1785  005542  012777  064001  174156         MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1786
 1787                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1788  005550  012777  006000  174144         MOV     #ISYMOD!EIGHT!NOPAR!0,@PARCSR
 1789  005556  052777  000020  174126         BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
```

```
 1790                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
 1791  005564  042777  020000  174134         BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1792  005572  052777  020000  174126         BIS     #CLK,@TXCSR     ;POKE CLK UP
 1793                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1794  005600  042777  020000  174120         BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1795  005606  052777  020000  174112         BIS     #CLK,@TXCSR     ;POKE CLK UP
 1796  005614  012767  000012  173300         MOV     #10.,SHIFT      ;# OF SHIFTS
 1797  005622  012767  000252  173650         MOV     #252,$TMP1      ;DATA CHAR
 1798                                  ;NOTE: THE ABOVE CHARACTER IS MISSING STOP BIT
 1799  005630  004767  011256                 JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1800  005634  016703  174056                 MOV     RXDBUF,R3       ;FOR ERROR MESSAGE
 1801  005640  012700  120125                 MOV     #RXERR!FRMERR!125,R0    ;EXPECTED
 1802  005644  017701  174046                 MOV     @RXDBUF,R1      ;ACTUAL
 1803  005650  020001                         CMP     R0,R1   ;COMPARE EXP VS ACT
 1804  005652  001401                         BEQ     .+4
 1805  005654  104000                         ERROR   ;FRAME ERROR    4 & RX ERROR SHOULD BE SET
 1806                                                  ;IF LOWER BYTE DOES NOT MATCH  IT
 1807                                                  ;PROBABLY IS A LENGTH SELECT PROBLEM
 1808
 1809                                          ;;THIS TEST VERIFYS EVEPAR PARITY SENSE
 1810                                          ;;OF THE RECEIVER
 1811                                          ;;MODE:ISOC (ISYMOD)
 1812                                          ;;PARITY:EVEPAR
 1813                                          ;;LENGTH:FIVE PLUS PARITY
 1814                                          ;;CHAR: 25
 1815                                          ;;
 1816                                  ;;**************************************************************
 1817  005656  000004                 TST13:  SCOPE
 1818  005660  052777  000400  174040         BIS     #MRESET,@TXCSR  ;MASTER RESET
 1819  005666  012777  000000  174026         MOV     #ISYMOD,@PARCSR ;SET THE MODE
 1820  005674  052777  000400  174024         BIS     #MRESET,@TXCSR  ;MASTER RESET
 1821
 1822                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1823  005702  012777  064001  174016         MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1824
 1825                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1826  005710  012777  001400  174004         MOV     #ISYMOD!FIVE!EVEPAR!0,@PARCSR
 1827  005716  052777  000020  173766         BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
 1828                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
 1829  005724  042777  020000  173774         BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1830  005732  052777  020000  173766         BIS     #CLK,@TXCSR     ;POKE CLK UP
 1831                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1832  005740  042777  020000  173760         BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1833  005746  052777  020000  173752         BIS     #CLK,@TXCSR     ;POKE CLK UP
 1834  005754  016703  173736                 MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
 1835  005760  012700  110025                 MOV     #RXERR!PARER!25,R0      ;EXPECTED
 1836  005764  012767  000010  173130         MOV     #8.,SHIFT       ;# OF SHIFTS
 1837  005772  012767  000252  173500         MOV     #252,$TMP1      ;DATA CHAR
 1838  006000  004767  011106                 JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1839  006004  105777  173702                 TSTB    @RXCSR  ;RXDONE ?
 1840  006010  100401                         BMI     .+4
 1841  006012  104004                         ERROR   4               ;RXDONE SHOULD BE ASSERTED
 1842  006014  017701  173676                 MOV     @RXDBUF,R1      ;ACTUAL
 1843  006020  020001                         CMP     R0,R1   ;COMPARE EXP VS. ACT
 1844  006022  001401                         BEQ     .+4
 1845  006024  104000                         ERROR   ;PARITY ERROR   4 &RXERR SHOULD BE SET
```

```
                                                        ;NOTE THAT THE PARITY BIT SHOULD
                                                        ;SHOW UP IN THE DATA
                                                        ;IE. BIT FIVE FOR FIVE LEVEL CODE

                                                ;;THIS TEST VERIFYS EVEPAR PARITY SENSE
                                                ;;OF THE RECEIVER
                                                ;;MODE:ISOC (ISYMOD)
                                                ;;PARITY:EVEPAR
                                                ;;LENGTH:SIX PLUS PARITY
                                                ;;CHAR: 25
                                                ;;
                                        ;;**************************************************************
006026  000004                          TST14:  SCOPE
006030  052777  000400  173670                  BIS     #MRESET,@TXCSR  ;MASTER RESET
006036  012777  000000  173656                  MOV     #ISYMOD,@PARCSR ;SET THE MODE
006044  052777  000400  173654                  BIS     #MRESET,@TXCSR  ;MASTER RESET

                                        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
006052  012777  064001  173646                  MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                        ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
006060  012777  003400  173634                  MOV     #ISYMOD!SIX!EVEPAR!0,@PARCSR
006066  052777  000020  173616                  BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
                                                ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
006074  042777  020000  173624                  BIC     #CLK,@TXCSR     ;POKE CLK DOWN
006102  052777  020000  173616                  BIS     #CLK,@TXCSR     ;POKE CLK UP

                                        ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
006110  042777  020000  173610                  BIC     #CLK,@TXCSR     ;POKE CLK DOWN
006116  052777  020000  173602                  BIS     #CLK,@TXCSR     ;POKE CLK UP
006124  016703  173566                          MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
006130  012700  110025                          MOV     #RXERR!PARER!25,R0      ;EXPECTED
006134  012767  000011  172760                  MOV     #9.,SHIFT       ;# OF SHIFTS
006142  012767  000452  173330                  MOV     #452,$TMP1      ;DATA CHAR
006150  004767  010736                          JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
006154  105777  173532                          TSTB    @RXCSR  ;RXDONE ?
006160  100401                                  BMI     .+4
006162  104004                                  ERROR   4               ;RXDONE SHOULD BE ASSERTED
006164  017701  173526                          MOV     @RXDBUF,R1      ;ACTUAL
006170  020001                                  CMP     R0,R1   ;COMPARE EXP VS. ACT
006172  001401                                  BEQ     .+4
006174  104000                                  ERROR   ;PARITY ERROR   4 &RXERR SHOULD BE SET
                                                        ;NOTE THAT THE PARITY BIT SHOULD
                                                        ;SHOW UP IN THE DATA
                                                        ;IE. BIT SIX FOR SIX LEVEL CODE

                                                ;;THIS TEST VERIFYS EVEPAR PARITY SENSE
                                                ;;OF THE RECEIVER
                                                ;;MODE:ISOC (ISYMOD)
                                                ;;PARITY:EVEPAR
                                                ;;LENGTH:SEVEN PLUS PARITY
                                                ;;CHAR: 325
                                                ;;
                                        ;;**************************************************************
006176  000004                          TST15:  SCOPE
006200  052777  000400  173520                  BIS     #MRESET,@TXCSR  ;MASTER RESET
006206  012777  000000  173506                  MOV     #ISYMOD,@PARCSR ;SET THE MODE
```

```
                                        INITIALIZE THE COMMON TAGS

1902  006214  052777  000400  173504          BIS     #MRESET,@TXCSR  ;MASTER RESET
1903
1904                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1905  006222  012777  064001  173476          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1906
1907                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1908  006230  012777  005400  173464          MOV     #ISYMOD!SEVEN!EVEPAR!0,@PARCSR
1909  006236  052777  000020  173446          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1910                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1911  006244  042777  020000  173454          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1912  006252  052777  020000  173446          BIS     #CLK,@TXCSR     ;POKE CLK UP
1913                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1914  006260  042777  020000  173440          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1915  006266  052777  020000  173432          BIS     #CLK,@TXCSR     ;POKE CLK UP
1916  006274  016703  173416                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1917  006300  012700  110325                  MOV     #RXERR!PARER!325,R0     ;EXPECTED
1918  006304  012767  000012  172610          MOV     #10.,SHIFT      ;# OF SHIFTS
1919  006312  012767  001652  173160          MOV     #1652,$TMP1     ;DATA CHAR
1920  006320  004767  010566                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
1921  006324  105777  173362                  TSTB    @RXCSR  ;RXDONE ?
1922  006330  100401                          BMI     .+4
1923  006332  104004                          ERROR   4               ;RXDONE SHOULD BE ASSERTED
1924  006334  017701  173356                  MOV     @RXDBUF,R1      ;ACTUAL
1925  006340  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
1926  006342  001401                          BEQ     .+4
1927  006344  104000                          ERROR   ;PARITY ERROR   4 &RXERR SHOULD BE SET
                                                      ;NOTE THAT THE PARITY BIT SHOULD
1929                                                  ;SHOW UP IN THE DATA
1930                                                  ;IE. BIT SEVEN FOR SEVEN LEVEL CODE
1931
1932                                          ;;THIS TEST VERIFYS EVEPAR PARITY SENSE
1933                                          ;;OF THE RECEIVER
1934                                          ;;MODE:ISOC (ISYMOD)
1935                                          ;;PARITY:EVEPAR
1936                                          ;;LENGTH:EIGHT PLUS PARITY
1937                                          ;;CHAR: 125
1938                                          ;;
1939                                  ;;*****************************************************************
1940  006346  000004                  TST16:  SCOPE
1941  006350  052777  000400  173350          BIS     #MRESET,@TXCSR  ;MASTER RESET
1942  006356  012777  000000  173336          MOV     #ISYMOD,@PARCSR ;SET THE MODE
1943  006364  052777  000400  173334          BIS     #MRESET,@TXCSR  ;MASTER RESET
1944
1945                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1946  006372  012777  064001  173326          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1947
1948                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1949  006400  012777  007400  173314          MOV     #ISYMOD!EIGHT!EVEPAR!0,@PARCSR
1950  006406  052777  000020  173276          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1951                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1952  006414  042777  020000  173304          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1953  006422  052777  020000  173276          BIS     #CLK,@TXCSR     ;POKE CLK UP
1954                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1955  006430  042777  020000  173270          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1956  006436  052777  020000  173262          BIS     #CLK,@TXCSR     ;POKE CLK UP
1957  006444  016703  173246                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
```

```
 1958  006450  012700  110125                  MOV     #RXERR!PARER!125,R0     ;EXPECTED
 1959  006454  012767  000013  172440          MOV     #11.,SHIFT      ;# OF SHIFTS
 1960  006462  012767  003252  173010          MOV     #3252,$TMP1     ;DATA CHAR
 1961  006470  004767  010416                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 1962  006474  105777  173212                  TSTB    @RXCSR  ;RXDONE ?
 1963  006500  100401                          BMI     .+4
 1964  006502  104004                          ERROR   4               ;RXDONE SHOULD BE ASSERTED
 1965  006504  017701  173206                  MOV     @RXDBUF,R1      ;ACTUAL
 1966  006510  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
 1967  006512  001401                          BEQ     .+4
 1968  006514  104000                          ERROR   ;PARITY ERROR   4 &RXERR SHOULD BE SET
 1969
 1970                                          ;;THIS TEST VERIFYS ODDPAR PARITY SENSE
 1971                                          ;;OF THE RECEIVER
 1972                                          ;;MODE:ISOC (ISYMOD)
 1973                                          ;;PARITY:ODDPAR
 1974                                          ;;LENGTH:FIVE PLUS PARITY
 1975                                          ;;CHAR: 65
 1976                                          ;;
 1977                                  ;;**************************************************************
 1978  006516  000004                  TST17:  SCOPE
 1979  006520  052777  000400  173200          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1980  006526  012777  000000  173166          MOV     #ISYMOD,@PARCSR ;SET THE MODE
 1981  006534  052777  000400  173164          BIS     #MRESET,@TXCSR  ;MASTER RESET
 1982
 1983                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 1984  006542  012777  064001  173156          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 1985
 1986                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 1987  006550  012777  001000  173144          MOV     #ISYMOD!FIVE!ODDPAR!0,@PARCSR
 1988  006556  052777  000020  173126          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
 1989                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
 1990  006564  042777  020000  173134          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1991  006572  052777  020000  173126          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1992                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 1993  006600  042777  020000  173120          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 1994  006606  052777  020000  173112          BIS     #CLK,@TXCSR     ;POKE CLK UP
 1995  006614  016703  173076                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
 1996  006620  012700  110065                  MOV     #RXERR!PARER!65,R0      ;EXPECTED
 1997  006624  012767  000010  172270          MOV     #8.,SHIFT       ;# OF SHIFTS
 1998  006632  012767  000352  172640          MOV     #352,$TMP1      ;DATA CHAR
 1999  006640  004767  010246                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 2000  006644  105777  173042                  TSTB    @RXCSR  ;RXDONE ?
 2001  006650  100401                          BMI     .+4
 2002  006652  104004                          ERROR   4               ;RXDONE SHOULD BE ASSERTED
 2003  006654  017701  173036                  MOV     @RXDBUF,R1      ;ACTUAL
 2004  006660  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
 2005  006662  001401                          BEQ     .+4
 2006  006664  104000                          ERROR   ;PARITY ERROR   4 &RXERR SHOULD BE SET
 2007                                                  ;NOTE THAT THE PARITY BIT SHOULD
 2008                                                  ;SHOW UP IN THE DATA
 2009                                                  ;IE. BIT FIVE FOR FIVE LEVEL CODE
 2010
 2011                                          ;;THIS TEST VERIFYS ODDPAR PARITY SENSE
 2012                                          ;;OF THE RECEIVER
 2013                                          ;;MODE:ISOC (ISYMOD)
```

```
                                                ;;PARITY:ODDPAR
                                                ;;LENGTH:SIX PLUS PARITY
                                                ;;CHAR: 125
                                                ;;
                                        ;;**************************************************************************
006666  000004                          TST20:  SCOPE
006670  052777  000400  173030                  BIS     #MRESET,@TXCSR  ;MASTER RESET
006676  012777  000000  173016                  MOV     #ISYMOD,@PARCSR ;SET THE MODE
006704  052777  000400  173014                  BIS     #MRESET,@TXCSR  ;MASTER RESET

                                        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
006712  012777  064001  173006                  MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                        ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
006720  012777  003000  172774                  MOV     #ISYMOD!SIX!ODDPAR!0,@PARCSR
006726  052777  000020  172756                  BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
                                                ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
006734  042777  020000  172764                  BIC     #CLK,@TXCSR     ;POKE CLK DOWN
006742  052777  020000  172756                  BIS     #CLK,@TXCSR     ;POKE CLK UP

                                        ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
006750  042777  020000  172750                  BIC     #CLK,@TXCSR     ;POKE CLK DOWN
006756  052777  020000  172742                  BIS     #CLK,@TXCSR     ;POKE CLK UP
006764  016703  172726                          MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
006770  012700  110125                          MOV     #RXERR!PARER!125,R0     ;EXPECTED
006774  012767  000011  172120                  MOV     #9.,SHIFT       ;# OF SHIFTS
007002  012767  000652  172470                  MOV     #652,STMP1      ;DATA CHAR
007010  004767  010076                          JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
007014  105777  172672                          TSTB    @RXCSR  ;RXDONE ?
007020  100401                                  BMI     .+4
007022  104004                                  ERROR   4               ;RXDONE SHOULD BE ASSERTED
007024  017701  172666                          MOV     @RXDBUF,R1      ;ACTUAL
007030  020001                                  CMP     R0,R1   ;COMPARE EXP VS. ACT
007032  001401                                  BEQ     .+4
007034  104000                                  ERROR   ;PARITY ERROR   4 &RXERR SHOULD BE SET
                                                        ;NOTE THAT THE PARITY BIT SHOULD
                                                        ;SHOW UP IN THE DATA
                                                        ;IE. BIT SIX FOR SIX LEVEL CODE

                                                ;;THIS TEST VERIFYS ODDPAR PARITY SENSE
                                                ;;OF THE RECEIVER
                                                ;;MODE:ISOC (ISYMOD)
                                                ;;PARITY:ODDPAR
                                                ;;LENGTH:SEVEN PLUS PARITY
                                                ;;CHAR: 125
                                                ;;
                                        ;;**************************************************************************
007036  000004                          TST21:  SCOPE
007040  052777  000400  172660                  BIS     #MRESET,@TXCSR  ;MASTER RESET
007046  012777  000000  172646                  MOV     #ISYMOD,@PARCSR ;SET THE MODE
007054  052777  000400  172644                  BIS     #MRESET,@TXCSR  ;MASTER RESET

                                        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
007062  012777  064001  172636                  MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                        ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
007070  012777  005000  172624                  MOV     #ISYMOD!SEVEN!ODDPAR!0,@PARCSR
```

