DL11-W

DL11-W DIAGNOSTIC

MD-11-DZDLD-B

EP-DZDLD-B-DL-A

FICHE 1 OF 1

AUG 1977

digital

MADE IN USA

IDENTIFICATION

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

PRODUCT NAME: DL11-W DIAGNOSTIC

DATE CREATED: JUNE, 1977

MAINTAINER: DIAGNOSTIC ENGINEERING (CC301)

AUTHOR: DAN CASALETTO

1.0 GENERAL INFORMATION

1.1 ABSTRACT

THIS DIAGNOSTIC IS A LOGIC TEST TO VERIFY THE OPERATION OF THE DL11-W SERIAL LINE/ REAL TIME CLOCK INTERFACE. THE PROGRAM WILL RUN WITHOUT ANY SPECIAL TEST FIXTURES BY DEFAULT. HOWEVER, A WRAP CABLE CAN BE USED AND TESTED BY OPTION (BIT7 OF SWR).

THIS TEST OPERATES ON THE CONSOLE DL11-W SERIAL LINE AND CLOCK INTERFACES AS WELL AS UP TO FIFTEEN(15) ADDITIONAL IDENTICALLY CONFIGURED DL11-W SERIAL LINE INTERFACES. THE DEFAULT ADDRESSES ARE:

A. CONSOLE - 177560 SERIAL LINE
177546 CLOCK

B. OTHER SERIAL LINE - 776500 FIRST SERIAL LINE ADDRESS OF 15 CONSECUTIVE SERIAL LINE ADDRESSES

THE PROGRAM IS DESIGNED TO RUN ON ANY PDP-11 WITH 8K OF MEMORY AND A DL11-W MODULE. IT CAN BE RUN UNDER XXDP, APT, AND ACT MONITORS, AND ON PROCESSORS WITH NO HARDWARE SWITCH REGISTER, SOFTWARE SWITCH REGISTER = LOCATION 176

POWER FAILURE IS SUPPORTED FOR SYSTEMS WITH CORE MEMORY.

NOTE: THIS DIAGNOSTIC WITH THE SWR = 000020
(CLOCK TESTS ONLY) SHOULD BE USED ON SWITCHLESS CPU'S TO TEST KW-11L LINE CLOCK MODULES.

1.2 SYSTEM REQUIREMENTS

1.2.1 EQUIPMENT

STANDARD 11 FAMILY COMPUTER WITH A CONSOLE OUTPUT DEVICE AND 8K OF MEMORY.

1.2.2 STORAGE

THE PROGRAM USES MEMORY FROM 00000 TO 22372

1.3 ASSUMPTIONS

A. IF THE UNIT UNDER TEST (UUT) IS THE CONSOLE , THE PROGRAM WILL ASSUME THE REAL TIME CLOCK (RTC) IS ENABLED AND WILL TEST IT UNLESS THE TESTS ARE DISABLED BY BIT6 OF THE SWR.

B. IF THE UUT IS NOT THE CONSOLE, THE RTC IS NOT TESTED FOR THAT DEVICE.

C. THE PROGRAM WILL ASSUME THE ERROR FLAG BITS AND THE BREAK FUNCTION OF THE DL11-W ARE DISABLED AND WILL NOT TEST THESE FUNCTIONS UNLESS ENABLED BY BIT10 (FOR ERROR FLAGS) AND BIT8 (FOR BREAK) OF THE SWR. THE DEFAULT CHARACTER SIZE IS 8 BITS (SEE PARA 2.3.2).

2.0 OPERATING INSTRUCTIONS

2.1 LOADING PROCEDURE

USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED TAPES.

2.2 STARTING PROCEDURE

LOAD THE SWITCH REGISTER WITH SETTING
(SOFTWARE SWITCH REGISTER LOCATION = 176)

A. START AT 200.
AFTER CHECKING THE TRANSMITTER, THE PROGRAM WILL PRINT ITS IDENTIFICATION AND REPORT THE NUMBER OF DEVICES UNDER TEST (NUMBER IS OCTAL).
"END PASS" IS PRINTED AFTER A FULL PASS HAS BEEN MADE ON ALL DEVICES UNDER TEST.

B. START AT 204.
THE "ECHO" TEST WILL BE EXECUTED. AN "*" IS PRINTED AT THE BEGINNING OF THE TEST. THE ECHO TEST READS A CHARACTER FROM THE TERMINAL,WRITES THAT CHARACTER TO THE TERMINAL AND REPORTS ANY ERROR FLAGS SET IN THE RECEIVER BUFFER.
A CONTROL-C HALTS THE TEST AND PRINTS "STOP" AT THE TERMINAL
CONTINUING RESTARTS THE ECHO TEST.

C. START AT 210.
THE TERMINAL OUTPUT TEST WILL BE EXECUTED. DEPRESSING ANY CHARACTER AT THE TERMINAL, HALTS THE TEST. CONTINUING RESTARTS THE TEST.
THE TEST OUTPUTS 32 CHARACTERS ON A LINE AND REPEATS THE PATTERN EVERY THREE LINES. THE PATTERN IS AS FOLLOWS
(OCTAL CODE 040 --> 377):

```
                                          (OCTAL CODE)
 !"#$%&'()*+,-./0123456789:;<=>?          (040 --> 077)
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑←          (100 --> 137)
`abcdefghijklmnopqrstuvwxyz{|             (140 --> 177) [LOWER CASE ALPHA]
```

THIS BOTTOM LINE COULD BE THE FOLLOWING IF THE TERMINAL DOES NOT HAVE LOWER CASE:

```
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\             [UPPER CASE ALPHA]
```

2.3 OPERATING PROCEDURE

2.3.1 OPERATIONAL SWITCH SETTINGS

NOTE: IF NO HARDWARE SWITCH REGISTER IS AVAILABLE THE PROGRAM WILL AUTOMATICALLY USE THE CONTENTS OF LOCATION 176 AS THE SOFTWARE SWITCH REGISTER. THE USER SHOULD SET THIS LOCATION BEFORE STARTING THE PROGRAM. IF A HARDWARE SWITCH IS AVAILABLE AND A SOFTWARE SWR IS DESIRED, PUT ALL SWITCHES UP.

BIT15 - HALT ON ERROR
BIT14 - SCOPE LOOP
BIT13 - INHIBIT ERROR TYPEOUT
BIT12 - UNUSED
BIT11 - UNUSED
BIT10 - ENABLE ERROR FLAGS TESTS
BIT09 - LOOP ON ERROR
BIT08 - ENABLE BREAK FUNCTION TESTS
BIT07 - ENABLE DATA TEST WITH WRAP CABLE
BIT06 - INHIBIT RTC TESTS (ALLOW ONLY SLU TESTS)
BIT05 - ALLOW MANUAL SETTING OF "$DEVM" (DEVICE MAP)
BIT04 - INHIBIT SLU TESTS (ALLOW ONLY LINE CLOCK TESTS)

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G (CNTL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS. BECAUSE THIS DIAGNOSTIC USES THE MAINTENANCE BIT OF THE SERIAL LINE, THE CONTROL-G SHOULD BE ISSUED DURING PROGRAM TYPEOUTS AT THAT TIME THE MAINTENANCE BIT IS SURE TO BE CLEAR.

IF A CONTROL-G IS DETECTED, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

```
(IE)    SWR=XXXXXX  NEW=
```

POSSIBLE RESPONSES ARE:

1.	<CR>	IF NO CHANGES ARE TO BE MADE
2.	6 DIGITS 0-7	TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE ;LAST DIGIT FOLLOWED BY <CR>.
3.	↑U	TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.

2.3.2 SETTING BITS PER CHARACTER

THIS PROGRAM DEFAULTS TO TESTING 8 BITS PER CHARACTER. IF THE SERIAL LINE IS SET FOR 5-->7 BITS PER CHARACTER, SET THE MEMORY LOCATION "$USWR" AS FOLLOWS:

CHAR. SIZE (# OF BITS)	"$USWR" CONTENTS (BINARY)	(OCTAL)
8	100000000	400
7	010000000	200
6	001000000	100
5	000100000	40

"$USWR" IS USED IN THE DATA PATH TESTS TO LIMIT THE BINARY COUNT TEST PATTERN TO THE NUMBER OF BITS SELECTED ON THE SERIAL LINE.

2.3.3 RUNNING UNDER APT

THE APT MAILBOX IS LOCATED AT LOCATION 500. TO ALLOW ADDITIONAL SERIAL LINE VECTOR ASSIGNMENTS TO THE 400 AREA OF MEMORY.

THE EXECUTION TIMES PROVIDED ARE FOR EXECUTION WITH AN 11/34 PROCESSOR, CORE MEMORY, AND 110 BAUD.

2.3.4 RUN WITH ALTERNATE CONSOLE ADDRESS

TO USE A CONSOLE ADDRESS OTHER THAN 177560, OR VECTOR OTHER THAN 60, THE OPERATOR MUST SUPPLY THE PROGRAM WITH THE CORRECT ADDRESSES BY INSERTING THEM AT THE TAG LABELED "CRCSR":

```
CRCSR:  ADDRESS OF RECEIVER RCSR
CRBUF:  ADDRESS OF RECEIVER BUFFER
CTCSR:  ADDRESS OF TRANSMITTER CSR
CTBUF:  ADDRESS OF TRANSMITTER BUFFER
CRVECT: ADDRESS OF RECEIVER VECTOR
CRPSW:  ADDRESS OF ASSOCIAED PSW
CTVECT: ADDRESS OF TRANSMITTER VECTOR
CTPSW:  ADDRESS OF ASSOCIATED PSW
```

2.3.5 TESTING ADDITIONAL SERIAL LINES

THIS PROGRAM WILL SUPPORT TESTING OF MULTIPLE SLU'S. IT REQUIRES THE ADDRESS OF THE FIRST ADDITIONAL RCSR (STORED AT "$BASE") AND ITS INTERRUPT VECTOR (STORED AT "$VECT1"); AND WILL BE ABLE TO ADDRESS ANY SLU STARTING AT THE SPECIFIED BASE ADDRESS UP TO 15 CONSECUTIVE DEVICES.

```
EXAMPLE:        $BASE:  776500
                $VECT1: 300
```

THE PROGRAM WILL BE ABLE TO TEST THE CONSOLE PLUS ANY ADDITIONAL DL11-W SLU'S WITHIN THE RANGE 776500 --> 776660

$BASE AND $VECT1 DEFAULT TO 776500 AND 300 RESPECTIVELY.

THE PROGRAM ASSOCIATES UNIT NUMBERS TO DEVICES AS FOLLOWS:
(NUMBERS IN PARENTHESIS ARE OCTAL)

```
UNIT# 0 -->  CONSOLE [ADDRESS STORED AT "CRCSR"]
UNIT# 1 -->  BASE ADDRESS STORED AT "$BASE"
             ASSOCIATED BASE VECTOR STORED AT"$VECT1"
UNIT# 2 -->  BASE ADDRESS + (10)
             BASE VECTOR + (10)
UNIT# 3 -->  BASE ADDRESS + (20)
             BASE VECTOR + (20)
UNIT# 4 -->  BASE ADDRESS + (30)
             BASE VECTOR + (30)
               :
               :
               :
               :
               :
               V
UNIT#15 -->  BASE ADDRESS + (160)
             BASE VECTOR + (160)
```

STARTING AT LOCATION 200 AND HAVING BITS OF SWR CLEAR, THE PROGRAM WILL SELF
SIZE THE NUMBER OF DEVICES (STARTING AT THE BAS ADDRESS) AND
STORE A BIT MAP AT "$DEVM" (DEVICE MAP) TO INDICATE WHICH UNIT NUMBERS
ARE PRESENT AND WILL BE TESTED:

```
-------------------------------------------------
! UNIT ! UNIT ! .....! UNIT ! UNIT ! CONSOLE!
!  15  !  14  ! .....!   2  !   1  !         !
-------------------------------------------------
```

A BIT MAP CAN BE ENTERED AT "$DEVM" PRIOR TO STARTING THE PROGRAM
SETTING BITS OF THE SWR INHIBITS THE SELF-SIZING AND DEVICE MAP
GENERATION,AND USES THE VALUE STORED BY THE OPERATOR.

EXAMPLE:

```
SWR  =  000040          [BINARY  0 000 000 000 100 000]
$BASE:  776500
$VECTI: 300

$DEVM:  13      [BINARY - 0 000 000 000 001 011]

THE PROGRAM WILL TEST -
        UNIT# 0 = CONSOLE
        UNIT# 1 = 776500 ; 300
        UNIT# 3 = 776520 ; 320
```

2.4 EXECUTION TIMES - (110 BAUD)

LONGEST SUBTEST TIME = 50 SECONDS
PASS TIME = 60 SECONDS
ADDITIONAL DEVICES = 55 SECONDS/DEVICE

3.0 ERROR REPORTING

IF A ROUTINE FAILS AND THE INHIBIT ERROR TYPEOUT (BIT13) OF THE SWR IS NOT SET, A PRINTOUT RESULTS IN THE FORM:

"(SOME ASCII MESSAGE)"
TEST# ERR PC RCSR [ANY APPLICABLE DAT HEADINGS]
XXXXXX XXXXXX XXXXXX [ANY APPLICABLE DATA]

NOTE: "RCSR" IS DEPENDENT ON THE FAILURE
& THEREFORE COULD BE TCSR,RBUF,TBUF,OR LKS

WHERE "XXXXXX" IS AN OCTAL NUMBER.
THIS ERROR PRINTOUT OCCURS PROVIDED THE ERROR THAT EXISTS WOULD NOT HINDER THE TYPEOUT. IN CASES WHERE IT IS NOT POSSIBLE TO PRINT AN ERROR MESSAGE (I.E. FATAL CONSOLE TRANSMITTER FAILURES), A HALT OCCURS AND THE PC CAN BE EXAMINED BY THE OPERATOR TO FIND THE ERROR INFORMATION IN THE PROGRAM LISTING.

NOTE: FOR SOFTWARE SWITCH OPERATION, THE SWITCH REGISTER CAN BE CHANGED BY TYPING A CONTROL-G AT THE CONSOLE DURING ERROR PRINTOUTS. AFTER CONTINUING FROM THE ERROR HALT THE OLD SWR CONTENTS IS DISPLAYED AND THE NEW CONTENTS CAN BE ENTERED.
IF ERROR HALTS ARE DISABLED, THE CONTROL-G RESPONSE OCCURS IMMEDIATELY FOLLOWING THE TYPEOUT.

4.0 SUBROUTINE ABSTRACTS

4.1 TRAPCATCHER

THIS IS A SERIES OF INSTRUCTIONS STARTING AT LOCATION 0 DESIGNED TO
DETECT AND ISOLATE UNEXPECTED TRAPS TO THE TRAP AND INTERRUPT VECTOR
AREA OF MEMORY.

EACH VECTOR ENTRANCE ADDRESS IS LOADED WITH THE ADDRESS OF THE NEXT
LOCATION. THE NEXT LOCATION IS LOADED WITH A BREAK POINT TRAP (000003).
THUS AN ILLEGAL TRAP OR INTERRUPT CAUSES A TRAP THROUGH THE BPT VECTOR(14)
WHICH POINTS TO THE "CATCH" ROUTINE.

THE "CATCH" ROUTINE REPORTS THE PC THAT CAUSED THE ORIGINAL TRAP AND
THE LOCATION OF THE TRAP VECTOR (IF UNDER APT, AN ERROR IS INDICATED TO APT).
AFTER REPORTING THE ERROR THE PROGRAM HALTS. THE PROGRAM MUST BE RESTARTED
AT THIS POINT.

4.2 WRPSW

THIS ROUTINE IS USED TO WRITE THE PSW BY POPPING VALUES FROM THE
STACK. THIS METHOD IS USED TO BE COMPATIBLE WITH ALL 11 FAMILY
PROCESSORS.

4.3 SCOPE

THIS ROUTINE CALL IS PLACED BETWEEN EACH SUBTEST. IT RECORDS THE STARTING
ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED AND UPDATES THE
TEST NUMBER. IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE START OF THE
SUBTEST AT WHICH THE SCOPE LOOP IS REQUESTED.

4.4 ERROR

THIS ROUTINE CALL IS PLACED WHEREEVER AN ERROR REPORT IS DESIRED. THE LOWER BYTE OF THIS CALL IS USED AS THE ERROR NUMBER AND AS A POINTER INTO THE ERROR TABLE. THIS ROUTINE REPORTS ERRORS TO APT USING "$APTYPE" AND TYPES ERROR REPORTS TO THE CONSOLE USING "$ERRTYPE".

4.5 $POWER

THIS ROUTINE SAVES ALL GENERAL REGISTERS DURING POWER-DOWN AND RESTORES THEM AT POWER-UP. IF A POWER FAILURE OCCURS "POWER" IS PRINTED AT THE CONSOLE AFTER POWER IS RESTORED.