```
                                        INITIALIZE THE COMMON TAGS

2070  007076  052777  000020  172606          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2071                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2072  007104  042777  020000  172614          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2073  007112  052777  020000  172606          BIS     #CLK,@TXCSR     ;POKE CLK UP
2074                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2075  007120  042777  020000  172600          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2076  007126  052777  020000  172572          BIS     #CLK,@TXCSR     ;POKE CLK UP
2077  007134  016703  172556                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2078  007140  012700  110125                  MOV     #RXERR!PARER!125,R0     ;EXPECTED
2079  007144  012767  000012  171750          MOV     #10.,SHIFT      ;# OF SHIFTS
2080  007152  012767  001252  172320          MOV     #1252,$TMP1     ;DATA CHAR
2081  007160  004767  007726                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2082  007164  105777  172522                  TSTB    @RXCSR  ;RXDONE ?
2083  007170  100401                          BMI     .+4
2084  007172  104004                          ERROR   4               ;RXDONE SHOULD BE ASSERTED
2085  007174  017701  172516                  MOV     @RXDBUF,R1      ;ACTUAL
2086  007200  020001                          CMP     R0,R1   ;COMPARE EXP VS. ACT
2087  007202  001401                          BEQ     .+4
2088  007204  104000                          ERROR   ;PARITY ERROR   4 &RXERR SHOULD BE SET
2089                                                  ;NOTE THAT THE PARITY BIT SHOULD
2090                                                  ;SHOW UP IN THE DATA
2091                                                  ;IE. BIT SEVEN FOR SEVEN LEVEL CODE
2092
2093                                          ;;THIS TEST VERIFYS ODDPAR PARITY SENSE
2094                                          ;;OF THE RECEIVER
2095                                          ;;MODE:ISOC (ISYMOD)
2096                                          ;;PARITY:ODDPAR
2097                                          ;;LENGTH:EIGHT PLUS PARITY
2098                                          ;;CHAR: 125
2099                                          ;;
2100                                  ;;**************************************************************************
2101  007206  000004                  TST22:  SCOPE
2102  007210  052777  000400  172510          BIS     #MRESET,@TXCSR  ;MASTER RESET
2103  007216  012777  000000  172476          MOV     #ISYMOD,@PARCSR ;SET THE MODE
2104  007224  052777  000400  172474          BIS     #MRESET,@TXCSR  ;MASTER RESET
2105
2106                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2107  007232  012777  064001  172466          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2108
2109                                  ;SET MODE  ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2110  007240  012777  007000  172454          MOV     #ISYMOD!EIGHT!ODDPAR!0,@PARCSR
2111  007246  052777  000020  172436          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2112                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2113  007254  042777  020000  172444          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2114  007262  052777  020000  172436          BIS     #CLK,@TXCSR     ;POKE CLK UP
2115                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2116  007270  042777  020000  172430          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2117  007276  052777  020000  172422          BIS     #CLK,@TXCSR     ;POKE CLK UP
2118  007304  016703  172406                  MOV     RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
2119  007310  012700  110125                  MOV     #RXERR!PARER!125,R0     ;EXPECTED
2120  007314  012767  000013  171600          MOV     #11.,SHIFT      ;# OF SHIFTS
2121  007322  012767  002252  172150          MOV     #2252,$TMP1     ;DATA CHAR
2122  007330  004767  007556                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2123  007334  105777  172352                  TSTB    @RXCSR  ;RXDONE ?
2124  007340  100401                          BMI     .+4
2125  007342  104004                          ERROR   4               ;RXDONE SHOULD BE ASSERTED
```

```
                                        INITIALIZE THE COMMON TAGS

  2126  007344  017701  172346                MOV     @RXDBUF,R1      ;ACTUAL
  2127  007350  020001                        CMP     R0,R1   ;COMPARE EXP VS. ACT
  2128  007352  001401                        BEQ     .+4
  2129  007354  104000                        ERROR   ;PARITY ERROR   4 &RXERR SHOULD BE SET
  2130
  2131                                        ;;THIS TEST PERFORMS BINARY DATA CHECK ON THE
  2132                                        ;;RECEIVER
  2133                                        ;;LENGTH:EIGHT PLUS PARITY
  2134                                        ;;MODE:ISYMOD
  2135                                        ;;PARITY:EVEPAR
  2136                                        ;;
  2137                                    ;;**********************************************************************
  2138  007356  000004                    TST23:  SCOPE
  2139  007360  052777  000400  172340            BIS     #MRESET,@TXCSR  ;MASTER RESET
  2140  007366  012777  000000  172326            MOV     #ISYMOD,@PARCSR ;SET THE MODE
  2141  007374  052777  000400  172324            BIS     #MRESET,@TXCSR  ;MASTER RESET
  2142
  2143                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
  2144  007402  012777  064001  172315            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
  2145
  2146                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
  2147  007410  012777  007400  172304            MOV     #ISYMOD!EIGHT!EVEPAR!0,@PARCSR
  2148  007416  052777  000020  172266            BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
  2149                                            ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
  2150  007424  042777  020000  172274            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
  2151  007432  052777  020000  172266            BIS     #CLK,@TXCSR     ;POKE CLK UP
  2152                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
  2153  007440  042777  020000  172260            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
  2154  007446  052777  020000  172252            BIS     #CLK,@TXCSR     ;POKE CLK UP
  2155  007454  016703  172236                    MOV     RXDBUF,R3       ;SET UP ERROR MESSAGE
  2156  007460  005004                            CLR     R4      ;DATA CHAR
  2157  007462  010400                    1$:     MOV     R4,R0   ;EXPECTED
  2158  007464  012767  000013  171430            MOV     #11.,SHIFT      ;# OF SHIFTS
  2159  007472  010467  172002                    MOV     R4,$TMP1        ;"TO BE SHIFTED CHARACTER"
  2160  007476  004767  007562                    JSR     PC,EVEN8        ;CALC PARITY
  2161  007502  000241                            CLC
  2162  007504  006167  171770                    ROL     $TMP1   ;GENERATE START BIT
  2163  007510  052767  002000  171762            BIS     #BIT10,$TMP1    ;GENERATE STOP BIT
  2164                                    ;$TMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
  2165  007516  004767  007370                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
  2166  007522  017701  172170                    MOV     @RXDBUF,R1      ;ACTUAL
  2167  007526  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
  2168  007530  001401                            BEQ     .+4
  2169  007532  104002                            ERROR   2       ;DATA CHARS SHOULD MATCH
  2170                                                            ;THERE SHOULD BE NO PARITY ERROR
  2171  007534  005204                            INC     R4      ;UPGRADE NEXT CHAR
  2172  007536  105704                            TSTB    R4      ;LAST CHAR ?
  2173  007540  001350                            BNE     1$
  2174
  2175                                        ;;THIS TEST PERFORMS BINARY DATA CHECK ON THE
  2176                                        ;;RECEIVER
  2177                                        ;;LENGTH:EIGHT PLUS PARITY
  2178                                        ;;MODE:ISYMOD
  2179                                        ;;PARITY:ODDPAR
  2180                                        ;;
  2181                                    ;;**********************************************************************
```

```
                                        INITIALIZE THE COMMON TAGS

 2182  007542  000004                   TST24:  SCOPE
 2183  007544  052777  000400  172154           BIS     #MRESET,@TXCSR  ;MASTER RESET
 2184  007552  012777  000000  172142           MOV     #ISYMOD,@PARCSR ;SET THE MODE
 2185  007560  052777  000400  172140           BIS     #MRESET,@TXCSR  ;MASTER RESET
 2186
 2187                                   ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 2188  007566  012777  064001  172132           MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 2189
 2190                                   ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 2191  007574  012777  007000  172120           MOV     #ISYMOD!EIGHT!ODDPAR!0,@PARCSR
 2192  007602  052777  000020  172102           BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
 2193                                           .POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
 2194  007610  042777  020000  172110           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 2195  007616  052777  020000  172102           BIS     #CLK,@TXCSR     ;POKE CLK UP
 2196                                   ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 2197  007624  042777  020000  172074           BIC     #CLK,@TXCSR     ;POKE CLK DOWN
 2198  007632  052777  020000  172066           BIS     #CLK,@TXCSR     ;POKE CLK UP
 2199  007640  016703  172052                   MOV     RXDBUF,R3       ;SET UP ERROR MESSAGE
 2200  007644  005004                           CLR     R4      ;DATA CHAR
 2201  007646  010400                   1$:     MOV     R4,R0   ;EXPECTED
 2202  007650  012767  000013  171244           MOV     #11.,SHIFT      ;# OF SHIFTS
 2203  007656  010467  171616                   MOV     R4,$TMP1        ;"TO BE SHIFTED CHARACTER"
 2204  007662  004767  007312                   JSR     PC,ODD8 ;CALC PARITY
 2205  007666  000241                           CLC
 2206  007670  006167  171604                   ROL     $TMP1   ;GENERATE START BIT
 2207  007674  052767  002000  171576           BIS     #BIT10,$TMP1    ;GENERATE STOP BIT
 2208                                   ;$TMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
 2209  007702  004767  007204                   JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
 2210  007706  017701  172004                   MOV     @RXDBUF,R1      ;ACTUAL
 2211  007712  020001                           CMP     R0,R1   ;COMPARE EXP VS ACT
 2212  007714  001401                           BEQ     .+4
 2213  007716  104002                           ERROR   2       ;DATA CHARS SHOULD MATCH
 2214                                                           ;THERE SHOULD BE NO PARITY ERROR
 2215  007720  005204                           INC     R4      ;UPGRADE NEXT CHAR
 2216  007722  105704                           TSTB    R4      ;LAST CHAR ?
 2217  007724  001350                           BNE     1$
 2218
 2219                                           ;;THIS TEST PERFORMS BINARY DATA CHECK ON THE
 2220                                           ;;RECEIVER
 2221                                           ;;LENGTH:EIGHT PLUS PARITY
 2222                                           ;;MODE:SYNEXT
 2223                                           ;;PARITY:EVEPAR
 2224                                           ;;
 2225                                   ;;*****************************************************************
 2226  007726  000004                   TST25:  SCOPE
 2227  007730  052777  000400  171770           BIS     #MRESET,@TXCSR  ;MASTER RESET
 2228  007736  012777  020000  171756           MOV     #SYNEXT,@PARCSR ;SET THE MODE
 2229  007744  052777  000400  171754           BIS     #MRESET,@TXCSR  ;MASTER RESET
 2230
 2231                                   ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
 2232  007752  012777  064001  171746           MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
 2233
 2234                                   ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
 2235  007760  012777  027400  171734           MOV     #SYNEXT!EIGHT!EVEPAR!0,@PARCSR
 2236  007766  052777  000020  171716           BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
 2237                                   ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
```

```
                                        INITIALIZE THE COMMON TAGS

2238  007774  042777  020000  171724            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2239  010002  052777  020000  171716            BIS     #CLK,@TXCSR     ;POKE CLK UP
2240  010010  016703  171702                    MOV     RXDBUF,R3       ;SET UP ERROR MESSAGE
2241  010014  005004                            CLR     R4      ;DATA CHAR
2242  010016  010400                    1$:     MOV     R4,R0   ;EXPECTED
2243  010020  012767  000011  171074            MOV     #9.,SHIFT       ;# OF SHIFTS
2244  010026  010467  171446                    MOV     R4,$TMP1        ;"TO BE SHIFTED CHARACTER"
2245  010032  004767  007226                    JSR     PC,EVEN8        ;CALC PARITY
2246                                    ;$TMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
2247  010036  004767  007050                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2248  010042  017701  171650                    MOV     @RXDBUF,R1      ;ACTUAL
2249  010046  020001                            CMP     R0,R1   ;COMPARE EXP VS ACT
2250  010050  001401                            BEQ     .+4
2251  010052  104002                            ERROR   2       ;DATA CHARS SHOULD MATCH
2252                                                            ;THERE SHOULD BE NO PARITY ERROR
2253  010054  005204                            INC     R4      ;UPGRADE NEXT CHAR
2254  010056  105704                            TSTB    R4      ;LAST CHAR ?
2255  010060  001356                            BNE     1$
2256
2257                                            ;;THIS TEST PERFORMS BINARY DATA CHECK ON THE
2258                                            ;;RECEIVER
2259                                            ;;LENGTH:EIGHT PLUS PARITY
2260                                            ;;MODE:SYNEXT
2261                                            ;;PARITY:ODDPAR
2262                                            ;;
2263                                    ;;**************************************************************************
2264  010062  000004                    TST26:  SCOPE
2265  010064  052777  000400  171634            BIS     #MRESET,@TXCSR  ;MASTER RESET
2266  010072  012777  020000  171622            MOV     #SYNEXT,@PARCSR ;SET THE MODE
2267  010100  052777  000400  171620            BIS     #MRESET,@TXCSR  ;MASTER RESET
2268
2269                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2270  010106  012777  064001  171612            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2271
2272                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2273  010114  012777  027000  171600            MOV     #SYNEXT!EIGHT!ODDPAR!0,@PARCSR
2274  010122  052777  000020  171562            BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2275
2276  010130  042777  020000  171570    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2277  010136  052777  020000  171562            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2278  010144  016703  171546                    BIS     #CLK,@TXCSR     ;POKE CLK UP
2279  010150  005004                            MOV     RXDBUF,R3       ;SET UP ERROR MESSAGE
2280  010152  010400                            CLR     R4      ;DATA CHAR
2281  010154  012767  000011  170740    1$:     MOV     R4,R0   ;EXPECTED
2282  010162  010467  171312                    MOV     #9.,SHIFT       ;# OF SHIFTS
2283  010166  004767  007006                    MOV     R4,$TMP1        ;"TO BE SHIFTED CHARACTER"
2284                                            JSR     PC,ODD8 ;CALC PARITY
2285  010172  004767  006714           ;$TMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
2286  010176  017701  171514                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2287  010202  020001                            MOV     @RXDBUF,R1      ;ACTUAL
2288  010204  001401                            CMP     R0,R1   ;COMPARE EXP VS ACT
2289  010206  104002                            BEQ     .+4
2290                                            ERROR   2       ;DATA CHARS SHOULD MATCH
2291  010210  005204                                            ;THERE SHOULD BE NO PARITY ERROR
2292  010212  105704                            INC     R4      ;UPGRADE NEXT CHAR
2293  010214  001356                            TSTB    R4      ;LAST CHAR ?
                                                BNE     1$
```

```
DZDUS-A MACY11 27(1006)  03-FEB-77  07:52  PAGE 49
DZDUSA.M11    13-OCT-76 08:39          INITIALIZE THE COMMON TAGS

                                                ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
                                                ;;OF THE RECEIVER LOGIC
                                                ;;MODE:ISYMOD
                                                ;;LENGTH:FIVE
                                                ;;NOTE: RXDONE SHOULD NEVER ASSERT
                                                ;;CHAR: 26 (SYNC)
                                                ;;
                                        ;;**************************************************************************
010216  000004                        TST27:  SCOPE
010220  052777  000400  171500                BIS     #MRESET,@TXCSR  ;MASTER RESET
010226  012777  000000  171466                MOV     #ISYMOD,@PARCSR ;SET THE MODE
010234  052777  000400  171464                BIS     #MRESET,@TXCSR  ;MASTER RESET

                                        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
010242  012777  064001  171456                MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                        ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
010250  012777  000026  171444                MOV     #ISYMOD!FIVE!NOPAR!26,@PARCSR
010256  052777  000020  171426                BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
                                                .POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
010264  042777  020000  171434                BIC     #CLK,@TXCSR     ;POKE CLK DOWN
010272  052777  020000  171426                BIS     #CLK,@TXCSR     ;POKE CLK UP

                                        ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
010300  042777  020000  171420                BIC     #CLK,@TXCSR     ;POKE CLK DOWN
010306  052777  020000  171412                BIS     #CLK,@TXCSR     ;POKE CLK UP
010314  052777  000400  171370                BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
010322  012767  000003  170574                MOV     #3,COUNT        ;# OF SYNC CHARS
010330  012767  000154  171142  1$:           MOV     #154,$TMP1      ;CHAR TO BE SHIFTED
010336  012767  000007  170556                MOV     #7,SHIFT        ;# OF SHIFTS
010344  004767  006542                        JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
010350  105777  171336                        TSTB    @RXCSR  ;RXDONE ?
010354  100001                                BPL     .+4
010356  104004                                ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
010360  005367  170540                        DEC     COUNT   ;# OF SYNC CHARS
010364  001361                                BNE     1$

                                                ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
                                                ;;OF THE RECEIVER LOGIC
                                                ;;MODE:ISYMOD
                                                ;;LENGTH:SIX
                                                ;;NOTE: RXDONE SHOULD NEVER ASSERT
                                                ;;CHAR: 26 (SYNC)
                                                ;;
                                        ;;**************************************************************************
010366  000004                        TST30:  SCOPE
010370  052777  000400  171330                BIS     #MRESET,@TXCSR  ;MASTER RESET
010376  012777  000000  171316                MOV     #ISYMOD,@PARCSR ;SET THE MODE
010404  052777  000400  171314                BIS     #MRESET,@TXCSR  ;MASTER RESET

                                        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
010412  012777  064001  171306                MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                        ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
010420  012777  002026  171274                MOV     #ISYMOD!SIX!NOPAR!26,@PARCSR
010426  052777  000020  171256                BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
```

```
2350
2351  010434  042777  020000  171264          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2352  010442  052777  020000  171256          BIS     #CLK,@TXCSR     ;POKE CLK UP
2353                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2354  010450  042777  020000  171250          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2355  010456  052777  020000  171242          BIS     #CLK,@TXCSR     ;POKE CLK UP
2356  010464  052777  000400  171220          BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2357  010472  012767  000003  170424          MOV     #3,COUNT        ;# OF SYNC CHARS
2358  010500  012767  000254  170772  1$:     MOV     #254,$TMP1      ;CHAR TO BE SHIFTED
2359  010506  012767  000010  170406          MOV     #8.,SHIFT       ;# OF SHIFTS
2360  010514  004767  006372                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2361  010520  105777  171166                  TSTB    @RXCSR  ;RXDONE ?
2362  010524  100001                          BPL     .+4
2363  010526  104004                          ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
2364  010530  005367  170370                  DEC     COUNT   ;# OF SYNC CHARS
2365  010534  001361                          BNE     1$
2366
2367                                          ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2368                                          ;;OF THE RECEIVER LOGIC
2369                                          ;;MODE:ISYMOD
2370                                          ;;LENGTH:SEVEN
2371                                          ;;NOTE: RXDONE SHOULD NEVER ASSERT
2372                                          ;;CHAR: 26 (SYNC)
2373                                          ;;
2374                                  ;;**************************************************************
2375  010536  000004                  TST31:  SCOPE
2376  010540  052777  000400  171160          BIS     #MRESET,@TXCSR  ;MASTER RESET
2377  010546  012777  000000  171146          MOV     #ISYMOD,@PARCSR ;SET THE MODE
2378  010554  052777  000400  171144          BIS     #MRESET,@TXCSR  ;MASTER RESET
2379
2380                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2381  010562  012777  064001  171136          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2382
2383                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2384  010570  012777  004026  171124          MOV     #ISYMOD!SEVEN!NOPAR!26,@PARCSR
2385  010576  052777  000020  171106          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2386                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2387  010604  042777  020000  171114          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2388  010612  052777  020000  171106          BIS     #CLK,@TXCSR     ;POKE CLK UP
2389                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2390  010620  042777  020000  171100          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2391  010626  052777  020000  171072          BIS     #CLK,@TXCSR     ;POKE CLK UP
2392  010634  052777  000400  171050          BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2393  010642  012767  000003  170254          MOV     #3,COUNT        ;# OF SYNC CHARS
2394  010650  012767  000454  170622  1$:     MOV     #454,$TMP1      ;CHAR TO BE SHIFTED
2395  010656  012767  000011  170236          MOV     #9.,SHIFT       ;# OF SHIFTS
2396  010664  004767  006222                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2397  010670  105777  171016                  TSTB    @RXCSR  ;RXDONE ?
2398  010674  100001                          BPL     .+4
2399  010676  104004                          ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
2400  010700  005367  170220                  DEC     COUNT   ;# OF SYNC CHARS
2401  010704  001361                          BNE     1$
2402
2403                                          ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2404                                          ;;OF THE RECEIVER LOGIC
2405                                          ;;MODE:ISYMOD
```