4.6 CKSWR

THIS ROUTINE CALL IS USED TO DETECT THE RECEPTION OF A CONTROL-G FROM THE CONSOLE. THE CALL USES "$READ" TO PERFORM THE CONTROL-G SEQUENCE OF DISPLAYING THE CONTENTS OF THE SOFTWARE SWITCH REGISTER AND THE ENTERING THE NEW CONTENTS FROM THE TERMINAL.

```
                        .ENABLE ABS
                        .LIST   ME
                        .NLIST  MC,MD,CND



                        .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
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
```

```
                        001000          SW09=   1000
                        000400          SW08=   400
                        000200          SW07=   200
                        000100          SW06=   100
                        000040          SW05=   40
                        000020          SW04=   20
                        000010          SW03=   10
                        000004          SW02=   4
                        000002          SW01=   2
                        000001          SW00=   1
                                        .EQUIV  SW09,SW9
                                        .EQUIV  SW08,SW8
                                        .EQUIV  SW07,SW7
                                        .EQUIV  SW06,SW6
                                        .EQUIV  SW05,SW5
                                        .EQUIV  SW04,SW4
                                        .EQUIV  SW03,SW3
                                        .EQUIV  SW02,SW2
                                        .EQUIV  SW01,SW1
                                        .EQUIV  SW00,SW0

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

                                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
                        000004          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
                        000010          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
                        000014          TBITVEC=14              ;;"T" BIT
                        000014          TRTVEC= 14              ;;TRACE TRAP
                        000014          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
                        000020          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
```

```
                        000024                  PWRVEC= 24              ;;POWER FAIL
                        000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
                        000034                  TRAPVEC=34              ;;"TRAP" TRAP
                        000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
                        000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
                        000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
                        176500                          ABASE=  176500
                        000300                          AVECT1= 300
                        000400                          AUSWR=  400
                        000001                          $TN=    1
                        161000                          $SWR=   161000
                        000003                          BPT=    000003          ;THIS IS THE COMMAND FOR A TRAP
                                                                                ;   THROUGH 14 (BPT TRAP)


                      * 000000                  .=0
                                                ;;**************************************************************
                                                ;*ALL UNUSED LOCATIONS FROM 4-776 CONTAIN A ".+2,BPT"
                                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS & INTERRUPPTS
                                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS


                        000014                  .=14                    ;THE BPT TRAP VECTOR POINTS TO THE
000014  012254                                  .WORD   CATCH           ;   ILLEGAL TRAP HANDLER "CATCH"
000016  000340                                  .WORD   340

                        000042                  .=      42
000042  000000                                  .WORD   0


                        000174                  .=      174
000174  000000                  DISPREG:.WORD   0
000176  000000                  SWREG:  .WORD   0

                        000200                  .=200
000200  000167  002342                          JMP     START           ;DO INTERFACE TEST
000204  000167  014552                          JMP     ECHO            ;DO ECHO TEST
000210  000167  014766                          JMP     OUTTST          ;DO OUTPUT TEST TO TERMINAL
```

```
                        .=      500
                .SBTTL  ACT11 HOOKS

                ;;*****************************************************************
                ;HOOKS REQUIRED BY ACT11
                        $SVPC=.                 ;SAVE PC
                        .=46
                        $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                        .=52
                        .WORD   0               ;;2)SET LOC.52 TO ZERO
                        .=$SVPC                 ;; RESTORE PC
                .SBTTL  APT PARAMETER BLOCK

                ;;*****************************************************************
                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                ;;*****************************************************************
                        .$X=.   ;;SAVE CURRENT LOCATION
                        .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
                        200     ;;FOR APT START UP
                        .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
                        $APTHDR ;;POINT TO APT HEADER BLOCK
                        .=.$X   ;;RESET LOCATION COUNTER
                ;;*****************************************************************
                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                ;INTERFACE SPEC.

                $APTHD:
                $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
                $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
                $TSTM:  .WORD   50      ;;RUN TIM OF LONGEST TEST
                $PASTM: .WORD   60      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
                $UNITM: .WORD   55      ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
                        .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                .SBTTL  COMMON TAGS

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


                        001000                  .=1000
001000                  001000  $CMTAG:                         ;;START OF COMMON TAGS
001000  000000                          .WORD   0
001002      000         001002  $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001003      000         001003  $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001004  000000          001004  $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001006  000000          001006  $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001010  000000          001010  $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001012  000000          001012  $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001014      000         001014  $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001015      001         001015  $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001016  000000          001016  $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001020  000000          001020  $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001022  000000          001022  $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001024  000000          001024  $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001026  000000          001026  $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001030  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001032  000000                  .WORD   0
001034      000         001034  $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001035      000         001035  $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001036  000000                  .WORD   0
001040  177570          001040  SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001042  177570          001042  DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001044  177560          001044  $TKS:   177560                  ;;TTY KBD STATUS
001046  177562          001046  $TKB:   177562                  ;;TTY KBD BUFFER
001050  177564          001050  $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001052  177566          001052  $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001054      000         001054  $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001055      002         001055  $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001056      012         001056  $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001057      000         001057  $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001060  000000          001060  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001062      077         001062  $QUES:  .ASCII  /?/              ;;QUESTION MARK
001063      015         001063  $CRLF:  .ASCII  <15>             ;;CARRIAGE RETURN
001064  000012          001064  $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .EVEN
001066                  001066  $MAIL:                          ;;APT MAILBOX
001066  000000          001066  $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001070  000000          001070  $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001072  000000          001072  $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001074  000000          001074  $PASS:  .WORD   APASS           ;;PASS COUNT
001076  000000          001076  $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
001100  000000          001100  $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
001102  000000          001102  $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
001104  000000          001104  $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
001106                  001106  $ETABLE:                        ;;APT ENVIRONMENT TABLE
```

```
                                $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
                                $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
                                $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
                                $USWR:  .WORD   AUSWR   ;;USER SWITCHES
                                $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
                                :*                      BITS 15-11=CPU TYPE
                                :*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                :*                              11/70=06,PDQ=07,Q=10
                                :*                      BIT 10=REAL TIME CLOCK
                                :*                      BIT  9=FLOATING POINT PROCESSOR
                                :*                      BIT  8=MEMORY MANAGEMENT
                                $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
                                $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
                                :*                      MEM.TYPE BYTE   --  (HIGH BYTE)
                                :*                              900 NSEC CORE=001
                                :*                              300 NSEC BIPOLAR=002
                                :*                              500 NSEC MOS=003
                                $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                                :*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
                                $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
                                $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
                                $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
                                $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
                                $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
                                $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
                                $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
                                $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
                                $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
                                $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
                                $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
                                $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
                                $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
                                $ETEND:
                                .MEXIT
```

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT


001146                          $ERRTB:

001146                          .$ERRTB:
001146  015266                          EM1                     ;"CAN NOT ACCESS TCSR"
001150  021451                          DH1                     ;"TEST# ERR PC  TCSR"
001152  022214                          DT1                     ;$TESTN,$ERRPC,TCSR
001154  000000                          0

001156  015312                          EM2                     ;"CAN NOT ACCESS TBUF"
001160  021476                          DH2                     ;"TEST# ERR PC  TBUF"
001162  022224                          DT2                     ;$TESTN,$ERRPC,TBUF
001164  000000                          0

001166  015336                          EM3                     ;"TCSR DONE NOT CLEARED WITH TBUF FULL"
001170  021451                          DH1                     ;"TEST# ERR PC  TCSR"
001172  022214                          DT1                     ;$TESTN,$ERRPC,TCSR
001174  000000                          0

001176  015403                          EM4                     ;"TCSR DONE NOT SET"
001200  021451                          DH1                     ;"TEST# ERR PC  TCSR"
001202  022214                          DT1                     ;$TESTN,$ERRPC,TCSR
001204  000000                          0

001206  015425                          EM5                     ;TCSR DONE NOT SET WITH RESET
001210  021451                          DH1                     ;"TEST# ERR PC  TCSR"
001212  022214                          DT1                     ;$TESTN,$ERRPC,TCSR
001214  000000                          0

001216  015462                          EM6                     ;"CAN NOT ACCESS RCSR"
001220  021523                          DH6                     ;"TEST# ERR PC  RCSR"
001222  022234                          DT6                     ;$TESTN,$ERRPC,RCSR
001224  000000                          0

001226  015506                          EM7                     ;"CAN NOT ACCESS RBUF"
001230  021550                          DH7                     ;"TEST# ERR PC  RBUF"
001232  022244                          DT7                     ;$TESTN,$ERRPC,RBUF
001234  000000                          0

001236  015532                          EM10                    ;"CAN NOT ACCESS LKS"
001240  021575                          DH10                    ;"TEST# ERR PC  LKS"
001242  022254                          DT10                    ;$TESTN,$ERRPC,LKS
001244  000000                          0
```

```
                                                EM11            ;"BITO OF TCSR NOT CLEAR AFTER RESET"
001246  015555                                  DH1             ;"TEST# ERR PC  TCSR
001250  021451                                  DT1             ;$TESTN,$ERRPC,TCSR
001252  022214                                  0
001254  000000
                                                EM12            ;"CAN NOT SET BITO OF TCSR"
001256  015620                                  DH1             ;"TEST# ERR PC  TCSR"
001260  021451                                  DT1             ;$TESTN,$ERRPC,TCSR
001262  022214                                  0
001264  000000
                                                EM13            ;"CAN NOT CLEAR BITO OF TCSR"
001266  015651                                  DH1             ;"TEST# ERR PC  TCSR"
001270  021451                                  DT1             ;$TESTN,$ERRPC,TCSR
001272  022214                                  0
001274  000000
                                                EM14            ;"RESET DID NOT CLEAR BITO OF TCSR"
001276  015704                                  DH1             ;"TEST# ERR PC  TCSR"
001300  021451                                  DT1             ;$TESTN,$ERRPC,TCSR
001302  022214                                  0
001304  000000
                                                EM15            ;"BIT2 OF TCSR NOT CLEAR AFTER RESET"
001306  015745                                  DH1             ;"TEST# ERR PC  TCSR"
001310  021451                                  DT1             ;$TESTN,$ERRPC,TCSR
001312  022214                                  0
001314  000000
                                                EM16            ;"CAN NOT SET BIT2 OF TCSR"
001316  016010                                  DH1             ;"TEST# ERR PC  TCSR"
001320  021451                                  DT1             ;$TESTN,$ERRPC,TCSR
001322  022214                                  0
001324  000000
                                                EM17            ;"CAN NOT CLEAR BIT2 OF TCSR"
001326  016041                                  DH1             ;"TEST# ERR PC  TCSR"
001330  021451                                  DT1             ;$TESTN,$ERRPC,TCSR
001332  022214                                  0
001334  000000
                                                EM20            ;"RESET DID NOT CLEAR BIT2 OF TCSR"
001336  016074                                  DH1             ;"TEST# ERR PC  TCSR"
001340  021451                                  DT1             ;$TESTN,$ERRPC,TCSR
001342  022214                                  0
001344  000000
                                                EM21            ;"BIT6 OF TCSR NOT CLEAR AFTER RESET2
001346  016135                                  DH1             ;"TEST# ERR PC  TCSR"
001350  021451                                  DT1             ;$TESTN,$ERRPC,TCSR
001352  022214                                  0
001354  000000
                                                EM22            ;"XMIT INTERRUPT WITH PRIORITY 7"
001356  016201                                  DH1             ;"TEST# ERR PC  TCSR"
001360  021451                                  DT1             ;$TESTN,$ERRPC,TCSR
001362  022214                                  0
001364  000000
                                                EM23            ;"CAN NOT SET BIT6 OF TCSR"
001366  016236                                  DH1             ;"TEST# ERR PC  TCSR"
001370  021451                                  DT1             ;$TESTN,$ERRPC,TCSR
001372  022214                                  0
001374  000000
```

```
  389  001376  016267                    EM24            ;"CAN NOT CLEAR BIT6 OF TCSR"
  390  001400  021451                    DH1             ;"TEST# ERR PC  TCSR"
  391  001402  022214                    DT1             ;$TESTN,$ERRPC,TCSR
  392  001404  000000                    0
  393
  394  001406  016322                    EM25            ;"RESET DID NOT CLEAR BIT6 OF TCSR"
  395  001410  021451                    DH1             ;"TEST# ERR PC  TCSR"
  396  001412  022214                    DT1             ;$TESTN,$ERRPC,TCSR
  397  001414  000000                    0
  398
  399  001416  016363                    EM26            ;"BIT6 OF RCSR NOT CLEAR AFTER RESET"
  400  001420  021523                    DH6             ;"TEST# ERR PC  RCSR"
  401  001422  022234                    DT6             ;$TESTN,$ERRPC,RCSR
  402  001424  000000                    0
  403
  404  001426  016426                    EM27            ;"RCVR INTERRUPT WITH PRIORITY 7"
  405  001430  021523                    DH6             ;"TEST# ERR PC  RCSR"
  406  001432  022234                    DT6             ;$TESTN,$ERRPC,RCSR
  407  001434  000000                    0
  408
  409  001436  016465                    EM30            ;"CAN NOT SET BIT6 OF RCSR"
  410  001440  021523                    DH6             ;"TEST# ERR PC  RCSR"
  411  001442  022234                    DT6             ;$TESTN,$ERRPC,RCSR
  412  001444  000000                    0
  413
  414  001446  016516                    EM31            ;"CAN NOT CLEAR BIT6 OF RCSR"
  415  001450  021523                    DH6             ;"TEST# ERR PC  RCSR"
  416  001452  022234                    DT6             ;$TESTN,$ERRPC,RCSR
  417  001454  000000                    0
  418
  419  001456  016551                    EM32            ;"CAN NOT CLEAR BIT6 OF RCSR WITH RESET2
  420  001460  021523                    DH6             ;"TEST# ERR PC  RCSR"
  421  001462  022234                    DT6             ;$TESTN,$ERRPC,RCSR
  422  001464  000000                    0
  423
  424  001466  016617                    EM33            ;"BIT6 OF LKS NOT CLEAR AFTER RESET"
  425  001470  021575                    DH10            ;"TEST# ERR PC  LKS"
  426  001472  022254                    DT10            ;$TESTN,$ERRPC,LKS
  427  001474  000000                    0
  428
  429  001476  016661                    EM34            ;"LKS INTERRUPT WITH PRIORITY 7"
  430  001500  021575                    DH10            ;"TEST# ERR PC  LKS"
  431  001502  022254                    DT10            ;$TESTN,$ERRPC,LKS
  432  001504  000000                    0
  433
  434  001506  016717                    EM35            ;"CAN NOT SET BIT6 OF LKS"
  435  001510  021575                    DH10            ;"TEST# ERR PC  LKS"
  436  001512  022254                    DT10            ;$TESTN,$ERRPC,LKS
  437  001514  000000                    0
  438
  439  001516  016747                    EM36            ;"CAN NOT CLEAR BIT6 OF LKS"
  440  001520  021575                    DH10            ;"TEST# ERR PC  LKS"
  441  001522  022254                    DT10            ;$TESTN,$ERRPC,LKS
  442  001524  000000                    0
  443
  444  001526  017001                    EM37            ;"RESET DID NOT CLEAR BIT6 OF LKS"
```

```
  445  001530  021575                        DH10            ;"TEST# ERR PC  LKS"
  446  001532  022254                        DT10            ;$TESTN,$ERRPC,LKS
  447  001534  000000                        0
  448
  449  001536  017041                        EM40            ;"DUAL ADDRESSING ERROR"
  450  001540  021621                        DH40            ;"TEST# ERR PC  GOOD    BAD"
  451  001542  022264                        DT40            ;$TESTN,$ERRPC,$GDADR,$BDCSR
  452  001544  000000                        0
  453
  454  001546  017067                        EM41            ;"BIT7 OF LKS NOT SET AFTER RESET2
  455  001550  021575                        DH10            ;"TEST# ERR PC  LKS"
  456  001552  022254                        DT10            ;$TESTN,$ERRPC,LKS
  457  001554  000000                        0
  458
  459  001556  017127                        EM42            ;"CAN NOT CLEAR BIT7 OF LKS"
  460  001560  021575                        DH10            ;"TEST# ERR PC  LKS"
  461  001562  022254                        DT10            ;$TESTN,$ERRPC,LKS
  462  001564  000000                        0
  463
  464  001566  017161                        EM43            ;"BIT7 OF LKS DOES NOT SET"
  465  001570  021575                        DH10            ;"TEST# ERR PC  LKS"
  466  001572  022254                        DT10            ;$TESTN,$ERRPC,LKS
  467  001574  000000                        0
  468
  469  001576  017212                        EM44            ;"RTC INTERRUPT AT PRIORITY 7"
  470  001600  021575                        DH10            ;"TEST# ERR PC  LKS"
  471  001602  022254                        DT10            ;$TESTN,$ERRPC,LKS
  472  001604  000000                        0
  473
  474  001606  017246                        EM45            ;"RTC INTERRUPTS WHEN DISABLED"
  475  001610  021575                        DH10            ;"TEST# ERR PC  LKS"
  476  001612  022254                        DT10            ;$TESTN,$ERRPC,LKS
  477  001614  000000                        0
  478
  479  001616  017303                        EM46            ;"RTC INTERRUPT DID NOT OCCUR"
  480  001620  021575                        DH10            ;"TEST# ERR PC  LKS"
  481  001622  022254                        DT10            ;$TESTN,$ERRPC,LKS
  482  001624  000000                        0
  483
  484  001626  017303                        EM47            ;"RTC INTERRUPT DID NOT OCCUR"
  485  001630  021575                        DH10            ;"TEST# ERR PC  LKS"
  486  001632  022254                        DT10            ;$TESTN,$ERRPC,LKS
  487  001634  000000                        0
  488
  489  001636  017337                        EM50            ;"RTC DOUBLE INTERRUPT"
  490  001640  021575                        DH10            ;"TEST# ERR PC  LKS"
  491  001642  022254                        DT10            ;$TESTN,$ERRPC,LKS
  492  001644  000000                        0
  493
  494  001646  017364                        EM51            ;"RESET DID NOT CLEAR RTC INTERRUPT"
  495  001650  021575                        DH10            ;"TEST# ERR PC  LKS"
  496  001652  022254                        DT10            ;$TESTN,$ERRPC,LKS
  497  001654  000000                        0
  498
  499  001656  017414                        EM52            ;"RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS"
  500  001660  021575                        DH10            ;"TEST# ERR PC  LKS"
```

```
  501  001662  022254                        DT10            ;$TESTN,$ERRPC,LKS
  502  001664  000000                        0
  503
  504  001666  017471                        EM53
  505  001670  021655                        DH53            ;"TEST# ERR PC  LKS   CNT1   CNT2"
  506  001672  022276                        DT53            ;$TESTN,$ERRPC,LKS,FIRST,SECND
  507  001674  000000                        0
  508
  509  001676  017515                        EM54            ;"XMIT INTERRUPTS WHEN DISABLED"
  510  001700  021451                        DH1             ;"TEST# ERR PC  TCSR"
  511  001702  022214                        DT1             ;$TESTN,$ERRPC,TCSR
  512  001704  000000                        0
  513
  514  001706  017653                        EM55            ;"XMIT DID NOT INTERRUPT"
  515  001710  021451                        DH1             ;"TEST# ERR PC  TCSR"
  516  001712  022214                        DT1             ;$TESTN,$ERRPC,TCSR
  517  001714  000000                        0
  518
  519  001716  017553                        EM56            ;"XMIT INTERRUPT AT PRIORITY 7"
  520  001720  021451                        DH1             ;"TEST# ERR PC  TCSR"
  521  001722  022214                        DT1             ;$TESTN,$ERRPC,TCSR
  522  001724  000000                        0
  523
  524  001726  017611                        EM57            ;"XMIT INTERRUPTS WITH ENABLE CLEAR"
  525  001730  021451                        DH1             ;"TEST# ERR PC  TCSR"
  526  001732  022214                        DT1             ;$TESTN,$ERRPC,TCSR
  527  001734  000000                        0
  528
  529  001736  017653                        EM60            ;"XMIT DID NOT INTERRUPT"
  530  001740  021451                        DH1             ;"TEST# ERR PC  TCSR"
  531  001742  022214                        DT1             ;$TESTN,$ERRPC,TCSR
  532  001744  000000                        0
  533
  534  001746  017702                        EM61            ;"XMIT RE-INTERRUPTED"
  535  001750  021451                        DH1             ;"TEST# ERR PC  TCSR"
  536  001752  022214                        DT1             ;$TESTN,$ERRPC,TCSR
  537  001754  000000                        0
  538
  539  001756  017726                        EM62            ;"LOADING TBUF DID NOT CLEAR INTERRUPT"
  540  001760  021451                        DH1             ;"TEST# ERR PC  TCSR"
  541  001762  022214                        DT1             ;$TESTN,$ERRPC,TCSR
  542  001764  000000                        0
  543
  544  001766  017773                        EM63            ;"RCVR ACTIVE NOT SET"
  545  001770  021523                        DH6             ;"TEST# ERR PC  RCSR"
  546  001772  022234                        DT6             ;$TESTN,$ERRPC,RCSR
  547  001774  000000                        0
  548
  549  001776  020017                        EM64            ;"RECEIVER DONE NEVER SET"
  550  002000  021523                        DH6             ;"TEST# ERR PC  RCSR"
  551  002002  022234                        DT6             ;$TESTN,$ERRPC,RCSR
  552  002004  000000                        0
  553
  554  002006  020043                        EM65            ;"RCVR ACTIVE NOT CLEARED WITH DONE SET2
  555  002010  021523                        DH6             ;"TEST# ERR PC  RCSR"
  556  002012  022234                        DT6             ;$TESTN,$ERRPC,RCSR
```

```
   557  002014  000000                        0
   558
   559  002016  020111                        EM66    ;"RESET DID NOT CLEAR RCVR DONE"
   560  002020  021523                        DH6     ;"TEST# ERR PC  RCSR"
   561  002022  022234                        DT6     ;$TESTN,$ERRPC,RCSR
   562  002024  000000                        0
   563
   564  002026  020147                        EM67    ;"RDR ENABLE SET DID NOT CLEAR RCVR DONE"
   565  002030  021523                        DH6     ;"TEST# ERR PC  RCSR"
   566  002032  022234                        DT6     ;$TESTN,$ERRPC,RCSR
   567  002034  000000                        0
   568
   569  002036  020212                        EM70    ;"READING RBUF DID NOT CLEAR RCVR DONE"
   570  002040  021523                        DH6     ;"TEST# ERR PC  RCSR"
   571  002042  022234                        DT6     ;$TESTN,$ERRPC,RCSR
   572  002044  000000                        0
   573
   574  002046  020257                        EM71    ;"RCVR INTERRUPTS WITH ENABLE CLEAR"
   575  002050  021523                        DH6     ;"TEST# ERR PC  RCSR"
   576  002052  022234                        DT6     ;$TESTN,$ERRPC,RCSR
   577  002054  000000                        0
   578
   579  002056  020426                        EM72    ;"RCVR DID NOT INTERRUPT"
   580  002060  021523                        DH6     ;"TEST# ERR PC  RCSR"
   581  002062  022234                        DT6     ;$TESTN,$ERRPC,RCSR
   582  002064  000000                        0
   583
   584  002066  020321                        EM73    ;"RCVR INTERRUPTS AT PRIORITY 7"
   585  002070  021523                        DH6     ;"TEST# ERR PC  RCSR"
   586  002072  022234                        DT6     ;$TESTN,$ERRPC,RCSR
   587  002074  000000                        0
   588
   589  002076  020357                        EM74    ;"RCVR INTERRUPT REQUEST PASSED WITH ENABLE CLEAR"
   590  002100  021523                        DH6     ;"TEST# ERR PC  RCSR"
   591  002102  022234                        DT6     ;$TESTN,$ERRPC,RCSR
   592  002104  000000                        0
   593
   594  002106  020426                        EM75    ;"RCVR DID NOT INTERRUPT"
   595  002110  021523                        DH6     ;"TEST# ERR PC  RCSR"
   596  002112  022234                        DT6     ;$TESTN,$ERRPC,RCSR
   597  002114  000000                        0
   598
   599  002116  020455                        EM76    ;"RECEIVER RE-INTERRUPTED"
   600  002120  021523                        DH6     ;"TEST# ERR PC  RCSR"
   601  002122  022234                        DT6     ;$TESTN,$ERRPC,RCSR
   602  002124  000000                        0
   603
   604  002126  020501                        EM77    ;"READING RBUF DID NOT CLEAR INTERRUPT"
   605  002130  021523                        DH6     ;"TEST# ERR PC  RCSR"
   606  002132  022234                        DT6     ;$TESTN,$ERRPC,RCSR
   607  002134  000000                        0
   608
   609  002136  020546                        EM100   ;"RESET DID NOT CLEAR RCVR INTERRUPT"
   610  002140  021523                        DH6     ;"TEST# ERR PC  RCSR"
   611  002142  022234                        DT6     ;$TESTN,$ERRPC,RCSR
   612  002144  000000                        0
```

```
                                EM101           ;"'OR' FLAG DID NOT SET"
002146  020611                  DH6
002150  021523                  DT6             ;$TESTN,$ERRPC,RCSR
002152  022234                  0
002154  000000
                                EM102           ;"'ERROR' NOT SET WITH 'OR' FLAG"
002156  020637                  DH6             ;"TEST# ERR PC  RCSR"
002160  021523                  DT6             ;$TESTN,$ERRPC,RCSR
002162  022234                  0
002164  000000
                                EM103           ;"BREAK DID NOT TRANSMIT ALL ZEROES"
002166  020676                  DH103           ;"TEST# ERR PC  RCSR  DATA"
002170  021722                  DT103           ;$TESTN,$ERRPC,RCSR,$BDDAT
002172  022312                  0
002174  000000
                                EM104           ;"BREAK DID NOT SET FRAMING ERROR"
002176  020734                  DH6             ;"TEST# ERR PC  RCSR"
002200  021523                  DT6             ;$TESTN,$ERRPC,RCSR
002202  022234                  0
002204  000000
                                EM105           ;"DATA COMPARE ERROR"
002206  020771                  DH105           ;"TEST# ERR PC  RCSR  GOOD  BAD"
002210  021757                  DT105           ;$TESTN,$ERRPC,RCSR,GD,BD
002212  022324                  0
002214  000000
                                EM106           ;"DATA COMPARE ERROR WITH CABLE"
002216  021014                  DH105           ;"TEST# ERR PC  RCSR    GOOD    BAD"
002220  021757                  DT105           ;$TESTN,$ERRPC,RCSR,GD,BD
002222  022324                  0
002224  000000
                                EM107           ;"TIMEOUT IN EXERCISER TEST"
002226  021052                  DH6             ;"TEST# ERR PC  RCSR"
002230  021523                  DT6             ;$TESTN,$ERRPC,RCSR
002232  022234                  0
002234  000000
                                EM110           ;"INCORRECT RECEIVE COUNT
002236  021104                  DH110           ;"TEST# ERR PC  RCSR    TRANS   RCV"
002240  022023                  DT110           ;$TESTN,$ERRPC,RCSR,XMTCNT,RCVCNT
002242  022340                  0
002244  000000
                                EM111           ;"DATA COMPARE ERROR IN EXERCISER"
002246  021134                  DH105           ;"TEST# ERR PC  RCSR    GOOD    BAD"
002250  021757                  DT105           ;$TESTN,$ERRPC,RCSR,GD,BD
002252  022324                  0
002254  000000
                                EM112           ;"TRAP CATCHER"
002256  021174                  DH112           ;"TEST#   ERR PC  RCSR   OLDPC  TRAP ADR"
002260  022067                  DT112           ;$TESTN,$ERRPC,RCSR,OLDPC,BDVECT
002262  022354                  0
002264  000000
                                EM113           ;"NO CLK INTERRUPT IN EXERCISER"
002266  021211                  DH10            ;"TEST#  ERR PC  LKS"
002270  021575                  DT10            ;$TESTN,$ERRPC,LKS
002272  022254                  0
002274  000000
```

```
                                        EM114                   ;"'ERROR' NOT SET WITH 'FR' FLAG"
                                        DH6                     ;"TEST# ERR PC  RCSR"
                                        DT6                     ;$TESTN,$ERRPC,RCSR
                                        0

                                        EM115                   ;RCV ACTVIE NOT CLEAR WITH INIT
                                        DH6                     ;"TEST# ERR PC  RCSR"
                                        DT6                     ;$TESTN,$ERRPC,RCSR
                                        0

                                        EM116                   ;RCV ACTIVE WITHOUT "START" BIT
                                        DH6                     ;"TEST# ERR PC  RCSR"
                                        DT6                     ;$TESTN,$ERRPC,RCSR
                                        0

                                        EM117                   ;RDR ENABLE NOT CLEAR WITH RCV ACTIVE
                                        DH6                     ;"TEST# ERR PC  RCSR"
                                        DT6                     ;$TESTN,$ERRPC,RCSR
                                        0

                                CTSTFL: .WORD   0               ;CONSLE UNDER TEST FLAG
                                TMP1:   .WORD   0               ;TEMP LOCATION FOR TABLE OFFSETS
                                TMP2:   .WORD   0               ;TEMP LOCATION FOR DEVICE COUNT
                                TMP3:   .WORD   0               ;LOCATION FOR DEVICE MAP BIT TEST MASK
                                ;REGISTER AND VECTOR ADDRESSES FOR THE DL-11W UNDER TEST

                                RCSR:   .WORD   0
                                RBUF:   .WORD   0
                                TCSR:   .WORD   0
                                TBUF:   .WORD   0
                                RVECT:  .WORD   0
                                RPSW:   .WORD   0
                                TVECT:  .WORD   0
                                TPSW:   .WORD   0

                                ;CONSOLE REGISTER AND VECTOR ADDRESSES FOR THE DL-11W

                                CRCSR:  177560                  ;ADDRESS OF RECEIVER COMMAND/STATUS REGISTER
                                CRBUF:  177562                  ;ADDRESS OF RECEIVER BUFFER
                                CTCSR:  177564                  ;ADDRESS OF TRANSMITTER COMMAND/STATUS REGISTER
                                CTBUF:  177566                  ;ADDRESS OF TRANSMITTER BUFFER
                                CRVECT: 60                      ;RECEIVER INTERRUPT VECTOR
                                CRPSW:  62
                                CTVECT: 64                      ;TRANSMITTER INTERRUPT VECTOR
                                CTPSW:  66

                                ;REAL TIME CLOCK REGISTER AND VECTOR ADDRESSES
                                LKS:    .WORD   177546
                                RTCVT:  .WORD   100
                                RTCPSW: .WORD   102

                                ADRTBL: .BLKW   20
                                VCTTBL: .BLKW   20
```

```
                                ;SUBROUTINE TO GENERATE DEVICE ADDRESS TABLE
                                
002514  012702  002414          DEVADR: MOV     #ADRTBL,R2      ;POINT R2 TO THE DEVICE ADDRESS TABLE
002520  016700  176416                  MOV     $BASE,R0        ;LOAD BASE DEVICE ADDRESS IN R0
002524  010001                          MOV     R0,R1           ;
002526  062701  000170                  ADD     #170,R1         ;POINT R1 TO LAST DEVICE ADDRESS
002532  010022                  1$:     MOV     R0,(R2)+        ;MOVE DEVICE ADDRESS TO TABLE
002534  062700  000010                  ADD     #10,R0          ;POINT R0 TO NEXT DEVICE ADDRESS
002540  020001                          CMP     R0,R1           ;FINISHED GENERATING TABLE?
002542  003773                          BLE     1$              ;BR, IF LAST DEVICE ADDRESS NOT LOADED
002544  000207                          RTS     PC
                                
                                
                                
002546  005067  176316          START:  CLR     $FATAL          ;CLEAR ERROR NO.
002552  005067  176310                  CLR     $MSGTYP         ;CLEAR MESSAGE TYPE
002556  005067  176310                  CLR     $TESTN          ;CLEAR TEST NO.
002562  005067  177550                  CLR     CTSTFL          ;CLEAR CONSOLE UNDER TEST FLAG
002566  005067  176304                  CLR     $DEVCT          ;CLEAR DEVICE COUNT
002572  005067  176302                  CLR     $UNIT           ;CLEAR UNIT NUMBER
002576  005767  176310                  TST     $USWR           ;IS $USWR LOADED?
002602  001003                          BNE     1$              ;BR IF YES
002604  012767  000400  176300          MOV     #400,$USWR      ;ELSE, DEFAULT TO $USWR=400
002612  012737  000006  000004  1$:     MOV     #6,@#4          ;INITIALIZE TIMEOUT VECTORS TO TRAP
002620  012737  000003  000006          MOV     #3,@#6          ;  CATCHER ROUTINE
                                
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
002626  012706  001000                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
002632  005026                  2$:     CLR     (R6)+           ;;CLEAR MEMORY LOCATION
002634  022706  001040                  CMP     #SWR,R6 ;;DONE?
002640  001374                          BNE     .-6             ;;LOOP BACK IF NO
002642  012706  001000                  MOV     #1000,SP                ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
002646  012737  013000  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
002654  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
002662  012737  012304  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
002670  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
002676  012737  014704  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
002704  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
002712  012737  012622  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
002720  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
002726  016767  007200  007170          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
002734  005067  176120                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
002740  112767  000001  176047          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
002746  012767  002746  176032          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
002754  012767  002754  176026          MOV     #.,$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.
002762  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
002766  012737  003022  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
002774  012767  177570  176036          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
003002  012767  177570  176032          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
003010  022777  177777  176022          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
003016  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
```