```
                                                ;;LENGTH:EIGHT
                                                ;;NOTE: RXDONE SHOULD NEVER ASSERT
                                                ;;CHAR: 26 (SYNC)
                                                ;;
                                        ;;**************************************************************************
010706  000004                        TST32:  SCOPE
010710  052777  000400  171010                BIS     #MRESET,@TXCSR  ;MASTER RESET
010716  012777  000000  170776                MOV     #ISYMOD,@PARCSR ;SET THE MODE
010724  052777  000400  170774                BIS     #MRESET,@TXCSR  ;MASTER RESET

                                        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
010732  012777  064001  170766                MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                        ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
010740  012777  006026  170754                MOV     #ISYMOD!EIGHT!NOPAR!26,@PARCSR
010746  052777  000020  170736                BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
                                                ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
010754  042777  020000  170744                BIC     #CLK,@TXCSR     ;POKE CLK DOWN
010762  052777  020000  170736                BIS     #CLK,@TXCSR     ;POKE CLK UP

                                        ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
010770  042777  020000  170730                BIC     #CLK,@TXCSR     ;POKE CLK DOWN
010776  052777  020000  170722                BIS     #CLK,@TXCSR     ;POKE CLK UP
011004  052777  000400  170700                BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
011012  012767  000003  170104                MOV     #3,COUNT        ;# OF SYNC CHARS
011020  012767  001054  170452        1$:     MOV     #1054,$TMP1     ;CHAR TO BE SHIFTED
011026  012767  000012  170066                MOV     #10.,SHIFT      ;# OF SHIFTS
011034  004767  006052                        JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
011040  105777  170646                        TSTB    @RXCSR  ;RXDONE ?
011044  100001                                BPL     .+4
011046  104004                                ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
011050  005367  170050                        DEC     COUNT   ;# OF SYNC CHARS
011054  001361                                BNE     1$

                                                ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
                                                ;;OF THE RECEIVER LOGIC
                                                ;;MODE:SYNEXT
                                                ;;LENGTH:FIVE
                                                ;;NOTE: RXDONE SHOULD NEVER ASSERT
                                                ;;CHAR: 26 (SYNC)
                                                ;;
                                        ;;**************************************************************************
011056  000004                        TST33:  SCOPE
011060  052777  000400  170640                BIS     #MRESET,@TXCSR  ;MASTER RESET
011066  012777  020000  170626                MOV     #SYNEXT,@PARCSR ;SET THE MODE
011074  052777  000400  170624                BIS     #MRESET,@TXCSR  ;MASTER RESET

                                        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
011102  012777  064001  170616                MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR

                                        ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
011110  012777  020026  170604                MOV     #SYNEXT!FIVE!NOPAR!26,@PARCSR
011116  052777  000020  170566                BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
                                        ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
011124  042777  020000  170574                BIC     #CLK,@TXCSR     ;POKE CLK DOWN
011132  052777  020000  170566                BIS     #CLK,@TXCSR     ;POKE CLK UP
011140  052777  000400  170544                BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
```

```
2462  011146  012767  000003  167750          MOV     #3,COUNT          ;# OF SYNC CHARS
2463  011154  012767  000026  170316  1$:     MOV     #26,$TMP1         ;CHAR TO BE SHIFTED
2464  011162  012767  000005  167732          MOV     #5,SHIFT          ;# OF SHIFTS
2465  011170  004767  005716                  JSR     PC,RPOKE          ;SHIFT IN THIS CHAR
2466  011174  105777  170512                  TSTB    @RXCSR  ;RXDONE ?
2467  011200  100001                          BPL     .+4
2468  011202  104004                          ERROR   4         ;RXDONE SHOULD NOT BE ASSERTED
2469  011204  005367  167714                  DEC     COUNT     ;# OF SYNC CHARS
2470  011210  001361                          BNE     1$
2471
2472                                          ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2473                                          ;;OF THE RECEIVER LOGIC
2474                                          ;;MODE:SYNEXT
2475                                          ;;LENGTH:SIX
2476                                          ;;NOTE: RXDONE SHOULD NEVER ASSERT
2477                                          ;;CHAR: 26 (SYNC)
2478                                          ;;
2479                                  ;;**************************************************************************
2480  011212  000004                  TST34:  SCOPE
2481  011214  052777  000400  170504          BIS     #MRESET,@TXCSR    ;MASTER RESET
2482  011222  012777  020000  170472          MOV     #SYNEXT,@PARCSR   ;SET THE MODE
2483  011230  052777  000400  170470          BIS     #MRESET,@TXCSR    ;MASTER RESET
2484
2485                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2486  011236  012777  064001  170462          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2487
2488                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2489  011244  012777  022026  170450          MOV     #SYNEXT!SIX!NOPAR!26,@PARCSR
2490  011252  052777  000020  170432          BIS     #SYNSCH,@RXCSR    ;SET SEARCH SYNC
2491                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2492  011260  042777  020000  170440          BIC     #CLK,@TXCSR       ;POKE CLK DOWN
2493  011266  052777  020000  170432          BIS     #CLK,@TXCSR       ;POKE CLK UP
2494  011274  052777  000400  170410          BIS     #STPSYN,@RXCSR    ;SET STRIP SYNC
2495  011302  012767  000003  167614          MOV     #3,COUNT          ;# OF SYNC CHARS
2496  011310  012767  000026  170162  1$:     MOV     #26,$TMP1         ;CHAR TO BE SHIFTED
2497  011316  012767  000006  167576          MOV     #6,SHIFT          ;# OF SHIFTS
2498  011324  004767  005562                  JSR     PC,RPOKE          ;SHIFT IN THIS CHAR
2499  011330  105777  170356                  TSTB    @RXCSR  ;RXDONE ?
2500  011334  100001                          BPL     .+4
2501  011336  104004                          ERROR   4         ;RXDONE SHOULD NOT BE ASSERTED
2502  011340  005367  167560                  DEC     COUNT     ;# OF SYNC CHARS
2503  011344  001361                          BNE     1$
2504
2505                                          ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2506                                          ;;OF THE RECEIVER LOGIC
2507                                          ;;MODE:SYNEXT
2508                                          ;;LENGTH:SEVEN
2509                                          ;;NOTE: RXDONE SHOULD NEVER ASSERT
2510                                          ;;CHAR: 26 (SYNC)
2511                                          ;;
2512                                  ;;**************************************************************************
2513  011346  000004                  TST35:  SCOPE
2514  011350  052777  000400  170350          BIS     #MRESET,@TXCSR    ;MASTER RESET
2515  011356  012777  020000  170336          MOV     #SYNEXT,@PARCSR   ;SET THE MODE
2516  011364  052777  000400  170334          BIS     #MRESET,@TXCSR    ;MASTER RESET
2517
```

```
2518                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2519  011372  012777  064001  170326            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2520
2521                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2522  011400  012777  024026  170314            MOV     #SYNEXT!SEVEN!NOPAR!26,@PARCSR
2523  011406  052777  000020  170276            BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2524                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2525  011414  042777  020000  170304            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2526  011422  052777  020000  170276            BIS     #CLK,@TXCSR     ;POKE CLK UP
2527  011430  052777  000400  170254            BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2528  011436  012767  000003  167460            MOV     #3,COUNT        ;# OF SYNC CHARS
2529  011444  012767  000026  170026  1$:       MOV     #26,$TMP1       ;CHAR TO BE SHIFTED
2530  011452  012767  000007  167442            MOV     #7,SHIFT        ;# OF SHIFTS
2531  011460  004767  005426                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2532  011464  105777  170222                    TSTB    @RXCSR  ;RXDONE ?
2533  011470  100001                            BPL     .+4
2534  011472  104004                            ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
2535  011474  005367  167424                    DEC     COUNT   ;# OF SYNC CHARS
2536  011500  001361                            BNE     1$
2537
2538                                            ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2539                                            ;;OF THE RECEIVER LOGIC
2540                                            ;;MODE:SYNEXT
2541                                            ;;LENGTH:EIGHT
2542                                            ;;NOTE: RXDONE SHOULD NEVER ASSERT
2543                                            ;;CHAR: 26 (SYNC)
2544                                            ;;
2545                                    ;;**************************************************************************
2546  011502  000004                    TST36:  SCOPE
2547  011504  052777  000400  170214            BIS     #MRESET,@TXCSR  ;MASTER RESET
2548  011512  012777  020000  170202            MOV     #SYNEXT,@PARCSR ;SET THE MODE
2549  011520  052777  000400  170200            BIS     #MRESET,@TXCSR  ;MASTER RESET
2550
2551                                    ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2552  011526  012777  064001  170172            MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2553
2554                                    ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2555  011534  012777  026026  170160            MOV     #SYNEXT!EIGHT!NOPAR!26,@PARCSR
2556  011542  052777  000020  170142            BIS     #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2557                                    ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2558  011550  042777  020000  170150            BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2559  011556  052777  020000  170142            BIS     #CLK,@TXCSR     ;POKE CLK UP
2560  011564  052777  000400  170120            BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2561  011572  012767  000003  167324            MOV     #3,COUNT        ;# OF SYNC CHARS
2562  011600  012767  000026  167672  1$:       MOV     #26,$TMP1       ;CHAR TO BE SHIFTED
2563  011606  012767  000010  167306            MOV     #8.,SHIFT       ;# OF SHIFTS
2564  011614  004767  005272                    JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2565  011620  105777  170066                    TSTB    @RXCSR  ;RXDONE ?
2566  011624  100001                            BPL     .+4
2567  011626  104004                            ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
2568  011630  005367  167270                    DEC     COUNT   ;# OF SYNC CHARS
2569  011634  001361                            BNE     1$
2570
2571                                            ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2572                                            ;;OF THE RECEIVER LOGIC
2573                                            ;;MODE:SYNINT
```

```
2574                                                    ;;LENGTH:FIVE
2575                                                    ;;NOTE: RXDONE SHOULD NEVER ASSERT
2576                                                    ;;CHAR: 26 (SYNC)
2577                                                    ;;
2578                                                ;;**********************************************************************
2579  011636  000004                              TST37:  SCOPE
2580  011640  052777  000400  170060                      BIS     #MRESET,@TXCSR  ;MASTER RESET
2581  011646  012777  030000  170046                      MOV     #SYNINT,@PARCSR ;SET THE MODE
2582  011654  052777  000400  170044                      BIS     #MRESET,@TXCSR  ;MASTER RESET
2583
2584                                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2585  011662  012777  064001  170036                      MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2586
2587                                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2588  011670  012777  030026  170024                      MOV     #SYNINT!FIVE!NOPAR!26,@PARCSR
2589  011676  052777  000020  170006                      BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2590                                                    ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2591  011704  042777  020000  170014                      BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2592  011712  052777  020000  170006                      BIS     #CLK,@TXCSR     ;POKE CLK UP
2593                                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2594  011720  042777  020000  170000                      BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2595  011726  052777  020000  167772                      BIS     #CLK,@TXCSR     ;POKE CLK UP
2596  011734  052777  000400  167750                      BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2597  011742  012767  000003  167154                      MOV     #3,COUNT        ;# OF SYNC CHARS
2598  011750  012767  000026  167522  1$:         MOV     #26,STMP1       ;CHAR TO BE SHIFTED
2599  011756  012767  000005  167136                      MOV     #5,SHIFT        ;# OF SHIFTS
2600  011764  004767  005122                      JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2601  011770  105777  167716                      TSTB    @RXCSR  ;RXDONE ?
2602  011774  100001                              BPL     .+4
2603  011776  104004                              ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
2604  012000  005367  167120                      DEC     COUNT   ;# OF SYNC CHARS
2605  012004  001361                              BNE     1$
2606
2607                                                    ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2608                                                    ;;OF THE RECEIVER LOGIC
2609                                                    ;;MODE:SYNINT
2610                                                    ;;LENGTH:SIX
2611                                                    ;;NOTE: RXDONE SHOULD NEVER ASSERT
2612                                                    ;;CHAR: 26 (SYNC)
2613                                                    ;;
2614                                                ;;**********************************************************************
2615  012006  000004                              TST40:  SCOPE
2616  012010  052777  000400  167710                      BIS     #MRESET,@TXCSR  ;MASTER RESET
2617  012016  012777  030000  167676                      MOV     #SYNINT,@PARCSR ;SET THE MODE
2618  012024  052777  000400  167674                      BIS     #MRESET,@TXCSR  ;MASTER RESET
2619
2620                                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2621  012032  012777  064001  167666                      MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2622
2623                                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2624  012040  012777  032026  167654                      MOV     #SYNINT!SIX!NOPAR!26,@PARCSR
2625  012046  052777  000020  167636                      BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2626                                                    ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2627  012054  042777  020000  167644                      BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2628  012062  052777  020000  167636                      BIS     #CLK,@TXCSR     ;POKE CLK UP
2629                                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
```

INITIALIZE THE COMMON TAGS

```
2630  012070  042777  020000  167630          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2631  012076  052777  020000  167622          BIS     #CLK,@TXCSR     ;POKE CLK UP
2632  012104  052777  000400  167600          BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2633  012112  012767  000003  167004          MOV     #3,COUNT        ;# OF SYNC CHARS
2634  012120  012767  000026  167352  1$:     MOV     #26,$TMP1       ;CHAR TO BE SHIFTED
2635  012126  012767  000006  166766          MOV     #6,SHIFT        ;# OF SHIFTS
2636  012134  004767  004752                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2637  012140  105777  167546                  TSTB    @RXCSR  ;RXDONE ?
2638  012144  100001                          BPL     .+4
2639  012146  104004                          ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
2640  012150  005367  166750                  DEC     COUNT   ;# OF SYNC CHARS
2641  012154  001361                          BNE     1$
2642
2643                                          ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2644                                          ;;OF THE RECEIVER LOGIC
2645                                          ;;MODE:SYNINT
2646                                          ;;LENGTH:SEVEN
2647                                          ;;NOTE: RXDONE SHOULD NEVER ASSERT
2648                                          ;;CHAR: 26 (SYNC)
2649                                          ;;
2650                                  ;;**************************************************************************
2651  012156  000004                  TST41:  SCOPE
2652  012160  052777  000400  167540          BIS     #MRESET,@TXCSR  ;MASTER RESET
2653  012166  012777  030000  167526          MOV     #SYNINT,@PARCSR ;SET THE MODE
2654  012174  052777  000400  167524          BIS     #MRESET,@TXCSR  ;MASTER RESET
2655
2656                                  ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2657  012202  012777  064001  167516          MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2658
2659                                  ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2660  012210  012777  034026  167504          MOV     #SYNINT!SEVEN!NOPAR!26,@PARCSR
2661  012216  052777  000020  167466          BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2662                                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2663  012224  042777  020000  167474          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2664  012232  052777  020000  167466          BIS     #CLK,@TXCSR     ;POKE CLK UP
2665                                  ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2666  012240  042777  020000  167460          BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2667  012246  052777  020000  167452          BIS     #CLK,@TXCSR     ;POKE CLK UP
2668  012254  052777  000400  167430          BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2669  012262  012767  000003  166634          MOV     #3,COUNT        ;# OF SYNC CHARS
2670  012270  012767  000026  167202  1$:     MOV     #26,$TMP1       ;CHAR TO BE SHIFTED
2671  012276  012767  000007  166616          MOV     #7,SHIFT        ;# OF SHIFTS
2672  012304  004767  004602                  JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2673  012310  105777  167376                  TSTB    @RXCSR  ;RXDONE ?
2674  012314  100001                          BPL     .+4
2675  012316  104004                          ERROR   4       ;RXDONE SHOULD NOT BE ASSERTED
2676  012320  005367  166600                  DEC     COUNT   ;# OF SYNC CHARS
2677  012324  001361                          BNE     1$
2678
2679
2680                                          ;END OF PASS
2681                                          ;TYPE NAME OF TEST
2682                                          ;UPDATE PASS COUNT
2683                                          ;CHECK FOR EXIT TO ACT-11
2684                                          ;RESTART TEST
2685
```