```
 781  003020  000403                        BR      65$             ;;BRANCH IF NO TIMEOUT
 782  003022  012716  003030        64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
 783  003026  000002                        RTI
 784  003030  012767  000176  176002  65$:  MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
 785  003036  012767  000174  175776        MOV     #DISPREG,DISPLAY
 786  003044  012637  000004        66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
 787
 788  003050  005067  176020                CLR     $PASS           ;;CLEAR PASS COUNT
 789  003054  132767  000200  176025        BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
 790  003062  001403                        BEQ     67$             ;;YES,USE NON-APT SWITCH
 791  003064  012767  001110  175746        MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
 792  003072                        67$:
 793  003072  032777  000020  175740        BIT     #BIT4,@SWR      ;TEST CLOCK ONLY?
 794  003100  001404                        BEQ     INIT            ;BR IF NOT
 795  003102  005267  177230                INC     CTSTFL          ;ELSE, SET CONSOLE TEST FLAG TO ENABLE CLOCK TESTS
 796  003106  000167  000744                JMP     ID              ;   AND JUMP TO TYPE PROGRAM ID
 797  003112  132767  000001  175766  INIT: BITB    #BIT0,$ENV      ;CHECK IF ON APT
 798  003120  001404                        BEQ     MANL            ;BR IF NOT APT
 799  003122  132767  000200  175757        BITB    #BIT7,$ENVM     ;DID APT SIZE
 800  003130  001056                        BNE     APTSZD          ;BR, IF APT SIZED
 801  003132  032777  000040  175700  MANL: BIT     #BIT5,@SWR      ;WAS "$DEVM" MANUALLY SET?
 802  003140  001052                        BNE     APTSZD          ;IF YES, SKIP SELF-SIZING
 803
 804  003142  004767  177346        SIZE:   JSR     PC,DEVADR       ;GENERATE DEVICE ADDRESS TABLE
 805  003146  005067  177170                CLR     TMP2            ;CLR TEMP LOCATION TO KEEP DEVICE COUNT
 806  003152  005067  175766                CLR     $DEVM           ;CLEAR DEVICE MAP
 807  003156  013703  000004                MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
 808  003162  012737  003212  000004        MOV     #4$,@#4         ;SET TIMEOUT POINTER
 809  003170  016700  175746                MOV     $BASE,R0        ;LOAD BASE ADDRESS
 810  003174  062700  000160                ADD     #160,R0         ;POINT R0 TO UNIT #15 (UNIT#0=CONSOLE)
 811  003200  005710                3$:     TST     (R0)            ;CHECK FOR DEVICE EXISTANCE
 812  003202  005267  175736                INC     $DEVM           ;INDICATE DEVICE EXISTANCE IN DEVICE MAP
 813  003206  005267  177130                INC     TMP2            ;INCREMENT DEVICE COUNT
 814  003212  012706  001000        4$:     MOV     #1000,SP                ;RESET STACK POINTER
 815  003216  006367  175722                ASL     $DEVM           ;ADJUST DEVICE MAP FOR NEXT UNIT CHECK
 816  003222  162700  000010                SUB     #10,R0          ;POINT R0 TO NEXT DEVICE NUMBER
 817  003226  026700  175710                CMP     $BASE,R0        ;FINISHED SIZING?
 818  003232  003762                        BLE     3$              ;BR, IF BASE ADDRESS HAS NOT BEEN CHECKED
 819  003234  016700  177126                MOV     CRCSR,R0        ;LOAD CONSOLE DEVICE ADDRESS
 820  003240  012737  003260  000004        MOV     #5$,@#4         ;SET UP TIMEOUT POINTER
 821  003246  005710                        TST     (R0)            ;TEST FOR CONSOLE EXISTANCE
 822  003250  005267  175670                INC     $DEVM           ;INDICATE CONSOLE EXISTANCE IN DEVICE MAP
 823  003254  005267  177062                INC     TMP2            ;INCREMENT DEVICE COUNT
 824  003260  010337  000004        5$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
 825  003264  000415                        BR      VCTADR          ;BR TO GENERATE VECTOR ADDRESS TABLE
 826
 827  003266  005067  177050        APTSZD: CLR     TMP2            ;CLEAR TEMP LOCATION TO KEEP DEVICE CNT
 828  003272  016702  175646                MOV     $DEVM,R2        ;MOVE DEVICE MAP TO R2
 829  003276  005702                TSTDVM: TST     R2              ;TEST MSB OF DEVICE MAP
 830  003300  100002                        BPL     1$              ;BR, IF MSB IS ZERO
 831  003302  005267  177034                INC     TMP2            ;INCREMENT DEVICE COUNT, IF MSB=1
 832  003306  006302                1$:     ASL     R2              ;SHIFT NEXT BIT INTO MSB POSITION
 833  003310  001401                        BEQ     DVADT           ;BR, IF NO OTHER BITS ARE SET IN $DEVM
 834  003312  000771                        BR      TSTDVM          ;CONTINUE CHECKING $DEVM, IF MORE BITS SET
 835  003314  004767  177174        DVADT:  JSR     PC,DEVADR       ;GENERATE DEVICE ADDRESS TABLE
 836
```

```
                                ;GENERATE VECTOR ADDRESS TABLE

003320  012702  002454        VCTADR: MOV     #VCTTBL,R2      ;GET LOCATION OF VECTOR TABLE
003324  113700  001136                MOVB    @#$VECTI,R0     ;COPY BASE VECTOR
003330  042700  177400                BIC     #177400,R0      ;CLEAR BYTE SIGN EXTENSION
003334  010001                      MOV     R0,R1           ;
003336  062701  000170                ADD     #170,R1         ;POINT R1 TO LAST DEVICE VECTOR
003342  010022                1$:   MOV     R0,(R2)+        ;PUT VECTOR ADDRESS IN TABLE
003344  062700  000010                ADD     #10,R0          ;POINT R0 TO NEXT VECTOR ADDRESS
003350  020001                      CMP     R0,R1           ;FINISHED GENERATING VECTOR TABLE?
003352  003773                      BLE     1$              ;BR, IF LAST VECTOR IS NOT LOADED

                                ;MOVE DEVICE COUNT INTO DEVICE COUNT MESSAGE

003354  016700  176762                MOV     TMP2,R0         ;COPY DEVICE COUNT INTO R0
003360  005001                      CLR     R1              ;CLEAR AUXILARY REGISTER
003362  000300                      SWAB    R0              ;PUT DEVICE COUNT IN UPPER BYTE OF R0
003364  006300                      ASL     R0              ;MOVE MSB OF COUNT INTO
003366  006300                      ASL     R0              ;MSB OF R0
003370  006300                SHIFT:ASL     R0              ;PUT MSB OF COUNT INTO CARRY
003372  106101                      ROLB    R1              ;MOVE MSB OF COUNT INTO R1
003374  006300                      ASL     R0              ;MOVE NEXT BIT TO CARRY
003376  106101                      ROLB    R1              ;MOVE INTO R1
003400  006300                      ASL     R0              ;MOVE LAST BIT OF DIGIT
003402  106101                      ROLB    R1              ;INTO R1
003404  062701  000060                ADD     #60,R1          ;CONVERT DIGIT TO ASCII
003410  000301                      SWAB    R1              ;MOVE DIGIT TO UPPER BYTE
003412  032701  000020                BIT     #BIT4,R1        ;HAVE BOTH DIGITS BEEN MOVED TO R1?
003416  001764                      BEQ     SHIFT           ;BR, IF NOT
003420  010167  016540                MOV     R1,M2A          ;MOVE DEVICE COUNT TO OUTPUT MESSAGE


003424  052767  000002  176712  BEGIN:BIS     #BIT1,TMP3      ;SET UP BIT MASK TO TEST $DEVM FOR DEVICES EXCEPT CONSOL
003432  005067  176702                CLR     TMP1            ;CLEAR LOCATION TO STORE TABLE OFFSETS
003436  032767  000001  175500        BIT     #BIT0,$DEVM     ;IS CONSOLE TO BE TESTED?
003444  001001                      BNE     TCONS           ;BR, IF CONSOLE IS TO BE TESTED
003446  000414                      BR      TSTDEV          ;BR, TO TEST OTHER DEVICES
003450  005267  176662          TCONS:INC     CTSTFL          ;INDICATE CONSOLE UNDER TEST
003454  012700  002366                MOV     #CRCSR,R0       ;SET UP CONSOLE DEVICE ADDRESSES
003460  012701  002346                MOV     #RCSR,R1        ;POINT R1 TO UUT ADDRESS TABLE
003464  012021                1$:   MOV     (R0)+,(R1)+     ;TRANSFER CONSOLE ADDRESSES
003466  022701  002364                CMP     #TPSW,R1        ;FINISHED TRANSFER?
003472  002374                      BGE     1$              ;BR, IF NOT
003474  000167  000122                JMP     TST1            ;GO TEST CONSOLE INTERFACE


                                ;PREPARE ADDRESSES AND VECTORS FOR UUT
003500  036767  176640  175436  TSTDEV: BIT     TMP3,$DEVM      ;CHECK TO SEE IF DEVICE IS TO BE TESTED
003506  001010                      BNE     SETADR          ;BR, IF YES
003510  006367  176630                ASL     TMP3            ;SHIFT MASK TO CHECK NEXT $DEVM BIT
003514  062767  000002  176616        ADD     #2,TMP1         ;INCREMENT TABLE INDEX
003522  005267  175352                INC     $UNIT           ;INCREMENT UNIT NUMBER
003526  000764                      BR      TSTDEV          ;GO TEST NEXT BIT OF DEVICE MAP

003530  005267  175344        SETADR: INC     $UNIT           ;UPDATE UNIT NUMBER
003534  006367  176604                ASL     TMP3            ;UPDATE DEVICE MAP TEST MASK
003540  016702  176574                MOV     TMP1,R2         ;MOVE TABLE OFFSET TO R2
```

```
 893  003544  062767  000002  176566          ADD     #2,TMP1          ;UPDATE TABLE OFFSET FOR NEXT DEVICE
 894  003552  016200  002414                  MOV     ADRTBL(R2),R0    ;PUT UUT ADDRESS INTO R0
 895  003556  012701  002346                  MOV     #RCSR,R1         ;POINT R1 TO STORAGE AREA FOR UUT ADDRESSES
 896  003562  010021                  ADR:    MOV     R0,(R1)+         ;TRANSFER UUT ADDRESS
 897  003564  062700  000002                  ADD     #2,R0            ;POINT TO NEXT UUT REGISTER
 898  003570  030027  000006                  BIT     R0,#6            ;FINISHED TRANSFER?
 899  003574  001372                          BNE     ADR              ;BR, IF NOT
 900
 901  003576  016200  002454                  MOV     VCTTBL(R2),R0    ;PUT UUT VECTOR INTO R0
 902  003602  010021                  VECT:   MOV     R0,(R1)+         ;TRANSFER UUT VECTORS TO ACTIVE TABLE AREA
 903  003604  062700  000002                  ADD     #2,R0            ;POINT TO NEXT VECTOR
 904  003610  030027  000006                  BIT     R0,#6            ;FINISHED TRANSFER?
 905  003614  001372                          BNE     VECT             ;BR, IF NOT
 906  003616  000167  000000                  JMP     TST1             ;GO TEST DEVICE
 907
 908
```

```
                                ;;*****************************************************************
                                ;*TEST 1         TEST ABILITY TO REFERENCE TCSR
                                ;;*****************************************************************
003622  000004                  TST1:   SCOPE
003624  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
003630  012737  003644  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
003636  005777  176510                  TST     @TCSR           ;REFERENCE THE XMIT COMMAND/STATUS REG.
003642  000412                        BR      4$              ; GO TO END OF TEST

003644  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
003646  005767  176464                  TST     CTSTFL          ;CHECK IF DEVICE IS CONSOLE
003652  001002                        BNE     2$              ;IF YES, SKIP ERROR TYPEOUT
003654  104001                        ERROR   1               ;REPORT ERROR TO APT & TTY
003656  000404                        BR      4$              ;BR TO END OF TEST
003660                          2$:
003660  004767  007304                  JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
003664  000001                        1                               ;;THE ERROR NUMBER (FROM APT LIST)
003666  000000                  3$:     HALT
003670  010337  000004          4$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR


                                ;;*****************************************************************
                                ;*TEST 2         TEST ABILITY TO REFERENCE TBUF
                                ;;*****************************************************************
003674  000004                  TST2:   SCOPE
003676  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
003702  012737  003716  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
003710  005777  176440                  TST     @TBUF           ;REFERENCE THE XMIT BUFFER
003714  000412                        BR      4$              ;GO TO END OF  TEST

003716  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
003720  005767  176412                  TST     CTSTFL          ;CHECK IF DEVICE IS CONSOLE
003724  001002                        BNE     2$              ;IF YES, SKIP ERROR TYPEOUT
003726  104002                        ERROR   2               ;REPORT ERROR TO APT & TTY
003730  000404                        BR      4$              ;BR TO END OF TEST
003732                          2$:
003732  004767  007232                  JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
003736  000002                        2                               ;;THE ERROR NUMBER (FROM APT LIST)
003740  000000                  3$:     HALT
003742  010337  000004          4$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
```

```
                                ;;*****************************************************************
                                ;*TEST 3        TEST THAT TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED
                                ;;*****************************************************************
003746  000004                  TST3:   SCOPE
003750  005077  176400                  CLR     @TBUF           ;LOAD XBUF
003754  105777  176372                  TSTB    @TCSR           ;CHECK DONE
003760  100016                          BPL     3$              ;BR IF CLEAR
                                                                ;FILL SECOND BUFFER, BECAUSE REFRESH COULD CAUSE
                                                                ;   FIRST TEST TO FAIL
003762  005077  176366                  CLR     @TBUF           ;FILL DOUBLE BUFFER
003766  105777  176360                  TSTB    @TCSR           ;CHECK DONE
003772  100011                          BPL     3$              ;BR IF CLEAR

003774  005767  176336                  TST     CTSTFL          ;CHECK IF DEVICE IS CONSOLE
004000  001002                          BNE     1$              ;IF YES, SKIP ERROR TYPEOUT
004002  104003                          ERROR   3               ;DONE NOT CLEARED WITH TBUF FULL
004004  000404                          BR      3$              ;BR TO END OF TEST
004006                          1$:
004006  004767  007156                  JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
004012  000003                          3                       ;;THE ERROR NUMBER (FROM APT LIST)
004014  000000                  2$:     HALT                    ;TCSR "DONE" NOT CLEARED WITH TBUF FULL
004016  005000                  3$:     CLR     R0              ;CLEAR TIMER
004020  105777  176326          4$:     TSTB    @TCSR           ;CHECK FOR XMIT DONE
004024  100414                          BMI     ID              ;IF DONE SETS, BR TO END OF TEST
004026  005200                          INC     R0              ;INCREMENT TIMER
004030  001373                          BNE     4$              ;BR IF TIMER NOT DONE

004032  005767  176300                  TST     CTSTFL          ;CHECK IF DEVICE IS CONSOLE
004036  001002                          BNE     5$
004040  104004                          ERROR   4               ;TCSR "DONE" DOES NOT SET
004042  000405                          BR      ID              ;BR TO END OF TEST
004044                          5$:
004044  004767  007120                  JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
004050  000004                          4                       ;;THE ERROR NUMBER (FROM APT LIST)
004052  000000                          HALT
004054  000424                          BR      TST4            ;BR TO NEXT TEST, AND SKIP THE TYPEOUT THAT FOLLOWS
                                                                ;  BECAUSE OF THIS FAILURE

004056  023737  000042  000046  ID:     CMP     @#42,@#46       ;UNDER ACT11?
004064  001412                          BEQ     6$              ;IF YES, SKIP IDENT. TYPEOUT
004066  005767  175002                  TST     $PASS           ;IS THIS THE FIRST PASS?
004072  001007                          BNE     6$              ;IF NOT BR TO NEXT TEST & SKIP THE IDENTIFICATION TYPEOU
004074  005767  174776                  TST     $DEVCT          ;IS THIS THE FIRST SUBPASS?
004100  001004                          BNE     6$              ;IF NOT, BRTO NEXT TEST
004102  104401                          TYPE                    ;TYPE PROGRAM INDENTIFICATION
004104  022140                          M1
004106  104401                          TYPE                    ;TYPE NUMBER OF DEVICES UNDER TEST
004110  022162                          M2
004112  032777  000020  174720  6$:     BIT     #BIT4,@SWR      ;CLOCK TEST ONLY?
004120  001402                          BEQ     TST4            ;BR IF NOT
004122  000167  000730                  JMP     TCLOCK          ;ELSE, JUMP TO TEST CLOCK
```

```
1005
1006
1007                                    ;;*****************************************************************
1008                                    ;*TEST 4         TEST THAT TCSR "DONE" SETS WITH RESET
1009                                    ;;*****************************************************************
1010  004126  000004                    TST4:   SCOPE
1011  004130  005077  176220                   CLR     @TBUF           ;LOAD TRANSMIT BUFFER
1012  004134  105777  176212            1$:    TSTB    @TCSR           ;WAIT FOR DONE
1013  004140  100375                           BPL     1$
1014  004142  005077  176206                   CLR     @TBUF           ;LOAD SECOND BUFFER
1015  004146  000240                           NOP
1016  004150  000005                           RESET                   ;CLEAR DONE WITH RESET
1017  004152  105777  176174                   TSTB    @TCSR           ;CHECK FOR DONE SET
1018  004156  100401                           BMI     TST5            ;BR TO NEXT TEST IF DONE SET
1019
1020  004160  104005                           ERROR   5               ;TCSR "DONE" DOES NOT SET WITH RESET
1021
1022
1023
1024                                    ;;*****************************************************************
1025                                    ;*TEST 5         TEST ABILITY TO ACCESS RCSR
1026                                    ;;*****************************************************************
1027  004162  000004                    TST5:   SCOPE
1028  004164  013703  000004                   MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
1029  004170  012737  004204  000004           MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
1030  004176  005777  176144                   TST     @RCSR           ;ACCESS RCSR
1031  004202  000402                           BR      2$              ;BR TO END OF TEST
1032
1033  004204  022626                    1$:    CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
1034  004206  104006                           ERROR   6               ;CAN NOT ACCESS RCSR
1035  004210  010337  000004            2$:    MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
```

```
1036
1037
1038                                    ;;*********************************************************************
1039                                    ;*TEST 6        TEST ABILITY TO ACCESS RBUF
1040                                    ;;*********************************************************************
1041   004214  000004                   TST6:   SCOPE
1042   004216  013703  000004                   MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
1043   004222  012737  004236  000004           MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
1044   004230  005777  176114                   TST     @RBUF           ;ACCESS RBUF
1045   004234  000402                           BR      2$              ;BR TO END OF TEST
1046
1047   004236  022626                   1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
1048   004240  104007                           ERROR   7               ;CAN NOT ACCESS RBUF
1049   004242  010337  000004           2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
1050
1051
1052
1053
```

```
                                        ;;*********************************************************************
                                        ;*TEST 7        TEST THAT BIT0(BREAK BIT) CAN BE SET & CLEARED & RESET
                                        ;;*********************************************************************
004246  000004                  TST7:   SCOPE
004250  032777  000400  174562          BIT     #BIT8,@SWR      ;IS BREAK FUNCTION ENABLED?
004256  001545                          BEQ     TST11           ;BR TO NEXT TEST, IF NOT
004260  000005                          RESET                   ;CLEAR EVERYTHING
004262  032777  000001  176062          BIT     #BIT0,@TCSR     ;CHECK BIT0 OF TCSR CLEAR
004270  001411                          BEQ     3$              ;BR IF CLEAR
004272  005767  176040                  TST     CTSTFL
004276  001002                          BNE     1$
004300  104011                          ERROR   11              ;BIT0 WAS NOT CLEAR AFTER RESET
004302  000404                          BR      3$
004304                          1$:
004304  004767  006660                  JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
004310  000011                          11                              ;;THE ERROR NUMBER (FROM APT LIST)
004312  000000                  2$:     HALT