```
2686  012326  000004                        .EOP:   SCOPE
2687  012330  004767  000344                        JSR     PC,CKSWR
2688  012334  104401                                TYPE                                    ;TYPE NAME OF TEST
2689  012336  015470                                MEPASS
2690  012340  104413  012572                        CONVRT  ,OUTCRY
2691  012344  104401  015307                        TYPE    ,DEVICE
2692  012350  105767  166576                        TSTB    MULTD   ;ARE YOU RUNNING MULTIPLE DEVICES ?
2693  012354  001511                                BEQ     CCC     ;NO,JUMP AROUND
2694  012356  005767  166604                        TST     ACTREG  ;ARE ANY DEVICES ACTIVE ?
2695  012362  001007                                BNE     RUNIT   ;YES
2696  012364  104401  015321                        TYPE    ,MCOW   ;NO
2697  012370  016700  166572                        MOV     ACTREG,R0       ;DISPLAY ACTREG
2698  012374  000000                                HALT            ;SELECT SOMETHING TO RUN @ ACTREG:
2699                                                ;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
2700  012376  000167  167550                        JMP     .START  ;START OVER AGAIN.....YOU DESELECTED EVERYTHING
2701  012402  062767  000010  166544  RUNIT:        ADD     #10,BASEADD     ;NEXT BLOCK (ADDRESSES)
2702  012410  062767  000010  166544  ZERO:         ADD     #10,BASEIV      ;NEXT BLOCK (VECTORS)
2703  012416  000241                                CLC
2704  012420  006167  166544                        ROL     ROTADD  ;UP DATE ROTATING POINTER
2705  012424  103410                                BCS     2$      ;IS IT THE LAST DEVICE
2706                                                ;TO BE TESTED IN THIS PASS ?
2707  012426  036767  166536  166532                BIT     ROTADD,ACTREG   ;TEST THIS DEVICE FOR ACTIVE STATUS
2708  012434  001762                                BEQ     RUNIT   ;IF NOT ACTIVE, TRY NEXT ADDRESS
2709  012436  004767  000034                        JSR     PC,REPLAY       ;CALCULATE NEW PARAMETERS
2710  012442  000167  000210                        JMP     RESTRT  ;YES IT WAS ACTIVE,TEST THIS DEVICE
2711  012446  012767  000001  166514  2$:           MOV     #1,ROTADD       ;OK!,NOW SET UP  ROTATING
2712                                                ;POINTER FOR NEXT MULTIPLE PASS
2713  012454  016767  166476  166472                MOV     KEEPADD,BASEADD ;RESTORE BASE ADDRESS
2714  012462  016767  166476  166472                MOV     KEEPIV,BASEIV   ;RESTORE BASE INTERRUPT VECTORS
2715  012470  004767  000002                        JSR     PC,REPLAY       ;CALC NEW PARAMETERS
2716  012474  000441                                BR      CCC     ;JUMP AROUND REPLAY
2717  012476  016767  166452  004404  REPLAY:       MOV     BASEADD,DUBASE  ;SET UP FOR NEW ADDRESSES
2718  012504  004767  004246                        JSR     PC,DUADDR       ;CREATE NEW ADDRESSES
2719  012510  016767  166446  167220                MOV     BASEIV,DURIV    ;CREATE DURIV
2720  012516  062767  000002  166436                ADD     #2,BASEIV
2721  012524  016767  166432  167206                MOV     BASEIV,DURIS    ;CREATE DURIS
2722  012532  062767  000002  166422                ADD     #2,BASEIV
2723  012540  016767  166416  167174                MOV     BASEIV,DUTIV    ;CREATE DUTIV
2724  012546  062767  000002  166406                ADD     #2,BASEIV
2725  012554  016767  166402  167162                MOV     BASEIV,DUTIS    ;CREATE DUTIS
2726  012562  016767  167150  166372                MOV     DURIV,BASEIV    ;RESTORE
2727  012570  000207                                RTS     PC
2728
2729  012572  000001                        OUTCRY: 1
2730  012574     006     002                        .BYTE   6,2
2731  012576  001712                                RXCSR
2732
2733  012600                                CCC:
2734  012600  005067  166576                        CLR     $TSTNM                  ;CLEAR TEST NUMBER
2735  012604  005067  166606                        CLR     $ERRPC                  ;CLEAR LAST ERROR PC
2736  012610  005067  166567                        CLR     $ERFLG                  ;CLEAR ERROR FLAG
2737  012614  005267  166272                        INC     PASCNT                  ;UPDATE PASS COUNT
2738  012620  016767  166266  166254                MOV     PASCNT,LIGHTS           ;DISPLAY PASS COUNT
2739  012626  016767  166260  166700                MOV     PASCNT,$PASS            ;PASS COUNT TO APT
2740  012634  013701  000042                        MOV     @#42,R1                 ;CHECK FOR ACT-11 OR DDP
2741  012640  001406                                BEQ     RESTRT          ;IF NO CONTINUE TESTING
```

INITIALIZE THE COMMON TAGS

```
2742  012642  000005                          RESET
2743  012644  000005                          RESET
2744  012646  004711                SENDAD:   JSR     PC,(R1)
2745  012650  000240                          NOP
2746  012652  000240                          NOP
2747  012654  000240                          NOP
2748  012656  106427  000340        RESTRT:   MTPS    #340    ;PREVENT INTERRUPTS (PRIO: 7)
2749  012662  004767  000012                  JSR     PC,CKSWR
2750  012666  012767  003364  166512          MOV     #TST1+2,$LPADR          ;SET LAST ADDRESS POINTER
2751  012674  000167  170462                  JMP     TST1
2752
2753                                          ;CHECK SWITCH REGISTER ROUTINE.
2754                                          ;CHECKS TO ALLOW FOR <↑G> TO ALLOW
2755                                          ;THE CHANGING OF LOCATION 176
2756
2757  012700  005737  000042        CKSWR:    TST     @#42
2758  012704  001040                          BNE     OUT
2759  012706  022767  000176  166524          CMP     #SWREG,SWR      ;SOFTWARE SWR PRESENT?
2760  012714  001034                          BNE     OUT             ;NO--LEAVE
2761  012716  105777  166522                  TSTB    @$TKS           ;CHECK TTY READY
2762  012722  100031                          BPL     OUT             ;NO--LEAVE
2763  012724  017767  166516  000422          MOV     @$TKB,.MSG      ;GET CHARACTER
2764  012732  042767  177600  000414          BIC     #177600,.MSG    ;STRIP JUNK
2765  012740  122767  000007  000406          CMPB    #7,.MSG         ;IS IT <↑G> ?
2766  012746  001017                          BNE     OUT             ;NO
2767  012750  104401  016075                  TYPE    ,MCNTG
2768  012754  005137  013014        CNTLU:    COM     @#RDSW
2769  012760  104401  016105                  TYPE    ,MMSWR
2770  012764  104413                          CONVRT
2771  012766  013016                          SWREGL
2772  012770  104406  016116                  INSTR,MMNEW
2773  012774  104410                          PARAM
2774  012776  000000                          0
2775  013000  177777                          177777
2776  013002  000176                          SWREG
2777  013004     000     001          .BYTE    0,1
2778  013006  005037  013014        OUT:      CLR     @#RDSW
2779  013012  000207                          RTS     PC
2780  013014  000000                RDSW:     .WORD   0
2781  013016  000001                SWREGL:   1
2782  013020     006     002                  .BYTE   6,2
2783  013022  000176                          SWREG
2784
2785  013024  000005                          5
2786
2787                                          ;CHECK FOR FREEZE ON CURRENT DATA
2788
2789  013026  004767  177646        .SCOP1:   JSR     PC,CKSWR
2790  013032  032777  001000  166400          BIT     #SW09,@SWR
2791  013040  001402                          BEQ     1$
2792  013042  016716  166042                  MOV     LOCK,(SP)
2793  013046  000002                1$:       RTI
2794                                .SBTTL    TYPE ROUTINE
2795
2796                                ;;*****************************************************************
2797                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
```

```
                                        TYPE ROUTINE

                                ;*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
                                ;*

013050  105767  166403          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
013054  100002                          BPL     1$              ;;BR IF YES
013056  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
013060  000430                          BR      3$              ;;LEAVE
013062  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
013064  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
013070  122767  000001  166450          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
013076  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
013100  132767  000100  166441          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
013106  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
013110  010067  000004'                 MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
013114  004767  000006'                 JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
013120  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
013122  132767  000040  166417  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
013130  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
013132  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
013134  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
013136  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
013140  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
013142  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
013146  000002                          RTI                     ;;RETURN
013150  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
013154  001430                          BEQ     8$
013156  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
013162  001006                          BNE     5$
013164  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
013166  104401                          TYPE                    ;;TYPE A CR AND LF
013170  001523                          $CRLF
013172  105067  000130                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
013176  000755                          BR      2$              ;;GET NEXT CHARACTER
013200  004767  000056          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
013204  126726  166246          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
013210  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
013212  016746  166236                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
013216  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
013222  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
013224  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
013230  105367  000072                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
013234  000770                          BR      7$              ;;LOOP

                                ;HORIZONTAL TAB PROCESSOR
```

```
                                        TYPE ROUTINE

2854  013236  112716  000040          8$:     MOVB    #'  ,(SP)          ;;REPLACE TAB WITH SPACE
2855  013242  004767  000014          9$:     JSR     PC,$TYPEC           ;;TYPE A SPACE
2856  013246  132767  000007  000052          BITB    #7,$CHARCNT         ;;BRANCH IF NOT AT
2857  013254  001372                          BNE     9$                  ;;TAB STOP
2858  013256  005726                          TST     (SP)+               ;;POP SPACE OFF STACK
2859  013260  000724                          BR      2$                  ;;GET NEXT CHARACTER
2860  013262  105777  166162          $TYPEC: TSTB    @$TPS               ;;WAIT UNTIL PRINTER IS READY
2861  013266  100375                          BPL     $TYPEC
2862  013270  116677  000002  166154          MOVB    2(SP),@$TPB         ;;LOAD CHAR TO BE TYPED INTO DATA REG.
2863  013276  122766  000015  000002          CMPB    #CR,2(SP)           ;;IS CHARACTER A CARRIAGE RETURN?
2864  013304  001003                          BNE     1$                  ;;BRANCH IF NO
2865  013306  105067  000014                  CLRB    $CHARCNT            ;;YES--CLEAR CHARACTER COUNT
2866  013312  000406                          BR      $TYPEX              ;;EXIT
2867  013314  122766  000012  000002  1$:     CMPB    #LF,2(SP)           ;;IS CHARACTER A LINE FEED?
2868  013322  001402                          BEQ     $TYPEX              ;;BRANCH IF YES
2869  013324  105227                          INCB    (PC)+               ;;COUNT THE CHARACTER
2870  013326  000000          $CHARCNT:.WORD  0                           ;;CHARACTER COUNT STORAGE
2871  013330  000207          $TYPEX: RTS     PC
2872
2873
2874                                  ;ASCII STRING INPUT ROUTINE
2875
2876  013332  017667  000000  000014  .INSTR: MOV     @(SP),.MSG          ;PICK UP MESSAGE
2877  013340  062716  000002                  ADD     #2,(SP)             ;JUMP AROUND MESSAGE FOR RTI
2878  013344  105767  166176                  TSTB    $ENV                ;APT CONTROL
2879  013350  001036                          BNE     INSTR2              ;YES NO TYPE
2880  013352  104401          .INST1: TYPE
2881  013354  000000          .MSG:   0
2882  013356  012704  016130                  MOV     #INBUF,R4           ;GET STARTING LOC OF INBUF
2883  013362  012703  000007                  MOV     #7,R3               ;MAX # OF CHARS
2884  013366  105777  166052          1$:     TSTB    @$TKS   ;TTY FLAG
2885  013372  100375                          BPL     1$
2886  013374  117714  166046                  MOVB    @$TKB,(R4)          ;TAKE CHAR
2887  013400  142714  000200                  BICB    #200,(R4)           ;STRIP
2888  013404  121427  000025                  CMPB    (R4),#25            ;IS IT <↑G>
2889  013410  001760                          BEQ     .INST1
2890  013412  122427  000015                  CMPB    (R4)+,#15           ;CHECK FOR CR
2891  013416  001413                          BEQ     INSTR2
2892  013420  105777  166024          2$:     TSTB    @$TPS   ;TEST FLAG
2893  013424  100375                          BPL     2$
2894  013426  117777  166014  166016          MOVB    @$TKB,@$TPB         ;ECHO CHARACTER
2895  013434  005303                          DEC     R3      ;DID YOU TYPE TOO MANY CHARS ?
2896  013436  001353                          BNE     1$
2897  013440  104401          .INSTE: TYPE
2898  013442  015415                          MQM     ;?
2899  013444  000742                          BR      .INST1  ;RETRY
2900  013446  000002          INSTR2: RTI
2901
2902
2903
2904  013450  011605          .PARAM: MOV     (SP),R5 ;PUT CONTENTS OF SP INTO R5
2905  013452  012567  000162                  MOV     (R5)+,LOLIM         ;PUT LOW LIMIT INTO LOLIM
2906  013456  012567  000160                  MOV     (R5)+,HILIM         ;PUT HIGH LIMIT INTO HILIM
2907  013462  012567  000156                  MOV     (R5)+,DEVADR        ;PUT STORE LOC INTO DEVADR
2908  013466  112567  000154                  MOVB    (R5)+,LOBITS        ;PUT MASK INTO LOBITS
2909  013472  112567  000151                  MOVB    (R5)+,ADRCNT        ;PUT COUNT INTO ADRCNT
```

```
2910  013476  010516                          MOV     R5,(SP) ;RESTORE RETURN ADDR ON STACK FOR RTI
2911  013500  005005                  PARAM1: CLR     R5
2912  013502  012704  016130                  MOV     #INBUF,R4
2913  013506  122714  000015                  CMPB    #15,(R4)        ;CR ?
2914  013512  001420                          BEQ     PARERR  ;YOU TYPED CR TOO SOON !
2915  013514  121427  000060          1$:     CMPB    (R4),#60        ;LOW LIMIT ASCII 0
2916  013520  002415                          BLT     PARERR
2917  013522  121427  000067                  CMPB    (R4),#67        ;HIGH LIMIT ASCII 7
2918  013526  003012                          BGT     PARERR
2919  013530  142714  000060                  BICB    #60,(R4)        ;CONVERT TO OCTAL
2920  013534  152405                          BISB    (R4)+,R5        ;STORE AWAY ITS AN OK CHAR
2921  013536  122714  000015                  CMPB    #15,(R4)        ;CR ?
2922  013542  001414                          BEQ     LIMITS  ;NOW CHECK FOR HIGH &LOW LIMIT CONDS
2923  013544  006305                          ASL     R5      ;ALLOCATE ROOM FOR NEXT CHAR
2924  013546  006305                          ASL     R5
2925  013550  006305                          ASL     R5
2926  013552  000760                          BR      1$
2927  013554  122714  000015          PARERR: CMPB    #15,(R4)        ;CR?
2928  013560  001003                          BNE     120$
2929  013562  005737  013014                  TST     @#RDSW          ;CK SWR USED
2930  013566  001023                          BNE     PARTI
2931  013570  104407                  120$:   INSTER  ;RETRY
2932  013572  000742                          BR      PARAM1
2933
2934                                          ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2935
2936  013574  020567  000042          LIMITS: CMP     R5,HILIM
2937  013600  101365                          BHI     PARERR  ;THE # IS TOO HIGH
2938  013602  020567  000032                  CMP     R5,LOLIM
2939  013606  103762                          BLO     PARERR  ;THE # IS TOO LOW
2940  013610  136705  000032                  BITB    LOBITS,R5       ;TEST BY MASKINGTHE #
2941  013614  001357                          BNE     PARERR
2942
2943                                          ;STORE NUMBER AT SPECIFIED ADDRESS
2944
2945  013616  016704  000022                  MOV     DEVADR,R4       ;GET STARTING ADDR OF
2946  013622  010524                  1$:     MOV     R5,(R4)+        ;STORE AT THIS ADDR
2947  013624  062705  000002                  ADD     #2,R5
2948  013630  105367  000013                  DECB    ADRCNT  ;HOW MANY TIMES + 2 ?
2949  013634  001372                          BNE     1$
2950  013636  000002                  PARTI:  RTI
2951  013640  000000                  LOLIM:  0
2952  013642  000000                  HILIM:  0
2953  013644  000000                  DEVADR: 0
2954  013646  000000                  LOBITS: 0
2955          013647                  ADRCNT=LOBITS+1
2956
2957                                          ;SAVE PC OF TEST THAT FAILED AND R0-R5
2958
2959  013650  016667  000004  165250  .SAV05: MOV     4(SP),SAVPC
2960
2961                                          ;SAVE R0-R5
2962
2963  013656  010567  165612          SV05:   MOV     R5,$REG5
2964  013662  010467  165604                  MOV     R4,$REG4
2965  013666  010367  165576                  MOV     R3,$REG3
```

```
2966  013672  010267  165570                  MOV     R2,$REG2
2967  013676  010167  165562                  MOV     R1,$REG1
2968  013702  010067  165554                  MOV     R0,$REG0
2969  013706  000002                           RTI
2970
2971                                          ;RESTORE R0-R5
2972
2973  013710  016700  165546          .RES05: MOV     $REG0,R0
2974  013714  016701  165544                  MOV     $REG1,R1
2975  013720  016702  165542                  MOV     $REG2,R2
2976  013724  016703  165540                  MOV     $REG3,R3
2977  013730  016704  165536                  MOV     $REG4,R4
2978  013734  016705  165534                  MOV     $REG5,R5
2979  013740  000002                           RTI
2980
2981                                          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2982
2983  013742  104401                  .CONVR: TYPE
2984  013744  015421                          MCRLF   ;CR  LF
2985  013746  017601  000000                  MOV     @(SP),R1        ;PICK UP DATA POINTER
2986  013752  062716  000002                  ADD     #2,(SP) ;SET UP SP FOR RTI
2987  013756  012167  000130                  MOV     (R1)+,WRDCNT    ;PICK UP # OF WORDS FROM TABLE
2988  013762  112167  000126          1$:     MOVB    (R1)+,CHRCNT    ;PICK UP # OF CHARS FROM TABLE
2989  013766  112167  000123                  MOVB    (R1)+,SPACNT    ;PICK UP # OF SPACES FROM TABLE
2990  013772  013167  000120                  MOV     @(R1)+,BINWRD   ;PICK UP ADDRESS OF  MSG
2991                                                                  ;FROM TABLE
2992  013776  016704  000114          2$:     MOV     BINWRD,R4       ;SAVE
2993  014002  116705  000106                  MOVB    CHRCNT,R5       ;SAVE
2994  014006  012700  016172                  MOV     #TEMP,R0        ;STARTING ADDRESS OF TEMP BLOCK
2995  014012  010403                  3$:     MOV     R4,R3           ;SAVE
2996  014014  042703  177770                  BIC     #177770,R3      ;CLR OUT UPPER BITS .. SAVE CHAR
2997  014020  062703  000260                  ADD     #260,R3 ;CONVERT TO ASCII
2998  014024  110320                          MOVB    R3,(R0)+        ;STORE AWAY
2999  014026  006204                          ASR     R4      ;SHIFT FOR NEXT #
3000  014030  006204                          ASR     R4      ;DITTO
3001  014032  006204                          ASR     R4      ;DITTO
3002  014034  005305                          DEC     R5      ;DEC CHAR COUNT
3003  014036  001365                          BNE     3$      ;DO IT AGAIN ?
3004  014040  012703  016234                  MOV     #MDATA,R3       ;STARTING ADDRESS OF MDATA BLOCK
3005  014044  114023          4$:     MOVB    -(R0),(R3)+     ;REVERSE THE ORDER OF NUMBERS
3006  014046  105367  000042                  DECB    CHRCNT  ;DEC CHAR COUNT
3007  014052  001374                          BNE     4$      ;DO IT AGAIN ?
3008  014054  105767  000035                  TSTB    SPACNT  ;HOW MANY SPACES ?
3009  014060  001405                          BEQ     6$      ;TYPE # IF BR =0
3010  014062  112723  000240          5$:     MOVB    #240,(R3)+      ;"SPACE" IN ASCII
3011  014066  105367  000023                  DECB    SPACNT  ;DEC # OF SPACE COUNT
3012  014072  001373                          BNE     5$      ;DO IT AGAIN ?
3013  014074  105013          6$:     CLRB    (R3)    ;INSERT "0" FOR TTY OUTPUT ROUTINE
3014  014076  104401                          TYPE
3015  014100  016234                          MDATA   ;THIS MESSAGE
3016  014102  005367  000004                  DEC     WRDCNT  ;HOW MANY #'S ?
3017  014106  001325                          BNE     1$      ;DO THIS ROUTINE AGAIN IF NOT EQUAL TO 0
3018  014110  000002                          RTI     ;RETURN TO PROGRAM
3019  014112  000000          WRDCNT: 0
3020  014114  000000          CHRCNT: 0
3021          014115          SPACNT=CHRCNT+1
```