004314  052777  000001  176030  3$:     BIS     #BIT0,@TCSR     ;SET BIT0 IN TCSR
004322  032777  000001  176022          BIT     #BIT0,@TCSR     ;TEST BIT0 OF TCSR
004330  001001                          BNE     4$              ;BR IF SET

004332  104012                          ERROR   12              ;BIT0 OF TCSR WILL NOT SET

004334  042777  000001  176010  4$:     BIC     #BIT0,@TCSR     ;CLEAR BIT0 OF TCSR
004342  032777  000001  176002          BIT     #BIT0,@TCSR     ;TEST BIT0 OF TCSR
004350  001411                          BEQ     7$
004352  005767  175760                  TST     CTSTFL
004356  001002                          BNE     5$
004360  104013                          ERROR   13              ;BIT0 OF TCSR WILL NOT CLEAR
004362  000404                          BR      7$
004364                          5$:
004364  004767  006600                  JSR     PC,$ATY4                ;;ONLY REPORT A FATAL ERROR
004370  000013                          13                              ;;THE ERROR NUMBER (FROM APT LIST)
004372  000000                  6$:     HALT

004374  052777  000001  175750  7$:     BIS     #BIT0,@TCSR     ;SET BIT0 IN TCSR
004402  000005                          RESET                   ;CLEAR BIT0 WITH RESET
004404  032777  000001  175740          BIT     #BIT0,@TCSR     ;TEST BIT0 CLEAR
004412  001467                          BEQ     TST11           ;BR IF CLEAR
004414  042777  000001  175730          BIC     #BIT0,@TCSR     ;CLEAR BIT0, TO PRINT ERROR
004422  104014                          ERROR   14              ;RESET DID NOT CLEAR BIT0 OF TCSR
```

```
1098
1099
1100                                    ;;*****************************************************************
1101                                    ;*TEST 10       TEST THAT BIT2(MAINT. BIT) CAN BE SET & RESET
1102                                    ;;*****************************************************************
1103  004424  000004                    TST10:  SCOPE
1104  004426  000005                            RESET                           ;CLEAR EVERYTHING
1105  004430  032777  000004  175714            BIT     #BIT2,@TCSR             ;TEST FOR BIT2 OF TCSR CLEAR
1106  004436  001411                            BEQ     3$                      ;BR IF CLEAR
1107
1108  004440  005767  175672                   TST     CTSTFL                  ;CHECK IF DEVICE IS CONSOLE
1109  004444  001002                            BNE     1$                      ;IF YES, SKIP ERROR TYPEOUT
1110  004446  104015                            ERROR   15                      ;BIT2 OF TCSR NOT CLEAR AFTER RESET
1111  004450  000404                            BR      3$
1112
1113  004452                            1$:
1114  004452  004767  006512                   JSR     PC,$ATY4                        ;;ONLY REPORT A FATAL ERROR
1115  004456  000015                            15                                      ;;THE ERROR NUMBER (FROM APT LIST)
1116  004460  000000                    2$:     HALT
1117
1118  004462  052777  000004  175662    3$:     BIS     #BIT2,@TCSR             ;SET BIT2 OF TCSR
1119  004470  032777  000004  175654            BIT     #BIT2,@TCSR             ;TEST FOR BIT2 SET
1120  004476  001001                            BNE     4$                      ;BR IF SET
1121
1122  004500  104016                            ERROR   16                      ;BIT2 OF TCSR WILL NOT SET
1123
1124  004502  042777  000004  175642    4$:     BIC     #BIT2,@TCSR             ;CLEAR BIT2 OF TCSR
1125  004510  032777  000004  175634            BIT     #BIT2,@TCSR             ;TEST BIT2 CLEAR
1126  004516  001411                            BEQ     7$                      ;BR IF CLEAR
1127
1128  004520  005767  175612                   TST     CTSTFL                  ;CHECK IF DEVICE IS CONSOLE
1129  004524  001002                            BNE     5$                      ;IF YES, SKIP ERROR TYPEOUT
1130  004526  104017                            ERROR   17
1131  004530  000404                            BR      7$
1132  004532                            5$:
1133  004532  004767  006432                   JSR     PC,$ATY4                        ;;ONLY REPORT A FATAL ERROR
1134  004536  000017                            17                                      ;;THE ERROR NUMBER (FROM APT LIST)
1135  004540  000000                    6$:     HALT                            ;BIT0 OF TCSR WILL NOT CLEAR
1136
1137  004542  052777  000004  175602    7$:     BIS     #BIT2,@TCSR             ;SET BIT2 OF TCSR
1138  004550  000005                            RESET                           ;CLEAR BIT2 WITH RESET
1139  004552  032777  000004  175572            BIT     #BIT2,@TCSR             ;TEST FOR BIT2 CLEAR
1140  004560  001461                            BEQ     TST12                   ;IF CLEAR, GO TO NEXT TEST
1141
1142  004562  042777  000004  175562            BIC     #BIT2,@TCSR             ;CLEAR BIT2, TO PRINT ERROR
1143  004570  104020                            ERROR   20
1144                                                                            ;RESET DID NOT CLEAR BIT2 OF TCSR
```

```
1145
1146
1147                                    ;;*****************************************************************
1148                                    ;*TEST 11       TEST THAT BIT6(XMIT INT EN) CAN BE SET & RESET
1149                                    ;;*****************************************************************
1150  004572  000004                    TST11:  SCOPE
1151  004574  000005                            RESET                   ;CLEAR EVERYTHING
1152  004576  017703  175560                    MOV     @TVECT,R3       ;SAVE XMIT VECTOR
1153  004602  012777  004632  175552            MOV     #1$,@TVECT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
1154  004610  004767  005426                    JSR     PC,WRPSW                ;SET PSW TO PRIORITY=7
1155  004614  000340                            .WORD   340
1156  004616  032777  000100  175526            BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
1157  004624  001404                            BEQ     2$              ;BR IF ZERO
1158  004626  104021                            ERROR   21
1159                                                                    ;BIT6 IN TCSR NOT CLEAR AFTER RESET
1160  004630  000402                            BR      2$
1161
1162  004632  022626                    1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
1163  004634  104022                            ERROR   22
1164                                                                    ;XMIT INTERRUPT OCCURRED  PRIO=7
1165
1166  004636  052777  000100  175506  2$:     BIS     #BIT6,@TCSR     ;SET BIT6 OF TCSR
1167  004644  032777  000100  175500            BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
1168  004652  001001                            BNE     3$              ;BR, IF SET
1169
1170  004654  104023                            ERROR   23
1171                                                                    ;CANNOT SET BIT6 OF TCSR
1172
1173  004656  042777  000100  175466  3$:     BIC     #BIT6,@TCSR     ;CLEAR BIT6 OF TCSR
1174  004664  032777  000100  175460            BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
1175  004672  001401                            BEQ     4$              ;BR IF CLEAR
1176  004674  104024                            ERROR   24
1177                                                                    ;CANNOT CLEAR BIT6 OF TCSR
1178
1179  004676  052777  000100  175446  4$:     BIS     #BIT6,@TCSR     ;SET BIT6 OF TCSR
1180  004704  000005                            RESET                   ;CLEAR BIT6 WITH RESET
1181  004706  032777  000100  175436            BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
1182  004714  001401                            BEQ     5$              ;BR IF CLEAR
1183
1184  004716  104025                            ERROR   25
1185                                                                    ;CANNOT CLEAR BIT6 OF TCSR WITH RESET
1186  004720  010377  175436            5$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
```

```
1187
1188
1189                                    ;;*****************************************************************
1190                                    ;*TEST 12       TEST THAT BIT6 OF RCSR CAN BE SET & RESET
1191                                    ;;*****************************************************************
1192  004724  000004                    TST12:  SCOPE
1193  004726  000005                            RESET                           ;CLEAR EVERYTHING
1194  004730  017703  175422                    MOV     @RVECT,R3               ;SAVE RECEIVE VECTOR
1195  004734  012777  004764  175414            MOV     #1$,@RVECT              ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
1196  004742  004767  005274                    JSR     PC,WRPSW                ;SET PSW TO PRIORITY=7
1197  004746  000340                                    .WORD   340
1198  004750  032777  000100  175370            BIT     #BIT6,@RCSR             ;TEST BIT6 OF RCSR
1199  004756  001404                            BEQ     2$
1200  004760  104026                            ERROR   26
1201                                                                            ;BIT6 OF RCSR NOT CLEAR AFTER RESET
1202  004762  000402                            BR      2$
1203
1204  004764  022626                    1$:     CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
1205  004766  104027                            ERROR   27
1206                                                                            ;RCVR INTERRUPT WITH PRIORITY=7
1207
1208  004770  052777  000100  175350    2$:     BIS     #BIT6,@RCSR             ;SET BIT6 OF RCSR
1209  004776  032777  000100  175342            BIT     #BIT6,@RCSR             ;TEST BIT6 OF RCSR
1210  005004  001001                            BNE     3$                      ;BR, IF SET
1211
1212  005006  104030                            ERROR   30
1213                                                                            ;CANNOT SET BIT6 OF RCSR
1214
1215  005010  042777  000100  175330    3$:     BIC     #BIT6,@RCSR             ;CLEAR BIT6 OF RCSR
1216  005016  032777  000100  175322            BIT     #BIT6,@RCSR             ;TEST BIT6 OF RCSR
1217  005024  001401                            BEQ     4$                      ;BR, IF CLEAR
1218
1219  005026  104031                            ERROR   31
1220                                                                            ;CANNOT CLEAR BIT6 OF RCSR
1221
1222  005030  052777  000100  175310    4$:     BIS     #BIT6,@RCSR             ;SET BIT6 OF RCSR
1223  005036  000005                            RESET                           ;CLEAR BIT6OF RCSR WITH RESET
1224  005040  032777  000100  175300            BIT     #BIT6,@RCSR             ;TEST BIT6 OF RCSR
1225  005046  001401                            BEQ     5$                      ;BR, IF CLEAR
1226
1227  005050  104032                            ERROR   32
1228                                                                            ;CANNOT CLEAR BIT6 OF RCSR WITH RESET
1229  005052  010377  175300            5$:     MOV     R3,@RVECT               ;RESTORE RECEIVE VECTOR
1230
```

```
1231
1232  005056  012767  000012  173716  TCLOCK: MOV     #12,$TSTNM      ;ADJUST TEST NUMBER TO (NEXT TEST - 1)
1233                                  ;;**************************************************************
1234                                  ;*TEST 13       TEST ABILITY TO ACCESS LKS
1235                                  ;;**************************************************************
1236  005064  000004                  TST13:  SCOPE
1237  005066  005767  175244                  TST     CTSTFL          ;IS CONSOLE UNDER TEST?
1238  005072  001420                  BEQ     TST14           ;IF NOT, SKIP THIS TEST
1239  005074  032777  000100  173736          BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
1240  005102  001014                  BNE     TST14           ;IF YES, SKIP THIS TEST
1241  005104  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
1242  005110  012737  005124  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
1243  005116  005777  175264                  TST     @LKS            ;ACCESS LKS
1244  005122  000402                  BR      2$              ;NO TIMEOUT -  BR TO END OF TEST
1245
1246  005124  022626          1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
1247  005126  104010                  ERROR   10              ;CAN NOT ACCESS LKS
1248
1249  005130  010337  000004          2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
1250
1251                                  ;;**************************************************************
1252                                  ;*TEST 14       TEST THAT BIT6 OF LKS CAN BE SET & RESET
1253                                  ;;**************************************************************
1254  005134  000004                  TST14:  SCOPE
1255  005136  005767  175174                  TST     CTSTFL          ;IS CONSOLE UNDER TEST?
1256  005142  001460                  BEQ     TST15           ;IF NOT, SKIP THIS TEST
1257  005144  032777  000100  173666          BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
1258  005152  001054                  BNE     TST15           ;IF YES, SKIP THIS TEST
1259  005154  000005                  RESET
1260  005156  017703  175226                  MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
1261  005162  012777  005212  175220          MOV     #1$,@RTCVT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
1262  005170  004767  005046                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
1263  005174  000340                          .WORD   340
1264  005176  032777  000100  175202          BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
1265  005204  001404                  BEQ     2$
1266  005206  104033                  ERROR   33
1267                                                          ;BIT6 OF LKS NOT CLEAR AFTER RESET
1268  005210  000402                  BR      2$
1269
1270  005212  022626          1$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
1271  005214  104034                  ERROR   34
1272                                                          ;LKS INTERRUPT WITH PRIORITY=7
1273
1274  005216  052777  000100  175162  2$:     BIS     #BIT6,@LKS      ;SET BIT6 OF LKS
1275  005224  032777  000100  175154          BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
1276  005232  001001                  BNE     3$              ;BR IF SET
1277
1278  005234  104035                  ERROR   35
1279                                                          ;CANNOT SET BIT6 OF LKS
1280
1281  005236  042777  000100  175142  3$:     BIC     #BIT6,@LKS      ;CLEAR BIT6 OF LKS
1282  005244  032777  000100  175134          BIT     #BIT6,@LKS      ;TEST BIT6 OF LK
1283  005252  001401                  BEQ     4$
1284  005254  104036                  ERROR   36
1285                                                          ;CANNOT CLEAR BIT6 OF LKS
1286  005256  052777  000100  175122  4$:     BIS     #BIT6,@LKS      ;SET BIT6 OF LKS
```

```
1287  005264  000005                          RESET                   ;CLEAR BIT6 OF LKS WITH RESET
1288  005266  032777  000100  175112          BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
1289  005274  001401                          BEQ     5$              ;BR IF CLEAR
1290
1291  005276  104037                          ERROR   37
1292                                                                  ;CANNOT CLEAR BIT6 OF LKS WITH RESET
1293  005300  010377  175104          5$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
```

```
1294
1295
1296                                    ;;*****************************************************************
1297                                    ;*TEST 15       TEST FOR DUAL ADDRESSING OF REGISTERS
1298                                    ;;*****************************************************************
1299  005304  000004                    TST15:  SCOPE
1300  005306  013703  000004                    MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
1301  005312  013704  000006                    MOV     @#6,R4          ;SAVE TIMEOUT PSW
1302  005316  012737  005450  000004            MOV     #4$,@#4         ;SET UP TIMEOUT VECTOR
1303  005324  012737  000340  000006            MOV     #340,@#6        ;KEEP PRIO=7
1304  005332  000005                            RESET                   ;CLEAR EVERYTHING
1305  005334  012700  000002                    MOV     #BIT1,R0        ;SET UP BIT MASK
1306  005340  032777  000020  173472            BIT     #BIT4,@SWR      ;CLOCK TEST ONLY?
1307  005346  001404                            BEQ     1$              ;BR IF NOT
1308  005350  016767  175032  173442            MOV     LKS,$GDADR      ;ELSE, MOVE GOOD LKS ADDRESS INTO $GDADR
1309  005356  000403                            BR      2$
1310  005360  016767  174762  173432  1$:       MOV     RCSR,$GDADR     ;MOVE GOOD RCSR ADDRESS INTO $GDADR
1311  005366  016767  173426  173426  2$:       MOV     $GDADR,$BDADR   ;MOVE GOOD ADDRESS INTO TEST ADDRESS LOCATION
1312  005374  040067  173422                    BIC     R0,$BDADR       ;CREATE BAD ADDRESS BY COMPLEMENTING ONE BIT
1313  005400  026767  173414  173414            CMP     $GDADR,$BDADR   ;ARE ADDRESSES IDENTICAL?
1314  005406  001002                            BNE     3$              ;IF NOT, TEST THIS ADDRESS
1315  005410  050067  173406                    BIS     R0,$BDADR       ;ELSE, BIT SET THIS BIT POSITION TO GENERATE BAD ADDRESS
1316  005414  017767  173402  173402  3$:       MOV     @$BDADR,$GDDAT  ;SAVE CONTENTS OF BAD ADDRESS IF IT EXISTS
1317  005422  052777  000100  173372            BIS     #BIT6,@$BDADR   ;SET BIT6 USING BAD ADDRESS
1318  005430  032777  000100  173362            BIT     #BIT6,@$GDADR   ;CHECK TO SEE IF GOOD ADDRESS CONTAINS BIT6
1319  005436  001011                            BNE     6$              ;BR IF SET ---> ERROR
1320  005440  016777  173360  173354            MOV     $GDDAT,@$BDADR  ;RESTORE ANY MEMORY LOCATION THAT WAS ALTERED
1321  005446  000401                            BR      5$              ;BR TO CONTINUE TEST
1322  005450  022626                    4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER TIMEOUT
1323  005452  006300                    5$:     ASL     R0              ;SHIFT BIT MASK TO NEXT POSITION
1324  005454  005700                            TST     R0              ;COMPLEMENTED ALL BITS FROM 1 - 15?
1325  005456  001343                            BNE     2$              ;BR, IF NOT
1326  005460  000401                            BR      7$              ;BR TO NEXT TEST
1327
1328  005462  104040                    6$:     ERROR   40              ;DUAL ADDRESSING ERROR
1329                                            ;$BDADR = DUAL ADDRESS
1330                                            ;$GDADR = GOOD ADDRESS
1331
1332  005464  010337  000004            7$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
1333  005470  010437  000006                    MOV     R4,@#6          ;RESTORE TIMEOUT PSW
```