```
3022  014116  000000                  BINWRD: 0
3023
3024                                          ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
3025                                          ;BUFFER TO THE CHARACTERS "N" AND "Y".
3026                                          ;IF THE CHARACTER IS "N" CLEAR THE FLAG
3027                                          ;IF THE CHARACTER IS "Y" SET THE FLAG
3028
3029  014120  017605  000000          .SETFLG:MOV     @(SP),R5
3030  014124  122767  000116  001776          CMPB    #'N,INBUF       ;IS IT "N" ?
3031  014132  001002                          BNE     1$
3032  014134  105015                          CLRB    (R5)    ;000
3033  014136  000406                          BR      2$
3034  014140  122767  000131  001762  1$:     CMPB    #'Y,INBUF       ;IS IT "Y" ?
3035  014146  001005                          BNE     3$
3036  014150  112715  177777                  MOVB    #-1,(R5)        ;377
3037  014154  062716  000002          2$:     ADD     #2,(SP)
3038  014160  000002                          RTI
3039  014162  104407                  3$:     INSTER  ;RETRY
3040  014164  000755                          BR      .SETFLG
3041                                  .SBTTL  ERROR HANDLER ROUTINE
3042
3043                                  ;;**********************************************************************
3044                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
3045                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
3046                                  ;*AND GO TO SAVIT ON ERROR
3047                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3048                                  ;*SW15=1         HALT ON ERROR
3049                                  ;*SW13=1         INHIBIT ERROR TYPEOUTS
3050                                  ;*SW10=1         BELL ON ERROR
3051                                  ;*SW09=1         LOOP ON ERROR
3052                                  ;*CALL
3053                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
3054
3055  014166                          $ERROR:
3056  014166  105267  165211          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
3057  014172  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
3058  014174  016777  165202  165240          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
3059  014202  032777  002000  165230          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
3060  014210  001402                          BEQ     1$              ;;NO - SKIP
3061  014212  104401  001516                  TYPE    ,$BELL          ;;RING BELL
3062  014216  005267  165170          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
3063  014222  011667  165170                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
3064  014226  162767  000002  165162          SUB     #2,$ERRPC
3065  014234  117767  165156  165152          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
3066  014242  032777  020000  165170          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
3067  014250  001004                          BNE     20$             ;;SKIP TYPEOUTS
3068  014252  004767  000072                  JSR     PC,SAVIT        ;;GO TO USER ERROR ROUTINE
3069  014256  104401  001523                  TYPE    ,$CRLF
3070  014262                          20$:
3071  014262  122767  000001  165256          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
3072  014270  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
3073  014272  116767  165116  000004          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
3074  014300  004767  000016'                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
3075  014304     000                  21$:     .BYTE   0
3076  014305     000                          .BYTE   0
3077  014306  000777                  22$:     BR      22$             ;;APT ERROR LOOP
```

```
                                        ERROR HANDLER ROUTINE

 3078  014310  005777  165124          2$:     TST     @SWR            ;;HALT ON ERROR
 3079  014314  100001                          BPL     3$              ;;SKIP IF CONTINUE
 3080  014316  000000                          HALT                    ;;HALT ON ERROR!
 3081  014320  032777  001000  165112  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
 3082  014326  001402                          BEQ     4$              ;;BR IF NO
 3083  014330  016716  165054                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
 3084  014334  005767  165154          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
 3085  014340  001402                          BEQ     5$              ;;BR IF NONE
 3086  014342  016716  165146                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
 3087  014346                          5$:
 3088  014346  000002                          RTI                     ;;RETURN
 3089  014350  010067  164554          SAVIT:  MOV     R0,HLD0
 3090  014354  010167  164552                  MOV     R1,HLD1
 3091  014360  010267  164550                  MOV     R2,HLD2
 3092  014364  010367  164546                  MOV     R3,HLD3
 3093  014370  010467  164544                  MOV     R4,HLD4
 3094  014374  010567  164542                  MOV     R5,HLD5
 3095  014400  016767  164776  164536        MOV     $TSTNM,HLD6
 3096
 3097                                  .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
 3098
 3099                                  ;;*****************************************************************
 3100                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
 3101                                  ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
 3102                                  ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
 3103
 3104  014406                          $ERRTYP:
 3105  014406  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 3106  014412  010046                          MOV     R0,-(SP)        ;;SAVE R0
 3107  014414  005000                          CLR     R0              ;;PICKUP THE ITEM INDEX
 3108  014416  153700  001414                  BISB    @#$ITEMB,R0
 3109  014422  001004                          BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
 3110                                                                  ;;TYPE THE PC OF THE ERROR
 3111  014424  016746  164766                  MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
 3112                                                                  ;;ERROR ADDRESS
 3113  014430  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 3114  014432  000426                          BR      6$              ;;GET OUT
 3115  014434  005300                  1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
 3116  014436  006300                          ASL     R0              ;;       WORK FOR THE ERROR TABLE
 3117  014440  006300                          ASL     R0
 3118  014442  006300                          ASL     R0
 3119  014444  062700  001652                  ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
 3120  014450  012067  000004                  MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
 3121  014454  001404                          BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
 3122  014456  104401                          TYPE                    ;;TYPE THE "ERROR MESSAGE"
 3123  014460  000000                  2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
 3124  014462  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 3125  014466  012067  000004          3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
 3126  014472  001404                          BEQ     5$              ;;SKIP TYPEOUT IF 0
 3127  014474  104401                          TYPE                    ;;TYPE THE "DATA HEADER"
 3128  014476  000000                  4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
 3129  014500  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 3130  014504  011000                  5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
 3131  014506  001004                          BNE     7$              ;;GO TYPE THE DATA
 3132  014510  012600                  6$:     MOV     (SP)+,R0        ;;RESTORE R0
 3133  014512  104401  001523                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
```

```
                                        ERROR MESSAGE TYPEOUT ROUTINE

3134  014516  000207                            RTS     PC              ;;RETURN
3135  014520                            7$:
3136  014520  013046                            MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
3137  014522  104402                            TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3138  014524  005710                            TST     (R0)            ;;IS THERE ANOTHER NUMBER?
3139  014526  001770                            BEQ     6$              ;;BR IF NO
3140  014530  104401  014536                    TYPE    ,8$             ;;TYPE TWO(2) SPACES
3141  014534  000771                            BR      7$              ;;LOOP
3142  014536  020040      000       8$:         .ASCIZ  /  /            ;;TWO(2) SPACES
3143          014542                            .EVEN
3144                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3145
3146                                  ;;**************************************************************
3147                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3148                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
3149                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3150                                  ;*CALL:
3151                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3152                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
3153                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3154                                  ;*      .BYTE   M               ;;M=1 OR 0
3155                                  ;*                                      ;;1=TYPE LEADING ZEROS
3156                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
3157                                  ;*
3158                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3159                                  ;*$TYPOS OR $TYPOC
3160                                  ;*CALL:
3161                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3162                                  ;*      TYPON                   ;;CALL FOR TYPEOUT
3163                                  ;*
3164                                  ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3165                                  ;*CALL:
3166                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3167                                  ;*      TYPOC                   ;;CALL FOR TYPEOUT
3168
3169  014542  017646  000000            $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
3170  014546  116667  000001  000211            MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
3171  014554  112667  000207                    MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
3172  014560  062716  000002                    ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
3173  014564  000406                            BR      $TYPON
3174  014566  112767  000001  000171  $TYPOC:  MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
3175  014574  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
3176  014602  112767  000005  000154  $TYPON:  MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
3177  014610  010346                            MOV     R3,-(SP)        ;;SAVE R3
3178  014612  010446                            MOV     R4,-(SP)        ;;SAVE R4
3179  014614  010546                            MOV     R5,-(SP)        ;;SAVE R5
3180  014616  116704  000145                    MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
3181  014622  005404                            NEG     R4
3182  014624  062704  000006                    ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
3183  014630  110467  000132                    MOVB    R4,$OMODE       ;;SAVE IT FOR USE
3184  014634  116704  000125                    MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
3185  014640  016605  000012                    MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
3186  014644  005003                            CLR     R3              ;;CLEAR THE OUTPUT WORD
3187  014646  006105                    1$:     ROL     R5              ;;ROTATE MSB INTO "C"
3188  014650  000404                            BR      3$              ;;GO DO MSB
3189  014652  006105                    2$:     ROL     R5              ;;FORM THIS DIGIT
```

```
3190  014654  006105                        ROL     R5
3191  014656  006105                        ROL     R5
3192  014660  010503                        MOV     R5,R3
3193  014662  006103                3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
3194  014664  105367  000076                DECB    $OMODE          ;;TYPE THIS DIGIT?
3195  014670  100016                        BPL     7$              ;;BR IF NO
3196  014672  042703  177770                BIC     #177770,R3      ;;GET RID OF JUNK
3197  014676  001002                        BNE     4$              ;;TEST FOR 0
3198  014700  005704                        TST     R4              ;;SUPPRESS THIS 0?
3199  014702  001403                        BEQ     5$              ;;BR IF YES
3200  014704  005204                4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
3201  014706  052703  000060                BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
3202  014712  052703  000040          5$:   BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
3203  014716  110367  000040                MOVB    R3,8$           ;;SAVE FOR TYPING
3204  014722  104401  014762                TYPE    ,8$             ;;GO TYPE THIS DIGIT
3205  014726  105367  000032          7$:   DECB    $OCNT           ;;COUNT BY 1
3206  014732  003347                        BGT     2$              ;;BR IF MORE TO DO
3207  014734  002402                        BLT     6$              ;;BR IF DONE
3208  014736  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
3209  014740  000744                        BR      2$              ;;GO DO THE LAST DIGIT
3210  014742  012605                6$:     MOV     (SP)+,R5        ;;RESTORE R5
3211  014744  012604                        MOV     (SP)+,R4        ;;RESTORE R4
3212  014746  012603                        MOV     (SP)+,R3        ;;RESTORE R3
3213  014750  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
3214  014756  012616                        MOV     (SP)+,(SP)
3215  014760  000002                        RTI                     ;;RETURN
3216  014762     000                8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
3217  014763     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
3218  014764     000                $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
3219  014765     000                $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
3220  014766  000000                $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
3221                                        ;ENTER HERE ON POWER FAILURE
3222
3223
3224  014770                        $PWRDN:
3225  014770  010046                .PFAIL: MOV     R0,-(SP)                ;SAVE R0-R5 ON PROCESSOR STACK
3226  014772  010146                        MOV     R1,-(SP)
3227  014774  010246                        MOV     R2,-(SP)
3228  014776  010346                        MOV     R3,-(SP)
3229  015000  010446                        MOV     R4,-(SP)
3230  015002  010546                        MOV     R5,-(SP)
3231  015004  016746  163014                MOV     24,-(SP)
3232  015010  010667  164102                MOV     SP,SAVSP                ;SAVE STACK POINTER
3233  015014  012767  015026  163002        MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
3234  015022  000000                        HALT                            ;HALT ON POWER DOWN NORMAL
3235  015024  000777                        BR      .
3236
3237                                        ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3238
3239  015026  016706  164064        RESTAR: MOV     SAVSP,SP                ;RESTORE STACK POINTER
3240  015032  012605                        MOV     (SP)+,R5                ;RESTORE R0-R5
3241  015034  012604                        MOV     (SP)+,R4
3242  015036  012603                        MOV     (SP)+,R3
3243  015040  012602                        MOV     (SP)+,R2
3244  015042  012601                        MOV     (SP)+,R1
3245  015044  012600                        MOV     (SP)+,R0
```

```
3246  015046  012767  014770  162750          MOV     #.PFAIL,24              ;SET UP FOR POWER FAILURE
3247  015054  106427  000340                  MTPS    #340
3248  015060  012706  001100                  MOV     #STACK,SP
3249  015064  005067  001102                  CLR     TEMP
3250  015070  005267  001076                  INC     TEMP
3251  015074  001375                          BNE     .-4
3252  015076  104413                          CONVRT
3253  015100  015122                          PFTAB
3254  015102  104401                          TYPE
3255  015104  015424                          MPFAIL
3256  015106  005067  164271                  CLR     $ERFLG
3257  015112  005067  164300                  CLR     $ERRPC
3258  015116  000177  163762                  JMP     @RETURN
3259  015122  000001                  PFTAB:  1
3260  015124      006     002                 .BYTE   6,2
3261  015126  000207                          RETURN
3262  015130  005015  042012  053125  MTITLE: .ASCIZ  <15><12><12>/DUV11 DZDUS-A TAPE C /<15><12>
3263  015136  030461  042040  042132
3264  015144  051525  040455  052040
3265  015152  050101  020105  020103
3266  015160  005015      000
3267  015163      015  053012  041505  MVECTO: .ASCIZ  <15><12>/VEC ADD-/
3268  015170  040440  042104  000055
3269  015176  005015  051461  020124  MREGAD: .ASCIZ  <15><12>/1ST DEV: REC CSR ADD-/
3270  015204  042504  035126  051040
3271  015212  041505  041440  051123
3272  015220  040440  042104  000055
3273  015226  005015  052515  052114  MMULT:  .ASCIZ  <15><12>/MULT DEV ? (Y OR N)-/
3274  015234  042040  053105  037440
3275  015242  024040  020131  051117
3276  015250  047040  026451      000
3277  015255      015  046012  051501  MLASTD: .ASCIZ  <15><12>/LAST DEV: REC CSR ADDR-/
3278  015262  020124  042504  035126
3279  015270  051040  041505  041440
3280  015276  051123  040440  042104
3281  015304  026522      000
3282  015307      075  042504  044526  DEVICE: .ASCIZ  /=DEVICE  /
3283  015314  042503  020040      000
3284  015321      015  051412  046105  MCOW:   .ASCIZ  <15><12>/SELECT TO RUN @ACTREG/
3285  015326  041505  020124  047524
3286  015334  051040  047125  040040
3287  015342  041501  051124  043505
3288  015350      000
3289  015351      015  047412  043126  MRANGE: .ASCIZ  <15><12>/OVFLO:RETYPE LAST DEV RXCSR ADDS-/
3290  015356  047514  051072  052105
3291  015364  050131  020105  040514
3292  015372  052123  042040  053105
3293  015400  051040  041530  051123
3294  015406  040440  042104  026523
3295  015414      000
3296  015415      040  037440      000  MQM:    .ASCIZ  /  ?/
3297  015421      015  000012           MCRLF:  .ASCIZ  <15><12>
3298  015424  043120  044501  026114  MPFAIL: .ASCIZ  /PFAIL,  RESTART AT TEST IN PROGRESS/
3299  015432  020040  042522  052123
3300  015440  051101  020124  052101
3301  015446  052040  051505  020124
```

```
3302  015454  047111  050040  047522
3303  015462  051107  051505  000123
3304  015470  005015  047105  020104  MEPASS: .ASCIZ  <15><12>/END OF PASS TAPE C/
3305  015476  043117  050040  051501
3306  015504  020123  040524  042520
3307  015512  041440     000
3308  015515     015  051012     000  MR:     .ASCIZ  <15><12>/R/
3309  015521     015  052012  051505  MTSTPC: .ASCIZ  <15><12>/TEST PC-/
3310  015526  020124  041520  000055
3311  015534  005015  047514  045503  MLOCK:  .ASCIZ  <15><12>/LOCK ON  TEST? (Y OR N)-/
3312  015542  047440  020116  052040
3313  015550  051505  037524  024040
3314  015556  020131  051117  047040
3315  015564  026451     000
3316  015567     015  021412  047440  MSYNC:  .ASCIZ  <15><12>/# OF SYNC CHARS SELECTED ( 1 OR 2)-/
3317  015574  020106  054523  041516
3318  015602  041440  040510  051522
3319  015610  051440  046105  041505
3320  015616  042524  020104  020050
3321  015624  020061  051117  031040
3322  015632  026451     000
3323  015635     015  044412  020123  MWIRE6: .ASCIZ  <15><12>/IS SEC XMIT SWITCH E55-2 IN? (Y OR N)-/
3324  015642  042523  020103  046530
3325  015650  052111  051440  044527
3326  015656  041524  020110  032505
3327  015664  026465  020062  047111
3328  015672  020077  054450  047440
3329  015700  020122  024516  000055
3330  015706  005015  051511  051440  MWIRE5: .ASCIZ  <15><12>/IS SEC REC SWITCH E55-3 IN? (Y OR N)-/
3331  015714  041505  051040  041505
3332  015722  051440  044527  041524
3333  015730  020110  032505  026465
3334  015736  020063  047111  020077
3335  015744  054450  047440  020122
3336  015752  024516  000055
3337  015756  005015  051511  047440  MWIRE4: .ASCIZ  <15><12>/IS OPT CLR ENABLE SWITCH E55-1 IN? (Y OR N)-/
3338  015764  052120  041440  051114
3339  015772  042440  040516  046102
3340  016000  020105  053523  052111
3341  016006  044103  042440  032465
3342  016014  030455  044440  037516
3343  016022  024040  020131  051117
3344  016030  047040  026451     000
3345  016035     015  005012  031510  MEXTJ:  .ASCIZ  <15><12><12>/H315 CONNECTOR ON ?(Y OR N)-/
3346  016042  032461  041440  047117
3347  016050  042516  052103  051117
3348  016056  047440  020116  024077
3349  016064  020131  051117  047040
3350  016072  026451     000
3351  016075     015  020012  043536  MCNTG:  .ASCIZ  <15><12>/ ↑G  /
3352  016102  020040     000
3353  016105     040  053523  036522  MMSWR:  .ASCIZ  / SWR=   /
3354  016112  020040  000040
3355  016116  020040  047040  053505  MMNEW:  .ASCIZ  /   NEW=  /
3356  016124  020075  000040
3357                                  .EVEN
```

```
3358
3359                                            ;BUFFERS FOR INPUT-OUTPUT
3360
3361  016130  000000                    INBUF:  0
3362          016172                    .=.+40
3363  016172  000000                    TEMP:   0
3364          016234                    .=.+40
3365  016234  000000                    MDATA:  0
3366          016276                    .=.+40
3367                                    .SBTTL  SCOPE HANDLER ROUTINE
3368
3369                                    ;;*****************************************************************
3370                                    ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
3371                                    ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
3372                                    ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
3373                                    ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3374                                    ;*SW14=1         LOOP ON TEST
3375                                    ;*SW11=1         INHIBIT ITERATIONS
3376                                    ;*SW09=1         LOOP ON ERROR
3377                                    ;*SW08=1         LOOP ON TEST IN SWR<7:0>
3378                                    ;*CALL
3379                                    ;*      SCOPE           ;;SCOPE=IOT
3380
3381  016276                            $SCOPE:
3382
3383                                            ;SCOPE LOOP AND INTERATION HANDLER
3384
3385  016276                            .SCOPE:
3386  016276  004767  174376                    JSR     PC,CKSWR
3387  016302  005067  163110                    CLR     $ERRPC          ;CLEAR LAST ERROR PC
3388  016306  022716  003364                    CMP     #TST1+2,(SP)    ;IS SCOPE AT BEGINING OF TEST 1?
3389  016312  001413                            BEQ     $XTSTR          ;YES NO LOOP.
3390
3391  016314  000406                    TTST:   BR      1$              ;GO TO 1$ (IF LOCK SW02=1)
3392  016316  105777  163122                    TSTB    @$TKS           ;KEYBOARD DONE?
3393  016322  100123                            BPL     $OVER           ;BR IF NO
3394  016324  017766  163116  177776            MOV     @$TKB,-2(SP)    ;CLEAR DONE BIT
3395  016332  032777  040000  163100  1$:       BIT     #BIT14,@$WR     ;;LOOP ON PRESENT TEST?
3396  016340  001114                            BNE     $OVER           ;;YES IF SW14=1
3397                                    ;#####START OF CODE FOR THE XOR TESTER#####
3398  016342  000416                    $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
3399                                                                    ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
3400  016344  013746  000004                    MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
3401  016350  012737  016370  000004            MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
3402  016356  005737  177060                    TST     @#177060        ;;TIME OUT ON XOR?
3403  016362  012637  000004                    MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
3404  016366  000463                            BR      $SVLAD          ;;GO TO THE NEXT TEST
3405  016370  022626                    5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
3406  016372  012637  000004                    MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
3407  016376  000423                            BR      7$              ;;LOOP ON THE PRESENT TEST
3408  016400                            6$:;#####END OF CODE FOR THE XOR TESTER#####
3409  016400  032777  000400  163032            BIT     #BIT08,@$WR     ;;LOOP ON SPEC. TEST?
3410  016406  001404                            BEQ     2$              ;;BR IF NO
3411  016410  127767  163024  162764            CMPB    @$WR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
3412  016416  001465                            BEQ     $OVER           ;;BR IF YES
3413  016420  105767  162757            2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
```

```
3414  016424  001421                        BEQ     3$                ;;BR IF NO
3415  016426  126767  162763  162747        CMPB    $ERMAX,$ERFLG     ;;MAX. ERRORS FOR THIS TEST OCCURRED?
3416  016434  101015                        BHI     3$                ;;BR IF NO
3417  016436  032777  001000  162774        BIT     #BIT09,@SWR       ;;LOOP ON ERROR?
3418  016444  001404                        BEQ     4$                ;;BR IF NO
3419  016446  016767  162736  162732  7$:   MOV     $LPERR,$LPADR     ;;SET LOOP ADDRESS TO LAST SCOPE
3420  016454  000446                        BR      $OVER
3421  016456  105067  162721          4$:   CLRB    $ERFLG            ;;ZERO THE ERROR FLAG
3422  016462  005067  163024                CLR     $TIMES            ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
3423  016466  000415                        BR      1$                ;;ESCAPE TO THE NEXT TEST
3424  016470  032777  004000  162742  3$:   BIT     #BIT11,@SWR       ;;INHIBIT ITERATIONS?
3425  016476  001011                        BNE     1$                ;;BR IF YES
3426  016500  005767  163030                TST     $PASS             ;;IF FIRST PASS OF PROGRAM
3427  016504  001406                        BEQ     1$                ;;      INHIBIT ITERATIONS
3428  016506  005267  162672                INC     $ICNT             ;;INCREMENT ITERATION COUNT
3429  016512  026767  162774  162664        CMP     $TIMES,$ICNT      ;;CHECK THE NUMBER OF ITERATIONS MADE
3430  016520  002024                        BGE     $OVER             ;;BR IF MORE ITERATION REQUIRED
3431  016522  012767  000001  162654  1$:   MOV     #1,$ICNT          ;;REINITIALIZE THE ITERATION COUNTER
3432  016530  016767  000056  162754        MOV     $MXCNT,$TIMES     ;;SET NUMBER OF ITERATIONS TO DO
3433  016536  105267  162640          $SVLAD: INCB  $TSTNM            ;;COUNT TEST NUMBERS
3434  016542  116767  162634  162762        MOVB    $TSTNM,$TESTN     ;;SET TEST NUMBER IN APT MAILBOX
3435  016550  011667  162632                MOV     (SP),$LPADR       ;;SAVE SCOPE LOOP ADDRESS
3436  016554  011667  162630                MOV     (SP),$LPERR       ;;SAVE ERROR LOOP ADDRESS
3437  016560  005067  162730                CLR     $ESCAPE           ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
3438  016564  112767  000001  162623        MOVB    #1,$ERMAX         ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
3439  016572  016777  162604  162642  $OVER:  MOV     $TSTNM,@DISPLAY   ;;DISPLAY TEST NUMBER
3440  016600  016716  162602                MOV     $LPADR,(SP)       ;;FUDGE RETURN ADDRESS
3441  016604  000002                  4$:   RTI
3442  016606  001407                  BRW:  1407
3443  016610  000432                  BRX:  432
3444  016612  000005                  $MXCNT: 5                         ;;MAX. NUMBER OF ITERATIONS
3445                                  .SBTTL  TRAP DECODER
3446
3447                                  ;;**************************************************************
3448                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3449                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3450                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
3451                                  ;*GO TO THAT ROUTINE.
3452
3453  016614  010046                  $TRAP:  MOV     R0,-(SP)          ;;SAVE R0
3454  016616  016600  000002                  MOV     2(SP),R0          ;;GET TRAP ADDRESS
3455  016622  005740                          TST     -(R0)             ;;BACKUP BY 2
3456  016624  111000                          MOVB    (R0),R0           ;;GET RIGHT BYTE OF TRAP
3457  016626  006300                          ASL     R0                ;;POSITION FOR INDEXING
3458  016630  016000  016650                  MOV     $TRPAD(R0),R0     ;;INDEX TO TABLE
3459  016634  000200                          RTS     R0                ;;GO TO ROUTINE
3460
3461
3462                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
3463
3464  016636  011646                  $TRAP2: MOV     (SP),-(SP)        ;;MOVE THE PC DOWN
3465  016640  016666  000004  000002          MOV     4(SP),2(SP)       ;;MOVE THE PSW DOWN
3466  016646  000002                          RTI                       ;;RESTORE THE PSW
3467
3468                                  .SBTTL  TRAP TABLE
3469
```

```
                                        TRAP TABLE

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

                        ;       ROUTINE
                        ;       -------
016650  016636          $TRPAD: .WORD   $TRAP2
016652  013050                  $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
016654  014566                  $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
016656  014542                  $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
016660  014602                  $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)

016662  013026                  .SCOP1  ;;CALL=SCOP1    TRAP+5(104405)
016664  013332                  .INSTR  ;;CALL=INSTR    TRAP+6(104406)
016666  013440                  .INSTER ;;CALL=INSTER   TRAP+7(104407)
016670  013450                  .PARAM  ;;CALL=PARAM    TRAP+10(104410)
016672  013650                  .SAVOS  ;;CALL=SAVOS    TRAP+11(104411)
016674  013710                  .RESOS  ;;CALL=RESOS    TRAP+12(104412)
016676  013742                  .CONVRT ;;CALL=CONVRT   TRAP+13(104413)
016700  014120                  .SETFLG ;;CALL=SETFLG   TRAP+14(104414)
                        ;*****************************************************************
                                ;UTILITIES
                        ;*****************************************************************

                        ;THIS UTILITY CALCULATES PRIORITY LEVEL
016702  006367  000044  DULEV:  ASL     DUPRT   ;SHIFT LEFT
016706  006367  000040          ASL     DUPRT   ;
016712  006367  000034          ASL     DUPRT   ;
016716  006367  000030          ASL     DUPRT   ;
016722  006367  000024          ASL     DUPRT   ;
016726  016767  000020  000020  MOV     DUPRT,LESS1     ;MOVE THIS TO LESS1
016734  162767  000001  000012  SUB     #1,LESS1        ;CREATE LESS1
016742  042767  000037  000004  BIC     #37,LESS1       ;CLEAR TNZVC
016750  000207                  RTS     PC
016752  000240          DUPRT:  PR5
016754  000200          LESS1:  PR4     ;LEVEL TO ALLOW INTERRUPTS

                                ;NEW DU ADDRESSES
016756  016767  000126  162726  DUADDR: MOV     DUBASE,RXCSR    ;XXX0
016764  005267  000120          INC     DUBASE
016770  016767  000114  162716          MOV     DUBASE,HRXCSR   ;XXX1
016776  005267  000106          INC     DUBASE
017002  016767  000102  162706          MOV     DUBASE,RXDBUF   ;XXX2
017010  016767  000074  162704          MOV     DUBASE,PARCSR   ;XXX2
017016  005267  000066          INC     DUBASE
017022  016767  000062  162670          MOV     DUBASE,HRXDBUF  ;XXX3
017030  016767  000054  162666          MOV     DUBASE,HPARCSR  ;XXX3
017036  005267  000046          INC     DUBASE
017042  016767  000042  162656          MOV     DUBASE,TXCSR    ;XXX4
017050  005267  000034          INC     DUBASE
017054  016767  000030  162646          MOV     DUBASE,HTXCSR   ;XXX5
017062  005267  000022          INC     DUBASE
017066  016767  000016  162636          MOV     DUBASE,TXDBUF   ;XXX6
017074  005267  000010          INC     DUBASE
017100  016767  000004  162626          MOV     DUBASE,HTXDBUF  ;XXX7
017106  000207                  RTS     PC
```

```
3526  017110  000000                  DUBASE: 0
3527
3528                                          ;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3529                                          ;INFORMATION CONTAINED IN $TMP1 AND IT IS
3530                                          ;SHIFTED IN BY THE CONTENTS OF SHIFT
3531  017112  042777  040000  162606  RPOKE:  BIC     #MTDATA,@TXCSR
3532  017120  005067  162356                  CLR     $TMP2
3533  017124  006067  162350                  ROR     $TMP1   ;FORCE CARRY
3534  017130  006067  162346                  ROR     $TMP2   ;PICK UP CARRY IN BIT 15
3535  017134  006267  162342                  ASR     $TMP2   ;SHIFT INTO BIT 14
3536  017140  042767  100000  162334          BIC     #BIT15,$TMP2    ;CLR BIT 15
3537  017146  056777  162330  162552          BIS     $TMP2,@TXCSR    ;POKE MAINT DATA
3538  017154  042777  020000  162544          BIC     #CLK,@TXCSR     ;POKE CLK
3539  017162  052777  020000  162536          BIS     #CLK,@TXCSR     ;
3540  017170  005367  161726                  DEC     SHIFT
3541  017174  001346                          BNE     RPOKE
3542  017176  000207                          RTS     PC
3543                                          ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3544  017200  016767  162274  162274  ODD8:   MOV     $TMP1,$TMP2     ;SAVE TEMP1
3545  017206  005067  162272                  CLR     $TMP3
3546  017212  012727  000010                  MOV     #8.,(PC)+
3547  017216  000000                  4$:     0
3548  017220  006067  162256          1$:     ROR     $TMP2
3549  017224  005567  162254                  ADC     $TMP3
3550  017230  005367  177762                  DEC     4$
3551  017234  001371                          BNE     1$
3552  017236  006067  162242                  ROR     $TMP3
3553  017242  103404                          BCS     2$
3554  017244  052767  000400  162226          BIS     #BIT8,$TMP1     ;SET ODD PARITY
3555  017252  000403                          BR      3$
3556  017254  042767  000400  162216  2$:     BIC     #BIT8,$TMP1     ;CLR EVEN PARITY
3557                                          ;$TMP1 NOW HAS ODD PARITY CHARACTER
3558  017262  000207                  3$:     RTS     PC
3559
3560                                          ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3561  017264  016767  162210  162210  EVEN8:  MOV     $TMP1,$TMP2     ;SAVE TEMP1
3562  017272  005067  162206                  CLR     $TMP3
3563  017276  012727  000010                  MOV     #8.,(PC)+
3564  017302  000000                  4$:     0
3565  017304  006067  162172          1$:     ROR     $TMP2
3566  017310  005567  162170                  ADC     $TMP3
3567  017314  005367  177762                  DEC     4$
3568  017320  001371                          BNE     1$
3569  017322  006067  162156                  ROR     $TMP3
3570  017326  103004                          BCC     2$
3571  017330  052767  000400  162142          BIS     #BIT8,$TMP1     ;SET EVEN PARITY
3572  017336  000403                          BR      3$
3573  017340  042767  000400  162132  2$:     BIC     #BIT8,$TMP1     ;CLR ODD PARITY
3574                                          ;$TMP1 NOW HAS EVEN PARITY CHARACTER
3575  017346  000207                  3$:     RTS     PC
3576
3577  017350  062716  000002          TRPREG: ADD     #2,(SP) ;ALLOW IT TO "CRUNCH" INTO HLT BACK
3578                                                          ;IN MAIN PART OF THE PROGRAM
3579  017354  000002                          RTI
3580          000001                  .END
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
AAA     003200           1291#
ABASE = 000000            874     915
ACDW1 = 000000            874     917
ACDW2 = 000000            874     918
ACPUOP= 000000            874     889
ACTREG  001166            733#   1247*   1261*   1262*   1269*   2694    2697    2707
ADDW0 = 000000            874     919
ADDW1 = 000000            874     920
ADDW10= 000000            874     929
ADDW11= 000000            874     930
ADDW12= 000000            874     931
ADDW13= 000000            874     932
ADDW14= 000000            874     933
ADDW15= 000000            874     934
ADDW2 = 000000            874     921
ADDW3 = 000000            874     922
ADDW4 = 000000            874     923
ADDW5 = 000000            874     924
ADDW6 = 000000            874     925
ADDW7 = 000000            874     926
ADDW8 = 000000            874     927
ADDW9 = 000000            874     928
ADEVCT= 000000            874     880
ADEVM = 000000            874     916
ADRCNT= 013647           2909*   2948*   2955#
AENV  = 000000            874     885
AENVM = 000000            874     886
AFATAL= 000000            874     877
AMADR1= 000000            874     902
AMADR2= 000000            874     906
AMADR3= 000000            874     909
AMADR4= 000000            874     912
AMAMS1= 000000            874     896
AMAMS2= 000000            874     904
AMAMS3= 000000            874     907
AMAMS4= 000000            874     910
AMSGAD= 000000            874     882
AMSGLG= 000000            874     883
AMSGTY= 000000            874     876
AMTYP1= 000000            874     897
AMTYP2= 000000            874     905
AMTYP3= 000000            874     908
AMTYP4= 000000            874     911
APASS = 000000            874     879
APRIOR= 000000            874
APTCSU= 000040            534#   2824
APTENV= 000001            534#   2817    3071
APTSIZ= 000200            534#   1166
APTSPO= 000100            534#   2819
ASWREG= 000000            874     887
ATESTN= 000000            874     878
AUNIT = 000000            874     881
AUSWR = 000000            874     888
AVECT1= 000000            874     913
AVECT2= 000000            874     914
BASEAD  001154            728#   1229*   1266*   1267    1273*   1275*   2701*   2713*   2717
```