```
                                        ;;*********************************************************************
                                        ;*TEST 16       TEST THAT BIT7 OF LKS SETS & CAN BE CLEARED
                                        ;;*********************************************************************
005474  000004                  TST16:  SCOPE
005476  005767  174634                  TST     CTSTFL          ;IS CONSOLE UNDER TEST?
005502  001437                          BEQ     TST17           ;IF NOT, SKIP THIS TEST
005504  032777  000100  173326          BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
005512  001033                          BNE     TST17           ;IF YES, SKIP THIS TEST
005514  000005                          RESET                   ;CLEAR EVERYTHING & SET BIT7 OF LKS
005516  105777  174664          1$:     TSTB    @LKS            ;TEST FOR BIT7 OF LKS
005522  100401                          BMI     2$              ;BR IF SET

005524  104041                          ERROR   41              ;BIT7 OF LKS DID NOT SET WITH RESET

005526  042777  000200  174652  2$:     BIC     #BIT7,@LKS      ;CLEAR BIT7 OF LKS
005534  032777  000200  174644          BIT     #BIT7,@LKS      ;TEST BIT7 OF LKS
005542  001410                          BEQ     3$
005544  042777  000200  174634          BIC     #BIT7,@LKS      ;TRY ONE MORE TIME BECAUSE THE CLOCK
005552  032777  000200  174626          BIT     #BIT7,@LKS      ;  MAY HAVE SET IMMEDIATELY AFTER THE FIRST CLEAR
005560  001401                          BEQ     3$

005562  104042                          ERROR   42              ;CAN NOT CLEAR BIT7 OF LKS

005564  005000                  3$:     CLR     R0              ;CLEAR TIMER
005566  105777  174614          CONT:   TSTB    @LKS            ;TEST FOR BIT7 OF LKS
005572  100403                          BMI     TST17           ;BR, IF SET
005574  005200                          INC     R0              ;INCREMENT TIMER
005576  001373                          BNE     CONT            ;CONTINUE UNTIL TIME EXPIRES

005600  104043                          ERROR   43              ;BIT7 OF LKS DOES NOT SET
```

```
1367
1368                                    ;;*********************************************************************
1369                                    ;*TEST 17       TEST THAT THE REAL TIME CLOCK INTERRUPTS PROPERLY
1370                                    ;;*********************************************************************
1371  005602  000004                    TST17:  SCOPE
1372  005604  005767  174526                    TST     CTSTFL          ;IS CONSOLE UNDER TEST?
1373  005610  001503                            BEQ     TST20           ;IF NOT, SKIP THIS TEST
1374  005612  032777  000100  173220            BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
1375  005620  001077                            BNE     TST20           ;IF YES, SKIP THIS TEST
1376  005622  004767  004414                    JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
1377  005626  000340                                    .WORD   340
1378  005630  017703  174554                    MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
1379  005634  017704  174552                    MOV     @RTCPSW,R4      ;SAVE LINE CLOCK PSW VECTOR
1380  005640  012777  005702  174542            MOV     #2$,@RTCVT      ;SET RTC INTERRUPT VECTOR TO ERROR REPORT
1381  005646  012777  000340  174536            MOV     #340,@RTCPSW    ;KEEP PRIORITY AT 7
1382  005654  042777  000200  174524            BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
1383  005662  052777  000100  174516            BIS     #BIT6,@LKS      ;SET INTERRUPT ENABLE
1384  005670  105777  174512          1$:       TSTB    @LKS            ;WAIT FOR RTC DONE(INTERRUPT REQUEST)
1385  005674  100375                            BPL     1$
1386  005676  000240                            NOP                     ;GIVE TIME FOR ANY INTERRUPTS
1387  005700  000402                            BR      3$              ;BR, IF NO INTERRUPT OCCURS
1388
1389  005702  022626                  2$:       CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
1390  005704  104044                            ERROR   44              ;RTC INTERRUPTS AT PRIORITY 7
1391
1392  005706  005077  174474          3$:       CLR     @LKS    ;DISABLE RTC INTERRUPTS & CLEAR DONE
1393  005712  012777  005740  174470            MOV     #4$,@RTCVT      ;SET RTC INTERRUPT VECTOR FOR ERROR
1394  005720  004767  004316                    JSR     PC,WRPSW        ;CHANGE PSW TO PRIORITY 5
1395  005724  000240                                    .WORD   240
1396  005726  105777  174454          20$:      TSTB    @LKS            ;WAIT FOR DONE (INTERRUPT REQUEST)
1397  005732  100375                            BPL     20$
1398  005734  000240                            NOP                     ;GIVE TIME FOR ANY INTERRUPT
1399  005736  000402                            BR      5$              ;IF NO INTERRUPT - BR TO CONTINUE TEST
1400
1401  005740  022626                  4$:       CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
1402  005742  104045                            ERROR   45              ;RTC INTERRUPTS WITH INTERRUPTS DISABLED
1403
1404  005744  012777  006000  174436  5$:       MOV     #7$,@RTCVT      ;POINT RTC VECTOR TO END OF TEST
1405  005752  042777  000200  174426            BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
1406  005760  052777  000100  174420            BIS     #BIT6,@LKS      ;ALLOW INTERRUPTS
1407  005766  105777  174414          6$:       TSTB    @LKS            ;WAIT FOR RTC DONE
1408  005772  100375                            BPL     6$
1409  005774  000240                            NOP                     ;GIVE TIME FOR INTERRUPT
1410
1411  005776  104046                            ERROR   46              ;RTC INTERRUPT DID NOT OCCUR
1412
1413  006000  022626                  7$:       CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
1414  006002  042777  000100  174376            BIC     #BIT6,@LKS      ;DISABLE INTERRUPTS
1415  006010  010377  174374                    MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
1416  006014  010477  174372                    MOV     R4,@RTCPSW      ;RESTORE LINE CLOCK PSW VECTOR
1417
1418
```

```
                                        ;;*****************************************************************
                                        ;*TEST 20       TEST RTC FOR DOUBLE INTERRUPTS
                                        ;;*****************************************************************
006020  000004                  TST20:  SCOPE
006022  005767  174310                  TST     CTSTFL          ;IS CONSOLE UNDER TEST?
006026  001457                          BEQ     TST21           ;IF NOT, SKIP THIS TEST
006030  032777  000100  173002          BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
006036  001053                          BNE     TST21           ;IF YES, SKIP THIS TEST
006040  000005                          RESET                   ;CLEAR EVERYTHING
006042  017703  174342                  MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
006046  017704  174340                  MOV     @RTCPSW,R4      ;SAVE LINE CLOCK PSW VECTOR
006052  012777  006122  174330          MOV     #2$,@RTCVT      ;SET UP RTC INTERRUPT VECTOR
006060  012777  000340  174324          MOV     #340,@RTCPSW    ;DISALLOW INTERRUPTS AFTER THE INTERRUPT
006066  004767  004150                  JSR     PC,WRPSW        ;SET PRIORITY TO 5
006072  000240                          .WORD   240
006074  042777  000200  174304          BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
006102  052777  000100  174276          BIS     #BIT6,@LKS      ;ENABLE CLOCK INTERRUPTS
006110  105777  174272          1$:     TSTB    @LKS            ;WAIT FOR DONE
006114  100375                          BPL     1$
006116  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPT
                                
006120  104047                          ERROR   47              ;RTC INTERRUPT DID NOT OCCUR
                                
006122  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
006124  012777  006144  174256          MOV     #3$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
006132  004767  004104                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 5
006136  000240                          .WORD   240
006140  000240                          NOP                     ;GIVE SOME TIME FOR AN INTERRUPT
006142  000402                          BR      4$              ;NO INTERRUPT - BR TO END OF TEST
                                
006144  022626                  3$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
006146  104050                          ERROR   50              ;INTERRUPT SEQUENCE DID NOT CLEAR
                                                                ;INTERRUPT REQUEST
                                
006150  042777  000100  174230  4$:     BIC     #BIT6,@LKS      ;DISABLE CLOCK INTERRUPTS
006156  010377  174226                  MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
006162  010477  174224                  MOV     R4,@RTCPSW      ;RESTORE LINE CLOCK PSW VECTOR
```

```
1459
1460
1461                                    ;;*****************************************************************
1462                                    ;*TEST 21       TEST THAT RTC INTERRUPT CLEARS WITH RESET
1463                                    ;;*****************************************************************
1464   006166  000004                   TST21:  SCOPE
1465   006170  005767  174142                   TST     CTSTFL          ;IS CONSOLE UNDER TEST?
1466   006174  001442                           BEQ     TST22           ;IF NOT, SKIP THIS TEST
1467   006176  032777  000100  172634         BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
1468   006204  001036                           BNE     TST22           ;IF YES, SKIP THIS TEST
1469   006206  004767  004030                 JSR     PC,WRPSW        ;SET PRIORITY TO 7
1470   006212  000340                                   .WORD   340
1471   006214  017703  174170                 MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
1472   006220  012777  006272  174162         MOV     #2$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
1473   006226  042777  000200  174152         BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
1474   006234  052777  000100  174144         BIS     #BIT6,@LKS      ;ENABLE CLOCK INTERRUPTS
1475   006242  105777  174140         1$:     TSTB    @LKS            ;WAIT FOR DONE (INTERRUPT REQUEST)
1476   006246  100375                         BPL     1$
1477   006250  000005                         RESET                   ;CLEAR PENDING INTERRUPT WITH RESET
1478   006252  004767  003764                 JSR     PC,WRPSW        ;SET PRIORITY TO 5
1479   006256  000240                                   .WORD   240
1480   006260  000240                         NOP                     ;GIVE TIME FOR ANY INTERRUPT
1481   006262  042777  000100  174116         BIC     #BIT6,@LKS      ;DISALLOW INTERRUPTS
1482   006270  000402                         BR      3$              ;BR TO END OF TEST
1483
1484   006272  022626                 2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
1485   006274  104051                         ERROR   51              ;RESET DID NOT CLEAR INTERRUPT
1486
1487   006276  010377  174106         3$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
```

```
                                ;;*****************************************************************
                                ;*TEST 22       TEST THAT RTC INTERRUPT CLEARS BY CLEARING BIT7 OF LKS
                                ;;*****************************************************************
006302  000004                  TST22:  SCOPE
006304  005767  174026                  TST     CTSTFL          ;IS CONSOLE UNDER TEST?
006310  001447                          BEQ     TST23           ;IF NOT, SKIP THIS TEST
006312  032777  000100  172520          BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
006320  001043                          BNE     TST23           ;IF YES, SKIP THIS TEST
006322  004767  003714                  JSR     PC,WRPSW        ;SET PRIORITY TO 7
006326  000340                          .WORD   340
006330  017703  174054                  MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
006334  012777  006412  174046          MOV     #2$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
006342  042777  000200  174036          BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
006350  052777  000100  174030          BIS     #BIT6,@LKS      ;ENABLE CLOCK INTERRUPTS
006356  105777  174024          1$:     TSTB    @LKS            ;WAIT FOR DONE (INTERRUPT REQUEST)
006362  100375                          BPL     1$
006364  042777  000200  174014          BIC     #BIT7,@LKS      ;CLEAR DONE & INTERRUPT
006372  004767  003644                  JSR     PC,WRPSW        ;ALLOW INTERRUPTS
006376  000240                          .WORD   240
006400  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPT
006402  042777  000100  173776          BIC     #BIT6,@LKS      ;DISALLOW INTERRUPTS
006410  000402                          BR      3$              ;BR TO END OF TEST

006412  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
006414  104052                          ERROR   52              ;CLEARING BIT7 OF LKS DID NOT CLEAR INTERRUPT

006416  010377  173766          3$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
006422  004767  003614                  JSR     PC,WRPSW        ;RESTORE PRIORITY TO 7
006426  000340                          .WORD   340
```

```
                                ;;*********************************************************************
                                ;*TEST 23       TEST CLOCK REPEATABILITY
                                ;;*********************************************************************
006430  000004                  TST23:  SCOPE
006432  005767  173700                  TST     CTSTFL          ;IS CONSOLE UNDER TEST?
006436  001462                          BEQ     TST24           ;IF NOT, SKIP THIS TEST
006440  032777  000100  172372          BIT     #BIT6,@SWR      ;ARE LINE CLOCK TESTS INHIBITED?
006446  001056                          BNE     TST24           ;IF YES, SKIP THIS TEST
006450  042777  000100  173730          BIC     #BIT6,@LKS      ;DISALLOW INTERRUPTS

006456  005000                          CLR     R0              ;CLEAR A TIMER
006460  012701  177777                  MOV     #-1,R1          ;SET A FLAG INDICATING FIRST PASS THRU THIS LOOP
006464  005002                  1$:     CLR     R2              ;CLEAR CLOCK COUNTER
006466  005077  173714                  CLR     @LKS            ;CLEAR DONE
006472  105777  173710          2$:     TSTB    @LKS            ;SYNC ON DONE
006476  100375                          BPL     2$
006500  005077  173702                  CLR     @LKS            ;CLEAR DONE
006504  105777  173676          3$:     TSTB    @LKS            ;IS CLOCK DONE?
006510  100003                          BPL     4$              ;BR IF NOT , TO INCREMENT TIMER
006512  005202                          INC     R2              ;IF DONE, INCREMENT CLOCK COUNT
006514  005077  173666                  CLR     @LKS            ;CLEAR DONE
006520  005200                  4$:     INC     R0              ;INCREMENT TIMER
006522  001370                          BNE     3$              ;BR IF TIME REMAINS
006524  005201                          INC     R1              ;INCREMENT LOOP PASS FLAG
006526  001003                          BNE     CMPARE          ;BR IF TWO PASSES HAVE BEEN MADE
006530  010267  000044                  MOV     R2,FIRST        ;IF NOT, STORE FIRST CLOCK COUNT
006534  000753                          BR      1$              ;DO LOOP AGAIN
006536  016701  000036          CMPARE: MOV     FIRST,R1        ;RECALL FIRST CLOCK COUNT
006542  160201                          SUB     R2,R1           ;CALCULATE DIFFERENCE OF TWO COUNTS
006544  100001                          BPL     TOLER           ;IF POSITIVE,SKIP NEGATION OF DIFFERENCE
006546  005401                          NEG     R1              ;MAKE DIFFERENCE A POSITIVE NUMBER
006550  020127  000001          TOLER:  CMP     R1,#1           ;COMPARE DIFFERENCE WITH DESIRED TOLERANCE
006554  003403                          BLE     5$              ;BR, IF LOWER/EQUAL TO TOLERANCE

006556  010267  000020                  MOV     R2,SECND        ;STORE SECOND COUNT
006562  104053                          ERROR   53              ;CLOCK REPEATABILITY ERROR

006564  032777  000020  172246  5$:     BIT     #BIT4,@SWR      ;CLOCK TESTS ONLY?
006572  001404                          BEQ     TST24           ;BR IF NOT
006574  000167  003302                  JMP     $EOP            ;ELSE, JUMP TO END OF PASS ROUTINE

006600  000000          FIRST:  .WORD   0
006602  000000          SECND:  .WORD   0
```

```
                                        ;;*****************************************************************
                                        ;*TEST 24       TEST THAT XMIT INTERRUPTS ONLY WHEN ENABLED
                                        ;;*****************************************************************
006604  000004                  TST24:  SCOPE
006606  042777  000100  173536          BIC     #BIT6,@TCSR     ;CLEAR TRANSMIT INTERRUPT ENABLE
006614  017703  173542                  MOV     @TVECT,R3       ;SAVE XMIT VECTOR
006620  012777  006644  173534          MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
006626  105777  173520          1$:     TSTB    @TCSR           ;WAIT FOR DONE
006632  100375                          BPL     1$
006634  004767  003402                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
006640  000140                          .WORD   140
006642  000402                          BR      3$

006644  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
006646  104054                          ERROR   54
                                                                ;XMIT INTERRUPTS WITH INTERRUPT ENABLE CLEAR
006650  012777  006670  173504  3$:     MOV     #4$,@TVECT      ;SET XMIT VECTOR TO END OF TEST
006656  052777  000100  173466          BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
006664  000240                          NOP

006666  104055                          ERROR   55              ;XMIT DID NOT INTERRUPT

006670  042777  000100  173454  4$:     BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
006676  022626                          CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
006700  010377  173456                  MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
```

```
1593
1594                                    ;;*****************************************************************
1595                                    ;*TEST 25       TEST THAT XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
1596                                    ;;*****************************************************************
1597  006704  000004                    TST25:  SCOPE
1598  006706  042777  000100  173436            BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
1599  006714  004767  003322                    JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
1600  006720  000340                            .WORD   340
1601  006722  017703  173434                    MOV     @TVECT,R3       ;SAVE XMIT VECTOR
1602  006726  012777  006754  173426            MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
1603  006734  105777  173412          1$:       TSTB    @TCSR           ;WAIT FOR DONE
1604  006740  100375                            BPL     1$
1605  006742  052777  000100  173402            BIS     #BIT6,@TCSR     ;ENABLE INTERRUPT
1606  006750  000240                            NOP
1607  006752  000402                            BR      3$              ;CONTINUE TEST
1608
1609  006754  022626                  2$:       CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
1610  006756  104056                            ERROR   56
1611                                                                    ;XMIT INTERRUPTS AT PRIORITY=7
1612  006760  042777  000100  173364  3$:       BIC     #BIT6,@TCSR     ;CLEAR INTERRUPT ENABLE
1613  006766  012777  007006  173366            MOV     #4$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
1614  006774  004767  003242                    JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
1615  007000  000140                            .WORD   140
1616  007002  000240                            NOP
1617  007004  000402                            BR      5$              ;BR TO END OF TEST-NO INTERRUPT
1618
1619  007006  022626                  4$:       CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
1620  007010  104057                            ERROR   57
1621                                                                    ;XMIT INTERRUPT OCCURES WITH BIT6 CLEAR
1622  007012  010377  173344          5$:       MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
```