```
BASEIV  001162            731#  1239*  2702*  2714*  2719   2720*  2721   2722*  2723   2724*  2725   2726*
BBB     003026           1246   1250#
BINWRD  014116           2990*  2992   3022#
BITW  = 002000            800#   993#
BIT0  = 000001            663#   776    807    969   1000
BIT00 = 000001            653#   663
BIT01 = 000002            652#   662
BIT02 = 000004            651#   661
BIT03 = 000010            650#   660
BIT04 = 000020            649#   659
BIT05 = 000040            648#   658
BIT06 = 000100            647#   657
BIT07 = 000200            646#   656
BIT08 = 000400            645#   655   3409
BIT09 = 001000            644#   654   3081   3417
BIT1  = 000002            662#   775    968
BIT10 = 002000            643#   766    800    959    993   2163   2207   3059
BIT11 = 004000            642#   765    958   3424
BIT12 = 010000            641#   764    781    957    974
BIT13 = 020000            640#   763    780    799    956   973    992    3066
BIT14 = 040000            639#   762    779    798    955   972    991    3395
BIT15 = 100000            638#   761    778    797    954   971    990    3536
BIT2  = 000004            661#   774    967
BIT3  = 000010            660#   773    806    966    999
BIT4  = 000020            659#   772    805    965    998
BIT5  = 000040            658#   771    804    964    997
BIT6  = 000100            657#   770    803    963    996
BIT7  = 000200            656#   769    802    962    995
BIT8  = 000400            655#   768    784    801    961   977    994    3554   3556   3571   3573
BIT9  = 001000            654#   767    783    960    976
BPTVEC= 000014            670#
BREAK = 000001            807#  1000#  1356   1408   1460   1512   1564   1616   1668   1707   1746   1785   1823
                         1864   1905   1946   1984   2025   2066   2107   2144   2188   2232   2270   2309   2345
                         2381   2417   2453   2486   2519   2552   2585   2621   2657
BRW     016606           3442#
BRX     016610           3443#
CARDET= 010000            764#   957#
CCC     012600           2693   2716   2733#
CHRCNT  014114           2988*  2993   3006*  3020#  3021
CKSWR   01270C           2687   2749   2757#  2789   3386
CLK   = 020000            799#   992#  1356   1362   1363   1408   1414   1415   1460   1466   1467   1512   1518
                         1519   1564   1570   1571   1616   1622   1623   1668   1674   1675   1677   1678   1707
                         1713   1714   1716   1717   1746   1752   1753   1755   1756   1785   1791   1792   1794
                         1795   1823   1829   1830   1832   1833   1864   1870   1871   1873   1874   1905   1911
                         1912   1914   1915   1946   1952   1953   1955   1956   1984   1990   1991   1993   1994
                         2025   2031   2032   2034   2035   2066   2072   2073   2075   2076   2107   2113   2114
                         2116   2117   2144   2150   2151   2153   2154   2188   2194   2195   2197   2198   2232
                         2238   2239   2270   2276   2277   2309   2315   2316   2318   2319   2345   2351   2352
                         2354   2355   2381   2387   2388   2390   2391   2417   2423   2424   2426   2427   2453
                         2459   2460   2486   2492   2493   2519   2525   2526   2552   2558   2559   2585   2591
                         2592   2594   2595   2621   2627   2628   2630   2631   2657   2663   2664   2666   2667
                         3538   3539
CNTLU   012754           1196   2768#
CONVRT= 104413           2690   2770   3252   3488#
COUNT   001124            708#  2321*  2328*  2357*  2364*  2393*  2400*  2429*  2436*  2462*  2469*  2495*  2502*
                         2528*  2535*  2561*  2568*  2597*  2604*  2633*  2640*  2669*  2676*
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
CR     = 000015            578#   2863   2873
CRLF   = 000200            579#   2834   2873
CTS    = 020000            763#    956#
DDISP  = 177570            585#    841   1154
DEVADR  013644            2907*  2945   2953#
DEVICE  015307            2691   3282#
DF1     002132            1025   1029   1033   1037   1083#
DH1     002067            1023   1027   1031   1072#
DISPLA  001442             841#  1154*  1162*  1187*  3058*  3439*
DISPRE  000174             682#  1162   1187
DNA    = 100000            797#    990#
DNAINT= 000040             804#    997#
DSC    = 100000            761#    954#
DSINTE= 000040             771#    964#
DSR    = 001000            767#    960#
DSWR   = 177570            584#    840   1153
DTR    = 000002            775#    968#
DT1     002116            1024   1028   1032   1078#
DT4     002126            1036   1081#
DUADDR  016756            1228   2718   3508#
DUBASE  017110            1224   1227   2717*  3508   3509*  3510   3511*  3512   3513   3514*  3515   3516   3517*
                          3518   3519*  3520   3521*  3522   3523*  3524   3526#
DULEV   016702            1286   3495#
DUPRT   016752            1285*  3495*  3496*  3497*  3498*  3499*  3500   3504#
DURIS   001740            1052#  2721*
DURIV   001736            1051#  1235   1238   1239   2719*  2726
DUTIS   001744            1054#  2725*
DUTIV   001742            1053#  2723*
EIGHT  = 006000             792#   985#  1463   1515   1567   1619   1788   1949   2110   2147   2191   2235   2273
                          2420   2555
EMTVEC= 000030              673#  1138*  1139*
EM1     001762            1022   1058#
EM2     002022            1026   1064#
EM3     002043            1030   1067#
EM4     001746            1034   1056#
ERRCNT  001114             701#
ERRVEC= 000004              666#  1151   1152*  1163*  3400   3401*  3403*  3406*
EVENB   017264            2160   2245   3561#
EVEPAR= 001400              795#   988#  1826   1867   1908   1949   2147   2235
EVPAR  = 000400             784#   977#
FIVE   = 000000             789#   982#  1671   1826   1987   2312   2456   2588
FRMERR= 020000              780#   973#  1684   1723   1762   1801
GNS    = ****** U          3476   3477   3478   3479   3482   3483   3484   3485   3486   3487   3488   3489
HDXEN  = 000010             806#   999#
HILIM   013642            2906*  2936   2952#
HLD0    001130             710#  1078   3089*
HLD1    001132             711#  1078   3090*
HLD2    001134             712#  3091*
HLD3    001136             713#  3092*
HLD4    001140             714#  3093*
HLD5    001142             715#  3094*
HLD6    001144             716#  3095*
HOLD    001120             706#
HPARCS  001724            1045#  3516*
HRXCSR  001714            1041#  3510*
HRXDBU  001720            1043#  3515*
```

```
HT    = 000011            576#   2832   2873
HTXCSR  001730           1047#   3520*
HTXDBU  001734           1049#   3524*
INBUF   016130           1294    1298   2882   2912   3030   3034   3361#
INIFLG  001172            741#   1197   1200*
INSTER= 104407           1302    2931   3039   3484#
INSTR = 104406           1219    1230   1240   1251   1276   1292   1305   1309   1313   1317   1328   2772   3483#
INSTR2  013446           2879    2891   2900#
IOTVEC= 000020            671#   1136*  1137*
ISYMOD= 000000            788#    981#  1664   1671   1703   1710   1742   1749   1781   1788   1819   1826   1860
                         1867    1901   1908   1942   1949   1980   1987   2021   2028   2062   2069   2103   2110
                         2140    2147   2184   2191   2305   2312   2341   2348   2377   2384   2413   2420
JMRBY   001153            724#   1205*  1320
KEEPAD  001156            729#   1227*  1229   1273   1275   2713
KEEPIV  001164            732#   1238*  2714
LASTAD  001160            730#   1256   1267   1281
LESS1   016754           3500*   3501*  3502*  3505#
LF    = 000012            577#   2867   2873
LIGHTS  001102            691#   2738*
LIMITS  013574           2922    2936#
LOBITS  013646           2908*   2940   2954#  2955
LOCK    001110            699#   2792
LOKFLG  001174            743#
LOLIM   013640           2905*   2938   2951#
MCNTG   016075           2767    3351#
MCOW    015321           2696    3284#
MCRLF   015421           2984    3297#
MDATA   016234           3004    3015   3365#
MEPASS  015470           2689    3304#
MEXT  = 010000            811#   1004#
MEXTJ   016035           1318    3345#
MINT  = 004000            810#   1003#  1356   1408   1460   1512   1564   1616   1668   1707   1746   1785   1823
                         1864    1905   1946   1984   2025   2066   2107   2144   2188   2232   2270   2309   2345
                         2381    2417   2453   2486   2519   2552   2585   2621   2657
MLASTD  015255           1252    3277#
MLOCK   015534           3311#
MMNEW   016116           2772    3355#
MMSWR   016105           2769    3353#
MMULT   015226           1241    3273#
MPFAIL  015424           3255    3298#
MQM     015415           2898    3296#
MR      015515           1338    3308#
MRANGE  015351           1277    3289#
MREGAD  015176           1220    3269#
MRESET= 000400            801#    994#  1351   1353   1403   1405   1455   1457   1507   1509   1559   1561   1611
                         1613    1663   1665   1702   1704   1741   1743   1780   1782   1818   1820   1859   1861
                         1900    1902   1941   1943   1979   1981   2020   2022   2061   2063   2102   2104   2139
                         2141    2183   2185   2227   2229   2265   2267   2304   2306   2340   2342   2376   2378
                         2412    2414   2448   2450   2481   2483   2514   2516   2547   2549   2580   2582   2616
                         2618    2652   2654
MSYNC   015567           1293    3316#
MTDATA= 040000            798#    991#  1356   1408   1460   1512   1564   1616   1668   1707   1746   1785   1823
                         1864    1905   1946   1984   2025   2066   2107   2144   2188   2232   2270   2309   2345
                         2381    2417   2453   2486   2519   2552   2585   2621   2657   3531
MTITLE  015130           1199    3262#
MTSTPC  015521           1329    3309#
```

```
MULTD   001152            723#  1243    1244    2692
MVECTO  015163           1231   3267#
MWIRE4  015756           1314   3337#
MWIRE5  015706           1310   3330#
MWIRE6  015635           1306   3323#
NEXT    001106            698#
NOPAR = 000000            793#   986#  1359    1411    1463    1515    1567    1619    1671    1710    1749    1788    2312
                         2348   2384   2420    2456    2489    2522    2555    2588    2624    2660
ODDPAR= 001000            794#   987#  1987    2028    2069    2110    2191    2273
ODDB    017200           2204   2283   3544#
ONCE    002610           1198   1201#
OPTCLR  001151            722#  1316
OUT     013006           2758   2760   2762    2766    2778#
OUTCRY  012572           2690   2729#
OUTMUL  003166           1249   1274   1285#
OVRRUN= 040000            779#   972#
PARAM = 104410           1221   1232   1253    1278    1330    2773    3485#
PARAM1  013500           2911#  2932
PARCSR  001722           1044#  1352#  1359#   1404#   1411#   1456#   1463#   1508#   1515#   1560#   1567#   1612#   1619#
                         1664#  1671#  1703#   1710#   1742#   1749#   1781#   1788#   1819#   1826#   1860#   1867#   1901#
                         1908#  1942#  1949#   1980#   1987#   2021#   2028#   2062#   2069#   2103#   2110#   2140#   2147#
                         2184#  2191#  2228#   2235#   2266#   2273#   2305#   2312#   2341#   2348#   2377#   2384#   2413#
                         2420#  2449#  2456#   2482#   2489#   2515#   2522#   2548#   2555#   2581#   2588#   2617#   2624#
                         2653#  2660#  3513#
PAREN = 001000            783#   976#
PARER = 010000            781#   974#  1835    1876    1917    1958    1996    2037    2078    2119
PARERR  013554           2914   2916   2918    2927#   2937    2939    2941
PARTI   013636           2930   2950#
PASCNT  001112            700#  1174#  2737#   2738    2739
PFTAB   015122           3253   3259#
PIRQ  = 177772            583#
PIRQVE= 000240            677#
POPRO = 012600            756#   949#
POP1SP= 005726            754#   947#
POP2SP= 022626            758#   951#
PRO   = 000000            600#
PR1   = 000040            601#
PR2   = 000100            602#
PR3   = 000140            603#
PR4   = 000200            604#  3505
PR5   = 000240            605#  3504
PR6   = 000300            606#
PR7   = 000340            607#
PS    = 177776            580#   581
PSW   = 177776            581#
PUSHRO= 010046            755#   948#
PUSH1S= 005746            753#   946#
PUSH2S= 024646            757#   950#
PWRVEC= 000024            672#  1142#  1143#
RDSW    013014           2768#  2778#  2780#   2929
RECACT= 004000            765#   958#
REPLAY  012476           2709   2715   2717#
RESTAR  015026           3233   3239#
RESTRT  012656           2710   2741   2748#
RESVEC= 000010            667#
RESOS = 104412           3487#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
RETURN  001104            697#  1179*  1337*  1339   3258
RING  = 040000            762#   955#
RINTEN= 000100            770#   963#
ROTADD  001170            736#  1248*  1260*  1262   1264*  1269   1272*  2704*  2707   2711*
RPOKE   017112           1368   1380   1384   1420   1432   1436   1472   1484   1488   1524   1536   1540   1576
                         1588   1592   1628   1640   1644   1682   1721   1760   1799   1838   1879   1920   1961
                         1999   2040   2081   2122   2165   2209   2247   2285   2324   2360   2396   2432   2465
                         2498   2531   2564   2600   2636   2672   3531#  3541
RTS   = 000004            774#   967#
RUNA  = ****** U            1     51    137    511
RUNB  = ****** U            1     51    137    511
RUNC  = 000000              1     51    137    511
RUND  = ****** U           21     56    140    516
RUNE  = ****** U           21     56    140    516
RUNF  = ****** U           21     56    140    516
RUNIT   012402            2695   2701#  2708
RXCSR   001712            1040#  1360*  1369   1412*  1421   1464*  1473   1516*  1525   1568*  1577   1620*  1629
                          1672*  1711*  1750*  1789*  1827*  1839   1868*  1880   1909*  1921   1950*  1962   1988*
                          2000   2029*  2041   2070*  2082   2111*  2123   2148*  2192*  2236*  2274*  2313*  2320*
                          2325   2349*  2356*  2361   2385*  2392*  2397   2421*  2428*  2433   2457*  2461*  2466
                          2490*  2494*  2499   2523*  2527*  2532   2556*  2560*  2565   2589*  2596*  2601   2625*
                          2632*  2637   2661*  2668*  2673   2731   3508*
RXDBUF  001716            1042#  1364   1372   1387   1416   1424   1439   1468   1476   1491   1520   1528   1543
                          1572   1580   1595   1624   1632   1647   1683   1685   1722   1724   1761   1763   1800
                          1802   1834   1842   1875   1883   1916   1924   1957   1965   1995   2003   2036   2044
                          2077   2085   2118   2126   2155   2166   2199   2210   2240   2248   2278   2286   3512*
RXDONE= 000200             769#   962#
RXERR = 100000             778#   971#  1684   1723   1762   1801   1835   1876   1917   1958   1996   2037   2078
                          2119
SAVIT   014350            3068   3089#
SAVPC   001126             709#  2959*
SAVSP   001116             702#  3232*  3239
SAVOS = 104411            3486#
SCOPI = 104405            3482#
SEND  = 000020             805#   998#
SEREC   001150             721#  1312
SETFLG= 104414            1242   1307   1311   1315   1319   3489#
SEVEN = 004000             791#   984#  1359   1411   1749   1908   2069   2384   2522   2660
SEXMIT  001147             720#  1308
SHIFT   001122             707#  1366*  1378*  1382*  1418*  1430*  1434*  1470*  1482*  1486*  1522*  1534*  1538*
                          1574*  1586*  1590*  1626*  1638*  1642*  1679*  1718*  1757*  1796*  1836*  1877*  1918*
                          1959*  1997*  2038*  2079*  2120*  2158*  2202*  2243*  2281*  2323*  2359*  2395*  2431*
                          2464*  2497*  2530*  2563*  2599*  2635*  2671*  3540*
SIX   = 002000             790#   983#  1710   1867   2028   2348   2489   2624
SPACNT= 014115            2989*  3008   3011*  3021#
SRO   = 002000             766#   959#
STACK = 001100             571#  1134   1170   1324   3248
STD   = 000010             773#   966#
STFLG   001173             742#  1173*
STKLMT= 177774             582#
STPSYN= 000400             768#   961#  2320   2356   2392   2428   2461   2494   2527   2560   2596   2632   2668
SVOS    013656            2963#
SWR     001440             840#  1132   1153*  1155   1161*  1168*  1184   1186*  1191   1207   1326   2759   2790
                          3059   3066   3078   3081   3395   3409   3411   3417   3424
SWREG   000176             683#  1161   1186   1191   2759   2776   2783
SWREGL  013016            2771   2781#
```

DZDUSA.M11 13-OCT-76 08:39 CROSS REFERENCE TABLE -- USER SYMBOLS

```
SW0    = 000001            635#
SW00   = 000001            625#    635     1207
SW01   = 000002            624#    634     1326
SW02   = 000004            623#    633
SW03   = 000010            622#    632
SW04   = 000020            621#    631
SW05   = 000040            620#    630
SW06   = 000100            619#    629
SW07   = 000200            618#    628
SW08   = 000400            617#    627
SW09   = 001000            616#    626     2790
SW1    = 000002            634#
SW10   = 002000            615#
SW11   = 004000            614#
SW12   = 010000            613#
SW13   = 020000            612#
SW14   = 040000            611#
SW15   = 100000            610#
SW2    = 000004            633#
SW3    = 000010            632#
SW4    = 000020            631#
SW5    = 000040            630#
SW6    = 000100            629#
SW7    = 000200            628#
SW8    = 000400            627#
SW9    = 001000            626#
SYNCNO  001146             719#   1296*   1300*
SYNEXT= 020000             787#    980#   1352   1359   1404   1411   1456   1463   1508   1515   1560   1567   1612
                          1619   2228   2235   2266   2273   2449   2456   2482   2489   2515   2522   2548   2555
SYNINT= 030000             786#    979#   2581   2588   2617   2624   2653   2660
SYNSCH= 000020             772#    965#   1360   1412   1464   1516   1568   1620   1672   1711   1750   1789   1827
                          1868   1909   1950   1988   2029   2070   2111   2148   2192   2236   2274   2313   2349
                          2385   2421   2457   2490   2523   2556   2589   2625   2661
SYSTST= 014000             812#   1005#
TBITVE= 000014             663#
TEMP    016172            2994   3249*   3250*   3363#
TKVEC  = 000060            675#
TPVEC  = 000064            676#
TRAPVE= 000034             674#   1140*   1141*
TRPREG  017350            3577#
TRTVEC= 000014             669#
TST1    003362            1331   1337    1350#   2750   2751   3388
TST10   005216            1701#
TST11   005356            1740#
TST12   005516            1779#
TST13   005656            1817#
TST14   006026            1858#
TST15   006176            1899#
TST16   006346            1940#
TST17   006516            1978#
TST2    003574            1402#
TST20   006666            2019#
TST21   007036            2060#
TST22   007206            2101#
TST23   007356            2138#
TST24   007542            2182#
```