```
                                ;;*****************************************************************
                                ;*TEST 26       TEST TRANSMITTER FOR DOUBLE INTERRUPTS
                                ;;*****************************************************************
007016  000004                  TST26:  SCOPE
007020  042777  000100  173324          BIC     #BIT6,@TCSR     ;CLEAR INTERRUPT ENABLE
007026  017703  173330                  MOV     @TVECT,R3       ;SAVE XMIT VECTOR
007032  017704  173326                  MOV     @TPSW,R4        ;SAVE XMIT PSW VECTOR
007036  012777  007100  173316          MOV     #2$,@TVECT      ;SET UP XMIT VECTOR
007044  012777  000340  173312          MOV     #340,@TPSW      ;SET PIO 7 AFTER INTERRUPT
007052  004767  003164                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
007056  000140                          .WORD   140
007060  105777  173266          1$:     TSTB    @TCSR           ;WAIT FOR DONE
007064  100375                          BPL     1$
007066  052777  000100  173256          BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
007074  000240                          NOP

007076  104060                          ERROR   60
                                                                ;XMIT INTERRUPT DID NOT OCCUR
007100  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
007102  012777  007130  173252          MOV     #4$,@TVECT      ;POINT XMIT VECTOR TO ERROR
007110  004767  003126                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
007114  000140                          .WORD   140
007116  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPTS
007120  042777  000100  173224          BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
007126  000402                          BR      5$              ;BR TO END OF TEST

007130  022626                  4$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
007132  104061                          ERROR   61
                                                                ;XMIT RE-INTERRUPTED
007134  010377  173222          5$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
007140  010477  173220                  MOV     R4,@TPSW        ;RESTORE XMIT PSW VECTOR

                                ;;*****************************************************************
                                ;*TEST 27       TEST THAT XMIT INTERRUPT CLEARS WITH LOADING TBUF
                                ;;*****************************************************************
007144  000004                  TST27:  SCOPE
007146  042777  000100  173176          BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
007154  004767  003062                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 7
007160  000340                          .WORD   340
007162  017703  173174                  MOV     @TVECT,R3       ;SAVE XMIT VECTOR
007166  012777  007240  173166          MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR
007174  052777  000100  173150          BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
007202  005077  173146                  CLR     @TBUF           ;LOAD TBUF
007206  105777  173140          1$:     TSTB    @TCSR           ;WAIT FOR DONE (INTERRUPT)
007212  100375                          BPL     1$
007214  005077  173134                  CLR     @TBUF           ;FILL SECOND BUFFER TO RESET INT.
007220  004767  003016                  JSR     PC,WRPSW        ;ALLOW INTERRUPTS
007224  000140                          .WORD   140
007226  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPTS
007230  042777  000100  173114          BIC     #BIT6,@TCSR     :DISABLE INTERRUPTS
007236  000402                          BR      3$              ;BR TO END OF TEST

007240  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
007242  104062                          ERROR   62
```

```
 1679
 1680  007244  010377  173112          3$:     MOV     R3,@TVECT       ;LOADING TBUF DID NOT CLEAR INTEPRUPT.
 1681                                                                  ;RESTORE XMIT VECTOR
```

```
1682
1683                                    ;;*****************************************************************
1684                                    ;*TEST 30       TEST THAT RCVR ACTIVE (11) & DONE (7) SET & CLEAR PROPERLY
1685                                    ;;*****************************************************************
1686  007250  000004                   TST30:  SCOPE
1687  007252  000005                           RESET                           ;CLEAR EVERYTHING
1688  007254  052777  000004  173070           BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
1689  007262  005000                           CLR     R0                      ;CLEAR A TIMER
1690  007264  005077  173064                   CLR     @TBUF                   ;LOAD TRANSMIT BUFFER
1691  007270  032777  004000  173050   WACTV:  BIT     #BIT11,@RCSR            ;TEST RCVR ACTIVE BIT
1692  007276  001006                           BNE     2$                      ;BR IF SET
1693  007300  005200                           INC     R0                      ;INCREMENT TIMER IF NOT SET
1694  007302  001372                           BNE     WACTV                   ;CONTINUE WAIT IF TIME REMAINS
1695  007304  042777  000004  173040           BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
1696
1697  007312  104063                           ERROR   63                      ;RCVR ACTIVE DID NOT SET WHILE RECEIVING
1698
1699  007314  000005                   2$:     RESET
1700  007316  032777  004000  173022           BIT     #BIT11,@RCSR            ;VERIFY "INIT" CLEARS RCV ACTIVE
1701  007324  001401                           BEQ     3$
1702
1703  007326  104115                           ERROR   115                     ;INIT DID NOT CLEAR RCV ACTIVE
1704
1705  007330  005000                   3$:     CLR     R0                      ;CLEAR A TIMER
1706  007332  052777  000004  173012           BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
1707  007340  062700  000000           WT:     ADD     #0,R0                   ;WAIT AT LEAST ONE BIT TIME
1708  007344  005200                           INC     R0
1709  007346  001374                           BNE     WT
1710  007350  036777  174424  172770           BIT     BIT11,@RCSR             ;VERIFY RCV ACTIVE STILL CLEAR
1711  007356  001404                           BEQ     4$                      ;BR IF CLEAR
1712
1713  007360  042777  000004  172764           BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE WRAP
1714  007366  104116                           ERROR   116                     ;RCV ACTIVE WITHOUT "START" BIT
1715
1716  007370  005000                   4$:     CLR     R0                      ;CLEAR TIMER
1717  007372  005077  172756                   CLR     @TBUF                   ;LOAD TRANSMIT BUFFER
1718  007376  105777  172744           WDONE:  TSTB    @RCSR                   ;CHECK FOR RECEIVER DONE
1719  007402  100406                           BMI     5$                      ;BR, IF DONE
1720  007404  005200                           INC     R0                      ;INCREMENT TIMER, IF NOT DONE
1721  007406  001373                           BNE     WDONE                   ;CONTINUE WAIT IF TIME REMAINS
1722  007410  042777  000004  172734           BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
1723  007416  104064                           ERROR   64
1724                                                                           ;RECEIVER DONE NEVER SET
1725
1726  007420  032777  004000  172720   5$:     BIT     #BIT11,@RCSR            ;CHECK FOR RCVR ACTIVE CLEAR
1727  007426  001404                           BEQ     6$                      ;BR, IF CLEAR
1728  007430  042777  000004  172714           BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
1729  007436  104065                           ERROR   65
1730                                                                           ;RCVR ACTIVE DID NOT CLEAR WITH RCVR DONE
1731
1732  007440  000005                   6$:     RESET                           ;CLEAR DONE WITH RESET
1733  007442  105777  172700                   TSTB    @RCSR                   ;CHECK FOR DONE CLEAR
1734  007446  001404                           BEQ     7$
1735
1736  007450  042777  000004  172674           BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
1737  007456  104066                           ERROR   66
```

```
1738                                                    ;RESET DID NOT CLEAR RCVR DONE
1739
1740  007460  042777  000004  172660  7$:     BIC     #BIT2,@RCSR     ;CLEAR MAINTENANCE BIT
1741  007466  000400                  BR      TST31           ;BR TO NEXT TEST
```

```
                                ;;*****************************************************************
                                ;*TEST 31       TEST THAT RDR ENABLE CLEARS RECEIVER DONE FLAG
                                ;;*****************************************************************
007470  000004                  TST31:  SCOPE
007472  000005                          RESET                           ;CLEAR EVERYTHING
007474  052777  000001  172644          BIS     #BIT0,@RCSR             ;SET RDR ENABLE
007502  052777  000004  172642          BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
007510  005077  172640                  CLR     @TBUF                   ;LOAD TRANSMITTER
007514  105777  172626          1$:     TSTB    @RCSR                   ;WAIT FOR RECEIVER DONE
007520  100375                          BPL     1$
007522  032777  000001  172616          BIT     #BIT0,@RCSR             ;VERIFY RCV ACTIVE-CLEARED RDR ENABLE
007530  001401                          BEQ     2$                      ;BR IF CLEAR

007532  104117                          ERROR   117                     ;RDR ENABLE NOT CLEARED WITH RCV ACTIVE

007534  052777  000001  172604  2$:     BIS     #BIT0,@RCSR             ;CLEAR DONE BY SETTING RDR ENABLE
007542  105777  172600                  TSTB    @RCSR                   ;CHECK FOR DONE CLEAR
007546  001404                          BEQ     TST32                   ;BR, IF CLEAR TO NEXT TEST
007550  042777  000004  172574          BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
007556  104067                          ERROR   67
                                                                        ;SETTING RDR ENABLE DID NOT CLEAR RCVR DONE


                                ;;*****************************************************************
                                ;*TEST 32       TEST THAT READING RBUF CLEARS RECEIVER DONE
                                ;;*****************************************************************
007560  000004                  TST32:  SCOPE
007562  000005                          RESET                           ;CLEAR EVERYTHING
007564  052777  000004  172560          BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
007572  005077  172556                  CLR     @TBUF                   ;LOAD TRANSMITTER
007576  105777  172544          1$:     TSTB    @RCSR                   ;WAIT FOR RECEIVER DONE
007602  100375                          BPL     1$
007604  017700  172540                  MOV     @RBUF,R0                ;READ RECEIVE BUFFER
007610  042777  000004  172534          BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
007616  105777  172524                  TSTB    @RCSR                   ;CHECK FOR RECEIVE DONE CLEAR
007622  001401                          BEQ     TST33                   ;BR, IF CLEAR TO NEXT TEST
007624  104070                          ERROR   70
                                                                        ;READING RBUF DID NOT CLEAR RCVR DONE
```

```
1783
1784
1785                                    ;;*****************************************************************
1786                                    ;*TEST 33       TEST THAT RCVR INTERRUPTS ONLY WHEN ENABLED
1787                                    ;;*****************************************************************
1788   007626   000004                  TST33:  SCOPE
1789   007630   042777   000100  172514          BIC     #BIT6,@TCSR     ;DISABLE TRANSMIT INTERRUPTS
1790   007636   042777   000100  172502          BIC     #BIT6,@RCSR     ;DISABLE RECEIVER INTERRUPTS
1791   007644   052777   000004  172500          BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
1792   007652   017703   172500                  MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
1793   007656   012777   007714  172472          MOV     #2$,@RVECT      ;POINT RCV VECTOR TO ERROR REPORT
1794   007664   004767   002352                  JSR     PC,WRPSW        ;SET PSW TO PRIORITY 3
1795   007670   000140                          .WORD   140
1796   007672   005077   172456                  CLR     @TBUF           ;SEND A CHARACTER
1797   007676   105777   172444          1$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
1798   007702   100375                          BPL     1$
1799   007704   042777   000004  172440          BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
1800   007712   000405                          BR      3$              ;CONTINUE TEST
1801
1802   007714   042777   000004  172430  2$:     BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
1803   007722   022626                          CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
1804   007724   104071                          ERROR   71
1805                                                                    ;RECEIVER INTERRPUTS WITH INT. ENABLE CLEAR
1806
1807   007726   012777   007754  172422  3$:     MOV     #4$,@RVECT      ;POINT RCV VECTOR TO END OF TEST
1808   007734   052777   000100  172404          BIS     #BIT6,@RCSR     ;ENABLE RCV INTERRUPTS
1809   007742   000240                          NOP                     ;GIVE ANY INTERRUPTS TIME
1810   007744   042777   000004  172400          BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
1811   007752   104072                          ERROR   72
1812                                                                    ;RCVR DID NOT INTERRUPT
1813
1814   007754   042777   000100  172364  4$:     BIC     #BIT6,@RCSR     ;DISABLE INTERRUPTS
1815   007762   042777   000004  172362          BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
1816   007770   022626                          CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
1817   007772   010377   172360                  MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
```

```
1818
1819
1820                                    ;;*****************************************************************
1821                                    ;*TEST 34       TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED
1822                                    ;;*****************************************************************
1823  007776  000004                    TST34:  SCOPE
1824  010000  000005                            RESET                           ;CLEAR EVERYTHING
1825  010002  004767  002234                    JSR     PC,WRPSW                ;SET PSW TO PRIORITY 7
1826  010006  000340                            .WORD   340
1827  010010  017703  172342                    MOV     @RVECT,R3               ;SAVE RECEIVE VECTOR
1828  010014  012777  010054  172334            MOV     #2$,@RVECT              ;POINT RCVR VECTOR TO ERROR REPORT
1829  010022  052777  000004  172322            BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
1830  010030  005077  172320                    CLR     @TBUF                   ;SEND A CHARACTER
1831  010034  105777  172306            1$:     TSTB    @RCSR                   ;WAIT FOR RECEIVER DONE
1832  010040  100375                            BPL     1$
1833  010042  052777  000100  172276            BIS     #BIT6,@RCSR             ;ENABLE INTERRUPTS
1834  010050  000240                            NOP                             ;GIVE TIME FOR INTERRUPT
1835  010052  000405                            BR      3$                      ;CONTINUE TEST
1836  010054  042777  000004  172270  2$:      BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
1837  010062  022626                            CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
1838  010064  104073                            ERROR   73
1839                                                                            ;RCVR INTERRUPTS AT PRIORITY 7
1840
1841  010066  042777  000100  172252  3$:      BIC     #BIT6,@RCSR             ;CLEAR INTERRUPT ENABLE
1842  010074  012777  010122  172254            MOV     #4$,@RVECT              ;POINT RCVR VECTOR TO ERROR REPORT
1843  010102  004767  002134                    JSR     PC,WRPSW                ;SET PSW TO PRIORITY 3
1844  010106  000140                            .WORD   140
1845  010110  000240                            NOP                             ;GIVE TIME FOR ANY INTERRUPT
1846  010112  042777  000004  172232            BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
1847  010120  000405                            BR      5$                      ;BR TO END OF TEST, IF NO INTERRUPT
1848
1849  010122  042777  000004  172222  4$:      BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
1850  010130  022626                            CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
1851  010132  104074                            ERROR   74
1852                                                                            ;RCVR INTERRUPT REQUEST PASSED WITH BIT6 CLEAR
1853  010134  010377  172216            5$:     MOV     R3,@RVECT               ;RESTORE RECEIVE VECTOR
```

```
                                        ;;*****************************************************************
                                        ;*TEST 35       TEST RECEIVER FOR DOUBLE INTERRUPTS
                                        ;;*****************************************************************
010140  000004                  TST35:  SCOPE
010142  000005                          RESET                           ;CLEAR EVERYTHING
010144  017703  172206                  MOV     @RVECT,R3               ;SAVE RECEIVE VECTOR
010150  017704  172204                  MOV     @RPSW,R4                ;SAVE RECEIVE PSW VECTOR
010154  012777  010236  172174          MOV     #2$,@RVECT              ;POINT RCV VECTOR TO CONTINUE TEST
010162  012777  000340  172170          MOV     #340,@RPSW              ;SET PRIORITY TO 7 AFTER INTERRUPT
010170  004767  002046                  JSR     PC,WRPSW                ;SET PSW TO PRIORITY 3
010174  000140                          .WORD   140
010176  052777  000004  172146          BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
010204  005077  172144                  CLR     @TBUF                   ;SEND A CHARACTER
010210  105777  172132          1$:     TSTB    @RCSR                   ;WAIT FOR RCVR DONE
010214  100375                          BPL     1$
010216  042777  000004  172126          BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
010224  052777  000100  172114          BIS     #BIT6,@RCSR             ;ENABLE RCV INTERRUPTS
010232  000240                          NOP                             ;GIVE SOME TIME

010234  104075                          ERROR   75
                                                                        ;RCVR INTERRUPT DID NOT OCCUR

010236  022626                  2$:     CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
010240  012777  010276  172110          MOV     #3$,@RVECT              ;POINT RCV VECTOR TO ERROR REPORT
010246  004767  001770                  JSR     PC,WRPSW                ;RESET PSW TO PRIORITY 3
010252  000140                          .WORD   #140
010254  000240                          NOP                             ;GIVE SOME TIME
010256  042777  000100  172062          BIC     #BIT6,@RCSR             ;CLEAR INTERRUPT ENABLE
010264  010377  172066                  MOV     R3,@RVECT               ;RESTORE RECEIVE VECTOR
010270  010477  172064                  MOV     R4,@RPSW                ;RESTORE RECEIVE PSW VECTOR
010274  000402                          BR      4$                      ;BR TO END OF TEST

010276  022626                  3$:     CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
010300  104076                          ERROR   76
                                                                        ;RECEIVER RE-INTERRUPTED
010302  010377  172050          4$:     MOV     R3,@RVECT               ;RESTORE RECEIVE VECTOR
```