```
TST25   007726          2226#
TST26   010062          2264#
TST27   010216          2303#
TST3    004006          1454#
TST30   010366          2339#
TST31   010536          2375#
TST32   010706          2411#
TST33   011056          2447#
TST34   011212          2480#
TST35   011346          2513#
TST36   011502          2546#
TST37   011636          2579#
TST4    004220          1506#
TST40   012006          2615#
TST41   012156          2651#
TST5    004432          1558#
TST6    004644          1610#
TST7    005056          1662#
TTST    016314          3391#
TXCSR   001726          1046#  1351#  1353#  1356#  1362#  1363#  1403#  1405#  1408#  1414#  1415#  1455#  1457#
                        1460#  1466#  1467#  1507#  1509#  1512#  1518#  1519#  1559#  1561#  1564#  1570#  1571#
                        1611#  1613#  1616#  1622#  1623#  1663#  1665#  1668#  1674#  1675#  1677#  1678#  1702#
                        1704#  1707#  1713#  1714#  1716#  1717#  1741#  1743#  1746#  1752#  1753#  1755#  1756#
                        1780#  1782#  1785#  1791#  1792#  1794#  1795#  1818#  1820#  1823#  1829#  1830#  1832#
                        1833#  1859#  1861#  1864#  1870#  1871#  1873#  1874#  1900#  1902#  1905#  1911#  1912#
                        1914#  1915#  1941#  1943#  1946#  1952#  1953#  1955#  1956#  1979#  1981#  1984#  1990#
                        1991#  1993#  1994#  2020#  2022#  2025#  2031#  2032#  2034#  2035#  2061#  2063#  2066#
                        2072#  2073#  2075#  2076#  2102#  2104#  2107#  2113#  2114#  2116#  2117#  2139#  2141#
                        2144#  2150#  2151#  2153#  2154#  2183#  2185#  2188#  2194#  2195#  2197#  2198#  2227#
                        2229#  2232#  2238#  2239#  2265#  2267#  2270#  2276#  2277#  2304#  2306#  2309#  2315#
                        2316#  2318#  2319#  2340#  2342#  2345#  2351#  2352#  2354#  2355#  2376#  2378#  2381#
                        2387#  2388#  2390#  2391#  2412#  2414#  2417#  2423#  2424#  2426#  2427#  2448#  2450#
                        2453#  2459#  2460#  2481#  2483#  2486#  2492#  2493#  2514#  2516#  2519#  2525#  2526#
                        2547#  2549#  2552#  2558#  2559#  2580#  2582#  2585#  2591#  2592#  2594#  2595#  2616#
                        2618#  2621#  2627#  2628#  2630#  2631#  2652#  2654#  2657#  2663#  2664#  2666#  2667#
                        3518#  3531#  3537#  3538#  3539#
TXDBUF  001732          1048#  3522#
TXDONE= 000200           802#   995#
TXINTE= 000100           803#   996#
TYPE   = 104401         1199   1338   2688   2691   2696   2767   2769   2837   2880   2897   2983   3014   3061
                        3069   3105   3122   3124   3127   3129   3133   3140   3204   3254   3476#
TYPOC  = 104402         3113   3137   3477#
TYPON  = 104404         3479#
TYPOS  = 104403         3478#
USER   = 000000          809#  1002#
VOID   = 000001          776#   969#
WRDCNT  014112          2987*  3016*  3019#
ZERO    012410          2702#
$APTHD  002136          1105   1111#
$ASTAT= ****** U         534
$ATYC   000024R          534#
$ATY1   000000R          534#
$ATY3   000006R          534#  2822
$ATY4   000016R          534#  3074
$AUTOB  001434           837#
$BASE   001602           915#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
$BDADR  001422            832#
$BDDAT  001426            834#
$BELL   001516            866#   3061    3089
$CDW1   001606            917#
$CDW2   001610            918#
$CHARC  013326           2839*   2849*   2856    2865*   2870#
$CKSWR= ****** U         3482
$CMTAG  001400            820#   1129    1130    1138    1144    1145
$CM1  = 000006            852#    853#    854#    855#    856#    857#    858#
$CM2  = 000014            852#    853#    854#    855#    856#    857#    858#
$CM3  = 000006            850#    852
$CM4  = 000006            858#    859#    860#    861#    862#    863#    864#
$CPUOP  001554            889#
$CRLF   001523            868#   2838    2873    3069    3089    3105    3124    3129    3133
$DDW0   001612            919#
$DDW1   001614            920#
$DDW10  001636            929#
$DDW11  001640            930#
$DDW12  001642            931#
$DDW13  001644            932#
$DDW14  001646            933#
$DDW15  001650            934#
$DDW2   001616            921#
$DDW3   001620            922#
$DDW4   001622            923#
$DDW5   001624            924#
$DDW6   001626            925#
$DDW7   001630            926#
$DDW8   001632            927#
$DDW9   001634            928#
$DEVCT  001536            880#
$DEVM   001604            916#
$E    = 000002            532#
$ENDAD  012646           1092    2744#
$ENV    001546            534     885#   1201    2817    2878    3071
$ENVM   001547            534     886#   1166    2819    2824
$ERFLG  001403            823#   1175*   2736*   3056*   3089    3256*   3372    3413    3415    3421*   3445
$ERMAX  001415            829#   1146*   3415    3438*   3445
$ERROR  014166           1138    3055#
$ERRPC  001416            830#   1078    1081    1177*   2735*   3063*   3064*   3065    3089    3111    3257*   3387*
$ERRTB  001652           1020#   3119
$ERRTY  014406           3104#
$ERTTL  001412            827#   1176*   3062*   3089
$ESCAP  001514            865#   1145*   3084    3086    3089    3437*
$ETABL  001546            884#
$ETEND  001652            937#   1117
$FATAL  001530            534#    877#
$FFLG   000244R           534#*
$FILLC  001456            848#   2842    2873
$FILLS  001455            847#   2873
$GDADR  001420            831#
$GDDAT  001424            833#
$GTSWR= ****** U         3481
$HIBTS  002136           1112#
$ICNT   001404            824#   3428*   3429    3431*   3444
$INTAG  001435            838#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
$ITEMB  001414            828#   3065*  3073   3089   3108
$LF     001524            869#   2873   3089
$LFLG   000243R           534#*
$LPADR  001406            825#   1147*  2750*  3419*  3435*  3440   3444
$LPERR  001410            826#   1148*  3083   3419   3436*  3444
$MADR1  001560            902#
$MADR2  001564            906#
$MADR3  001570            909#
$MADR4  001574            912#
$MAIL   001526            875#   1113   1117   1165   2817   3071   3434
$MAMS1  001556            896#
$MAMS2  001562            904#
$MAMS3  001566            907#
$MAMS4  001572            910#
$MBADR  002140           1113#
$MFLG   000242R           534#*
$MSGAD  001542            534*    882#
$MSGLG  001544            534*    883#
$MSGTY  001526            534*    876#
$MTYP1  001557            897#
$MTYP2  001563            905#
$MTYP3  001567            908#
$MTYP4  001573            911#
$MXCNT  016612           3432    3444#
$N    = 000000            532#   2680#
$NULL   001454            846#   2844   2873
$NWTST= 000000           1349#   1401#  1453#  1505#  1557#  1609#  1661#  1700#  1739#  1778#  1816#  1857#  1898#
                         1939#   1977#  2018#  2059#  2100#  2137#  2181#  2225#  2263#  2302#  2338#  2374#  2410#
                         2446#   2479#  2512#  2545#  2578#  2614#  2650#
$OCNT   014764           3176*   3205*  3218#
$OMODE  014766           3171*   3175*  3180   3183*  3194*  3220#
$OVER   016572           3393    3396   3412   3420   3430   3439#
$PASS   001534            879#   1165*  2739*  3426   3445
$PASTM  002144           1115#
$PWRDN  014770           1142    3224#
$QUES   001522            867#   2873   3089
$ROCHR= ****** U         3482
$RODEC= ****** U         3482
$ROLIN= ****** U         3482
$ROOCT= ****** U         3482
$REGAD  001460            850#
$REG0   001462            852#   2968*  2973
$REG1   001464            853#   2967*  2974
$REG2   001466            854#   2966*  2975
$REG3   001470            855#   2965*  2976
$REG4   001472            856#   2964*  2977
$REG5   001474            857#   2963*  2978
$R2A  = ****** U         3482
$SAVRE= ****** U         3482
$SCOPE  016276           1136    3381#
$SETUP= 000017           1118#   1135   1136   1138   1140   1142   1144   1145   1147   3056   3081   3088   3382
$STUP = 177777           1118#
$SVADD  016536           3404    3433#
$SVPC = 002136           1090#   1095
$SWR  = 177400            523#    864    865    866   1144   1145   1147   1148   1351   1403   1455   1507   1559
                         1611    1663   1702   1741   1780   1818   1859   1900   1941   1979   2020   2061   2102
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
                        2139    2183    2227    2265    2304    2340    2376    2412    2448    2481    2514    2547    2580
                        2616    2652    3047    3048    3049    3050    3051    3059    3066    3078    3081    3089    3373
                        3374    3375    3376    3377    3395    3407    3409    3410    3413    3414    3415    3422    3423
                        3424    3436    3439    3444
$SWREG  001550           887#   1168
$SWRMK= 000000          3377    3378    3411
$TESTN  001532           878#   3434*
$TIMES  001512           864#   1144*   3422*   3429    3432*   3444
$TKB    001446           843#   2763    2886    2894    3394
$TKS    001444           842#   2761    2884    3392
$TMP0   001476           858#
$TMP1   001500           859#   1367*   1379*   1383*   1419*   1431*   1435*   1471*   1483*   1487*   1523*   1535*   1539*
                        1575*   1587*   1591*   1627*   1639*   1643*   1680*   1719*   1758*   1797*   1837*   1878*   1919*
                        1960*   1998*   2039*   2080*   2121*   2159*   2162*   2163*   2203*   2207*   2244*   2282*
                        2322*   2358*   2394*   2430*   2463*   2496*   2529*   2562*   2598*   2634*   2670*   3533*   3544
                        3554*   3556*   3561    3571*   3573*
$TMP2   001502           860#   3532*   3534*   3535*   3536*   3537    3544*   3548*   3561*   3565*
$TMP3   001504           861#   3545*   3549*   3552*   3562*   3566*   3569*
$TMP4   001506           862#
$TMP5   001510           863#
$TN    = 000042          534#   1349    1351#   1401    1403#   1453    1455#   1505    1507#   1557    1559#   1609    1611#
                        1661    1663#   1700    1702#   1739    1741#   1778    1780#   1816    1818#   1857    1859#   1898
                        1900#   1939    1941#   1977    1979#   2018    2020#   2059    2061#   2100    2102#   2137    2139#
                        2181    2183#   2225    2227#   2263    2265#   2302    2304#   2338    2340#   2374    2376#   2410
                        2412#   2446    2448#   2479    2481#   2512    2514#   2545    2547#   2578    2580#   2614    2616#
                        2650    2652#
$TPB    001452           845#   2862*   2873    2894*
$TPFLG  001457           849#   2811    2873
$TPS    001450           844#   2860    2873    2892
$TRAP   016614          1140    3453#
$TRAP2  016636          3464#   3475
$TRP   = 000015         3468#   3477#   3478#   3479#   3480#   3482    3483#   3484#   3485#   3486#   3487#   3488#   3489#
                        3490#
$TRPAD  016650          3458    3475#
$TSTM   002142          1114#
$TSTNM  001402           822#   1178*   2734*   3058    3089    3095    3372    3411    3433*   3434    3439    3445
$TYPBN= ****** U        3480
$TYPDS= ****** U        3480
$TYPE   013050           534    2811#   3468    3476
$TYPEC  013262          2841    2848    2855    2860#   2861
$TYPEX  013330          2866    2868    2871#
$TYPOC  014566          3174#   3477
$TYPON  014602          3173    3176#   3479
$TYPOS  014542          3169#   3478
$UNIT   001540           881#
$UNITM  002146          1116#
$USWR   001552           888#   1203
$VECT1  001576           913#
$VECT2  001600           914#
$XTSTR  016342          3389    3398#
$OFILL  014765          3170*   3174*   3184    3219#
$40CAT= ****** U        3068    3395
.      = 017356          534#    568#    681#    684#    689#    744#    745#    819#    870    1076#   1090    1091#   1093#
                        1095#   1101    1102#   1104#   1106#   1133    1147    1148    1370    1374    1389    1422    1426
                        1441    1474    1478    1493    1526    1530    1545    1578    1582    1597    1630    1634    1649
                        1687    1726    1765    1804    1840    1844    1881    1885    1922    1926    1963    1967    2001
```

```
DZDUS-A MACY11 27(1006)  03-FEB-77  07:52  PAGE 84
DZDUSA.M11    13-OCT-76 08:39          CROSS REFERENCE TABLE -- USER SYMBOLS

                      2005    2042    2046    2083    2087    2124    2128    2168    2212    2250    2288    2326    2362
                      2398    2434    2467    2500    2533    2566    2602    2638    2674    2873    3089    3143#   3235
                      3251    3362#   3364#   3366#   3444    3445
.BEGIN  003310        1195    1206    1209    1324#
.CONVR  013742        2983#   3488
.EOP    012326        2686#
.INSTE  013440        2897#   3484
.INSTR  013332        2876#   3483
.INST1  013352        2880#   2889    2899
.MSG    013354        2763*   2764*   2765    2875*   2881#
.PARAM  013450        2904#   3485
.PFAIL  014770        1172    3225#   3246
.RESOS  013710        2973#   3487
.SAVOS  013650        2959#   3486
.SCOPE  016276        3385#
.SCOP1  013026        2789#   3482
.SETFL  014120        3029#   3040    3489
.START  002152         685    1127#   1179    2700
.$ASTA= ****** U       534
.$X   = 002136        1101#   1106
```

CROSS REFERENCE TABLE -- MACRO NAMES

```
COMMEN    678#
ENDCOM    678#
ERROR     572#  1371  1375  1390  1423  1427  1442  1475  1479  1494  1527  1531  1546  1579  1583
         1598  1631  1635  1650  1688  1727  1766  1805  1841  1845  1882  1886  1923  1927  1964
         1968  2002  2006  2043  2047  2084  2088  2125  2129  2169  2213  2251  2289  2327  2363
         2399  2435  2468  2501  2534  2567  2603  2639  2675
ESCAPE    678#
GETPRI    678#
GETSWR    678#
MULT      678#
NEWTST    678#  1349  1401  1453  1505  1557  1609  1661  1700  1739  1778  1816  1857  1898  1939
         1977  2018  2059  2100  2137  2181  2225  2263  2302  2338  2374  2410  2446  2479  2512
         2545  2578  2614  2650
POP        534   678#
PRGEND     556#  2679
PRGFRT     535#   557
PUSH       534   678#
PUSSYF     553#
REPORT     523#  678#
RSETUP     540#  1351  1403  1455  1507  1559  1611  1663  1702  1741  1780  1818  1859  1900  1941
         1979  2020  2061  2102  2139  2183  2227  2265  2304  2340  2376  2412  2448  2481  2514
         2547  2580  2616  2652
SC         532#  3382
SCOPE      573#  1350  1402  1454  1506  1558  1610  1662  1701  1740  1779  1817  1858  1899  1940
         1978  2019  2060  2101  2138  2182  2226  2264  2303  2339  2375  2411  2447  2480  2513
         2546  2579  2615  2651  2686
SC1        532#  3441
SETPRI     678#
SETTRA     523#  3468#  3477  3478  3479  3482  3483  3484  3485  3486  3487  3488  3489
SETUP      678#  1127
SKIP       678#
SLASH      678#
SPACE      678#
STARS      534    678#   815   870   873  1088  1098  1100  1107  1349  1401  1453  1505  1557  1609
         1661  1700  1739  1778  1816  1857  1898  1939  1977  2018  2059  2100  2137  2181  2225
         2263  2302  2338  2374  2410  2446  2479  2512  2545  2578  2614  2650  2796  3043  3099
         3146  3369  3447
SWRSU      678#  1149#
TRMTRP    3468#
TSETUP     541#
TYP8IN     678#
TYPDEC     678#
TYPNAM     678#
TYPNUM     678#
TYPOCS     678#
TYPOCT     678#  3111  3135
TYPTXT     678#
$BEGIN     532#  1321
$BINAR     552#
$BUFFE     534#  3358
$CABLE     554#
$CATCH     526#   568
$CK        533#  2752
$CLRVE     532#  1201
$CONVR     534#  2980
$DNA       550#
```

CROSS REFERENCE TABLE -- MACRO NAMES

```
$EOP      533#  2679
$GETFL    532#  1240  1305  1309  1313  1317
$GETPA    532#  1219  1230  1250  1276  1328
$GETSY    532#  1287
$HEADE    523#   557
$INSTR    534#  2873
$ISOB     548#  2131  2175  2219  2257
$MATCH    545#
$MRR      539#
$MRRW     537#
$MRW      538#
$MSG      534#  3262
$NEED     555#  3089
$PARAM    534#  2901
$PFAIL    534#  3221
$POKE     543#
$POKER    544#  1361  1413  1465  1517  1569  1621  1676  1715  1754  1793  1831  1872  1913  1954
          1992  2033  2074  2115  2152  2196  2237  2275  2317  2353  2389  2425  2458  2491  2524
          2557  2593  2629  2665
$RCNET    555#
$RECAC    543#
$REG      534#  2956
$RESET    536#  1351  1353  1403  1405  1455  1457  1507  1509  1559  1561  1611  1613  1663  1665
          1702  1704  1741  1743  1780  1782  1818  1820  1859  1861  1900  1902  1941  1943  1979
          1981  2020  2022  2061  2063  2102  2104  2139  2141  2183  2185  2227  2229  2265  2267
          2304  2306  2340  2342  2376  2378  2412  2414  2448  2450  2481  2483  2514  2516  2547
          2549  2580  2582  2616  2618  2652  2654
$RXACT    542#
$SCOPE    532#  3367
$SCOPI    533#  2786
$SETFL    532#  3023
$SETVE    527#   678
$START    532#  1119
$STRIP    549#  2295  2331  2367  2403  2439  2472  2505  2538  2571  2607  2643
$SYMBO    523#   746   939
$SYNCR    545#  1672  1711  1750  1789  1827  1868  1909  1950  1988  2029  2070  2111  2148  2192
          2313  2349  2385  2421  2589  2625  2661
$TRPAR    551#
$TSTNO    532#  1349  1401  1453  1505  1557  1609  1661  1700  1739  1778  1816  1857  1898  1939
          1977  2018  2059  2100  2137  2181  2225  2263  2302  2338  2374  2410  2446  2479  2512
          2545  2578  2614  2650
$UNIBU    535#
$VARIA    530#   688
$WORDF    546#  1653  1692  1731  1770
$WORDO    545#  1341  1393  1445  1497  1549  1601
$WORDP    547#  1809  1850  1891  1932  1970  2011  2052  2093
$$CMRE    813#   852   853   854   855   856   857
$$CMTM    813#   858   859   860   861   862   863
$$ESCA    678#
$$NEXT    678#  1349  1401  1453  1505  1557  1609  1661  1700  1739  1778  1816  1857  1898  1939
          1977  2018  2059  2100  2137  2181  2225  2263  2302  2338  2374  2410  2446  2479  2512
          2545  2578  2614  2650
$$SET     3468# 3477  3478  3479  3482  3483  3484  3485  3486  3487  3488  3489
$$SETM    1165#
$$SKIP     678#
.EQUAT     523#  568
```

```
DZDUS-A MACY11 27(1006)  03-FEB-77  07:52  PAGE 88
DZDUSA.M11    13-OCT-76 08:39          CROSS REFERENCE TABLE -- MACRO NAMES

.MEADE    523#
.SETUP    523#   1118
.$ACT1    523#   1086
.$APTB    523#    871#
.$APTH    523#   1096
.$APTY    523#    534
.$CATC    523#
.$CMTA    523#    813
.$EOP     523#
.$ERRO    523#   3041
.$ERRT    523#   3097
.$POWE    523#
.$SCOP    523#   3367
.$TRAP    523#   3445
.$TYPE    523#   2794
.$TYPO    523#   3144


. ABS.  017356     000


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DZDUSA,DZDUSA.SEQ/SOL/CRF←DZDUS1/EQ:RUNC,DZDUS2,DZDUSA
 RUN-TIME: 18 12 1 SECONDS
 RUN-TIME RATIO: 118/33=3.5
 CORE USED:  31K  (62 PAGES)
```