```
                                ;;*****************************************************************
                                ;*TEST 36       TEST THAT RCVR INTERRUPT CLEARS BY READING RBUF
                                ;;*****************************************************************
010306  000004                  TST36:  SCOPE
010310  000005                          RESET                           ;CLEAR EVERYTHING
010312  004767  001724                  JSR     PC,WRPSW                ;SET PSW PRIORITY TO 7
010316  000340                          .WORD   340
010320  017703  172032                  MOV     @RVECT,R3               ;SAVE RECEIVE VECTOR
010324  012777  010412  172024          MOV     #2$,@RVECT              ;POINT RCV VECTOR TO ERROR REPORT
010332  052777  000100  172006          BIS     #BIT6,@RCSR             ;SET RCVR INTERRUPT ENABLE
010340  052777  000004  172004          BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
010346  005077  172002                  CLR     @TBUF                   ;SEND A CHARACTER
010352  105777  171770          1$:     TSTB    @RCSR                   ;WAIT FOR DONE (INTERRUPT)
010356  100375                          BPL     1$
010360  042777  000004  171764          BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
010366  005077  171756                  CLR     @RBUF                   ;READ RBUF TO CLEAR PENDING INTERRPUT
010372  004767  001644                  JSR     PC,WRPSW                ;SET PSW TO PRIORITY 3
010376  000140                          .WORD   140
010400  000240                          NOP                             ;ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
010402  042777  000100  171736          BIC     #BIT6,@RCSR             ;NO INTERRUPT-CLEAR INT. ENABLE
010410  000402                          BR      3$

010412  022626                  2$:     CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
010414  104077                          ERROR   77                      ;READING RBUF DID NOT CLEAR INTERRUPT

010416  010377  171734          3$:     MOV     R3,@RVECT               ;RESTORE RECEIVE VECTOR
```

```
                                ;;*********************************************************************
                                ;*TEST 37       TEST THAT RESET CLEARS RECEIVE INTERRUPT
                                ;;*********************************************************************
010422  000004                  TST37:  SCOPE
010424  000005                          RESET                           ;CLEAR EVERYTHING
010426  004767  001610                  JSR     PC,WRPSW                ;SET PSW TO PRIORITY 7
010432  000340                          .WORD   340
010434  017703  171716                  MOV     @RVECT,R3               ;SAVE RECEIVE VECTOR
010440  012777  010520  171710          MOV     #2$,@RVECT              ;POINT RCV VECTOR TO ERROR REPORT
010446  052777  000100  171672          BIS     #BIT6,@RCSR             ;SET RCV INTERRUPT ENABLE
010454  052777  000004  171670          BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
010462  012777  000377  171664          MOV     #377,@TBUF              ;SEND AN ALL 1'S CHARACTER
010470  105777  171652          1$:     TSTB    @RCSR                   ;WAIT FOR RCV DONE
010474  100375                          BPL     1$
010476  000005                          RESET                           ;CLEAR RCV INTERRUPT & RBUF
010500  004767  001536                  JSR     PC,WRPSW                ;SET PSW TO PRIORITY 3
010504  000140                          .WORD   140
010506  000240                          NOP                             ;ALLOW TIME FOR AN ERRONEOUS INTERRUPT
010510  042777  000100  171630          BIC     #BIT6,@RCSR             ;NO INTERRUPT-CLEAR INT. ENABLE
010516  000402                          BR      3$                      ;CONTINUE TEST

010520  022626                  2$:     CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
010522  104100                          ERROR   100                     ;RESET DID NOT CLEAR RCVR INTERRUPT

010524  010377  171626          3$:     MOV     R3,@RVECT               ;RESTORE RECEIVE VECTOR
```

```
1949
1950
1951                                    ;;*****************************************************************
1952                                    ;*TEST 40       TEST THAT THE "OR" ERROR (BIT14) & "ERROR" (BIT15) CAN BE SET
1953                                    ;;*****************************************************************
1954  010530  000004                    TST40:  SCOPE
1955  010532  032777  002000  170300            BIT     #BIT10,@SWR     ;IS THIS TEST ENABLED
1956  010540  001432                            BEQ     TST41           ;IF NOT ENABLED, BR TO NEXT TEST
1957  010542  000005                            RESET                   ;CLEAR EVERYTHING
1958  010544  052777  000004  171600            BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
1959  010552  012700  000003                    MOV     #3,R0           ;SET CHARACTER COUNT TO SEND 3 CHAR.
1960  010556  005077  171572            1$:     CLR     @TBUF           ;LOAD TRANSMIT BUFFER
1961  010562  105777  171564            2$:     TSTB    @TCSR           ;WAIT FOR TRANSMIT DONE
1962  010566  100375                            BPL     2$
1963  010570  005300                            DEC     R0              ;DECREMENT CHARACTER COUNT
1964  010572  001371                            BNE     1$              ;BR IF ALL CHARACTERS NOT TRANSMITTED
1965  010574  042777  000004  171550            BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
1966  010602  032777  040000  171540            BIT     #BIT14,@RBUF    ;TEST FOR "OR" ERROR FLAG
1967  010610  001001                            BNE     3$              ;BR, IF SET
1968  010612  104101                            ERROR   101
1969                                                                    ;"OR" ERROR FLAG DID NOT SET
1970
1971  010614  032777  100000  171526  3$:       BIT     #BIT15,@RBUF    ;TEST "ERROR" FLAG
1972  010622  001001                            BNE     4$              ;BR, IF SET
1973  010624  104102                            ERROR   102
1974                                                                    ;"ERROR" FLAG DID NOT SET WITH "OR" FLAG
1975  010626                            4$:
```

```
1976
1977
1978                                    ;;*****************************************************************
1979                                    ;*TEST 41       TEST THAT BREAK TRANSMITS ALL ZEROES
1980                                    ;;*****************************************************************
1981  010626  000004                    TST41:  SCOPE
1982  010630  032777  000400  170202            BIT     #BIT8,@SWR      ;IS BREAK FUNCTION TEST ENABLED?
1983  010636  001444                            BEQ     TST42           ;BR TO NEXT TEST, IF NOT ENABLED
1984  010640  000005                            RESET                   ;CLEAR EVERYTHING
1985  010642  052777  000004  171502            BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
1986  010650  012777  177777  171476            MOV     #-1,@TBUF       ;TRANSMIT ALL ONES TO RCVR
1987  010656  105777  171464            1$:     TSTB    @RCSR           ;WAIT FOR RCVR DONE
1988  010662  100375                            BPL     1$
1989  010664  005077  171460                    CLR     @RBUF           ;CLEAR DONE (LEAVING ALL ONES IN RBUF)
1990  010670  052777  000001  171454            BIS     #BIT0,@TCSR     ;TRANSMIT BREAK
1991  010676  005000                            CLR     R0              ;CLEAR A TIMER
1992  010700  117767  171442  170120  2$:      MOVB    @RCSR,$BDDAT    ;WAIT FOR RCVR DONE
1993  010706  100406                            BMI     CONT41          ;BR IF DONE
1994  010710  005200                            INC     R0              ;IF NOT, INCREMENT TIMER
1995  010712  001372                            BNE     2$              ;BR IF TIME REMAINS
1996
1997  010714  042777  000005  171430            BIC     #BIT0!BIT2,@TCSR  ;CLEAR MAINTENANCE & BREAK BITS
1998  010722  104103                            ERROR   103             ;BREAK DID NOT TRANSMIT ANYTHING
1999
2000  010724  105777  171420            CONT41: TSTB    @RBUF           ;CHECK RECEIVE BUFFER FOR ZERO
2001  010730  001404                            BEQ     3$              ;BR, IF ZERO
2002  010732  042777  000005  171412            BIC     #BIT0!BIT2,@TCSR  ;CLEAR MAINTENANCE & BREAK BITS
2003
2004  010740  104103                            ERROR   103             ;BREAK DID NOT TRANSMIT ALL ZEROES
2005
2006  010742  042777  000005  171402  3$:       BIC     #BIT0!BIT2,@TCSR ;CLEAR MAINTENANCE & BREAK BITS
```

```
2007
2008
2009                                    ;;*****************************************************************
2010                                    ;*TEST 42        TEST THAT "FR" ERROR CAN BE SET DURING BREAK
2011                                    ;;*****************************************************************
2012  010750  000004                    TST42:  SCOPE
2013  010752  032777  002000  170060            BIT     #BIT10,@SWR     ;IS THE "TEST ERROR FLAGS" BIT SET
2014  010760  001435                            BEQ     TST43           ;BR TO NEXT TEST, IF NOT SET
2015  010762  032777  000400  170050            BIT     #BIT8,@SWR      ;IS BREAK FUNCTION ENABLED
2016  010770  001431                            BEQ     TST43           ;BR TO NEXT TEST, IF NOT SET
2017  010772  000005                            RESET                   ;CLEAR EVERYTHING
2018  010774  052777  000004  171350            BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
2019  011002  052777  000001  171342            BIS     #BIT0,@TCSR     ;SEND BREAK
2020  011010  005077  171340                    CLR     @TBUF           ;TRANSMIT A CHARACTER TO TIME BREAK
2021  011014  105777  171326          1$:       TSTB    @RCSR           ;WAIT FOR RCVR DONE
2022  011020  100375                            BPL     1$
2023  011022  042777  000005  171322            BIC     #BIT0!BIT2,@TCSR ;CLEAR MAINTENANCE & BREAK BITS
2024  011030  032777  020000  171312            BIT     #BIT13,@RBUF    ;CHECK FOR FRAMING ERROR FLAG
2025  011036  001001                            BNE     2$              ;BR, IF SET
2026
2027  011040  104104                            ERROR   104
2028                                                                    ;BREAK DID NOT SET FRAMING ERROR
2029  011042  032777  100000  171300  2$:       BIT     #BIT15,@RBUF    ;TEST "ERROR"FLAG
2030  011050  001001                            BNE     3$              ;BR, IF SET
2031
2032  011052  104114                            ERROR   114
2033                                                                    ;"ERROR" FLAG DID NOT SET WITH "OR" FLAG
2034  011054                          3$:
```

```
                                ;;*****************************************************************
                                ;*TEST 43       TEST DATA PATH FROM TRANSMITTER TO RECEIVER USING MAINTENANCE WRAP
                                ;;*****************************************************************
011054  000004                  TST43:  SCOPE

011056  000005                          RESET                   ;CLEAR EVERYTHING
011060  005001                          CLR     R1              ;CLEAR REGISTER FOR TEST DATA
011062  052777  000004  171262          BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
                                                                ;TRANSMIT A BINARY COUNT PATTERN - UP
                                                                ;TO THE BIT POSITION INDICATED BY THE
                                                                ;CONTENTS OF LOCATION "$USWR"
011070  005201                  1$:     INC     R1              ;INCREMENT THE TEST DATA
011072  010177  171256                  MOV     R1,@TBUF        ;XMIT A CHARACTER
011076  105777  171244          2$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
011102  100375                          BPL     2$
011104  017702  171240                  MOV     @RBUF,R2        ;GET RECEIVED CHARACTER
011110  043701  001112                  BIC     @#$USWR,R1      ;CLEAR LOWEST UNUSED DATA BIT POSITITON IN TEST DATA
011114  020102                          CMP     R1,R2           ;COMPARE DATA
011116  001003                          BNE     3$              ;BR, IF NON-COMPARE
011120  105701                          TSTB    R1              ;TEST XMIT DATA FOR ZERO
011122  001411                          BEQ     4$              ;BR, IF FINISHED
011124  000761                          BR      1$              ;CONTINUE IF NOT
011126  010167  167672          3$:     MOV     R1,$GDDAT       ;STORE THE EXPECTED DATA
011132  010267  167670                  MOV     R2,$BDDAT       ;STORE RECEIVED DATA
011136  042777  000004  171206          BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
011144  104105                          ERROR   105             ;DATA COMPARE DATA

011146  042777  000004  171176  4$:     BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT



                                ;;*****************************************************************
                                ;*TEST 44       TEST DATA PATHS USING WRAP CABLE
                                ;;*****************************************************************
011154  000004                  TST44:  SCOPE
011156  032777  000200  167654          BIT     #BIT7,@SWR      ;IS THIS TEST ENABLED
011164  001432                          BEQ     TST45           ;BR, IF NOT
011166  000005                          RESET                   ;CLEAR EVERYTHING
011170  005001                          CLR     R1              ;CLEAR REGISTER FOR TEST DATA
                                                                ;TRANSMIT A BINARY COUNT PATTERN - UP
                                                                ;TO THE BIT POSITION INDICATED BY THE
                                                                ;CONTENTS OF LOCATION "$USWR"
011172  105201                  1$:     INCB    R1              ;INCREMENT THE TEST DATA
011174  010177  171154                  MOV     R1,@TBUF        ;XMIT A CHARACTER
011200  005000                          CLR     R0              ;CLEAR A TIMER
011202  105777  171140          2$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
011206  100403                          BMI     3$              ;BR IF DONE
011210  005200                          INC     R0              ;INCREMENT TIMER IF NOT
011212  001373                          BNE     2$              ;BR IF TIME REMAINS

011214  104064                          ERROR   64              ;RECEIVER DONE NOT SET

011216  017702  171126          3$:     MOV     @RBUF,R2        ;GET RECEIVED CHARACTER
011222  043701  001112                  BIC     @#$USWR,R1      ;CLEAR LOWEST UNUSED DATA BIT POSITITON IN TEST DATA
```

```
 2091  011226  020102                        CMP     R1,R2           ;COMPARE DATA
 2092  011230  001003                        BNE     4$              ;BR, IF NON-COMPARE
 2093  011232  105701                        TSTB    R1              ;TEST XMIT DATA FOR ZERO
 2094  011234  001406                        BEQ     TST45           ;BR, IF FINISHED
 2095  011236  000755                        BR      1$              ;CONTINUE IF NOT
 2096  011240  010167  167560          4$:   MOV     R1,$GDDAT       ;STORE EXPECTED DATA
 2097  011244  010267  167556                MOV     R2,$BDDAT       ;STORE RECEIVED DATA
 2098
 2099  011250  104106                        ERROR   106             ;DATA COMPARE ERROR WITH WRAP CABLE
```

```
                                ;;*****************************************************************
                                ;*TEST 45       TEST DL11-W LOGIC BY EXERCISING THE XMIT,RECEIVE, & CLOCK(IF AVAILABLE)
                                ;;*****************************************************************
011252  000004                 TST45:  SCOPE
011254  000005                         RESET                           ;CLEAR EVERYTHING
011256  004767  000760                 JSR     PC,WRPSW                ;SET PRIORITY TO 7
011262  000340                         .WORD   340
011264  017703  171072                 MOV     @TVECT,R3               ;SAVE XMIT VECTOR
011270  017704  171062                 MOV     @RVECT,R4               ;SAVE RECEIVE VECTOR
011274  017705  171110                 MOV     @RTCVT,R5               ;SAVE CLOCK VECTOR
011300  017767  171060  000426         MOV     @TPSW,STPSW             ;SAVE XMIT PSW VECTOR
011306  017767  171046  000422         MOV     @RPSW,SRPSW             ;SAVE RECEIVE PSW VECTOR
011314  017767  171072  000416         MOV     @RTCPSW,SCPSW           ;SAVE CLOCK PSW VECTOR
011322  012777  011650  171032         MOV     #XMIT,@TVECT            ;POINT TRANSMIT VECTOR TO TRANSMIT ROUTINE
011330  005077  171030                 CLR     @TPSW                   ;ALLOW INTERRUPTS AFTER XMIT INTERRUPT
011334  012777  011704  171014         MOV     #RCV,@RVECT             ;POINT RECEIVE VECTOR TO RECEIVE ROUTINE
011342  005077  171012                 CLR     @RPSW                   ;ALLOW INTERRPUTS AFTER RCVR INTERRUPT
011346  005767  170764                 TST     CTSTFL                  ;IS CONSOLE UNDER TEST?
011352  001414                         BEQ     1$                      ;IF NOT SKIP CLOCK SET UP
011354  032777  000100  167456         BIT     #BIT6,@SWR              ;IF YES, ARE CLOCK TEST DISABLED?
011362  001010                         BNE     1$                      ;IF YES, SKIP CLOCK SET UP
011364  012777  011720  171016         MOV     #CLK,@RTCVT             ;POINT VECTOR TO CLOCK INTERRUPT ROUTINE
011372  005077  171014                 CLR     @RTCPSW                 ;ALLOW INTERRUPTS AFTER CLOCK INTERRUPT
011376  052777  000100  171002         BIS     #BIT6,@LKS              ;ENABLE CLOCK INTERRUPTS
011404  052777  000004  170740  1$:     BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
011412  052777  000100  170732         BIS     #BIT6,@TCSR             ;ENABLE TRANSMIT INTERRUPTS
011420  052777  000100  170720         BIS     #BIT6,@RCSR             ;ENABLE RECEIVE INTERRUPTS
011426  005067  000276                 CLR     XMTCNT                  ;CLEAR XMIT INTERRUPT COUNTER
011432  005067  000270                 CLR     RCVCNT                  ;CLEAR RCV INTERRUPT COUNTER
011436  005001                         CLR     R1                      ;CLEAR A REGISTER FOR TEST DATA USE
011440  005000                         CLR     R0                      ;CLEAR TIMER
011442  012702  011742                 MOV     #BUF,R2                 ;POINT R2 TO RECEIVE DATA STORAGE
011446  005077  170702                 CLR     @TBUF                   ;SEND FIRST CHARACTER
011452  004767  000564                 JSR     PC,WRPSW                ;SET PSW TO PRIORITY 3
011456  000140                         .WORD   140
                                                                        ;WAIT FOR INTERRUPTS
011460  062700  000000         2$:     ADD     #0,R0                   ;ADD INSTRUCTIONS ARE USED TO LENGTHEN LOOP TIME
011464  062700  000001                 ADD     #1,R0                   ;   TO COVER THE SLOWEST BAUD RATE ON THE FASTEST CPU
011470  001373                         BNE     2$
011472  032777  000100  170652         BIT     #BIT6,@TCSR             ;FINISHED ENTIRE TRANSMISSION
011500  001402                         BEQ     3$                      ;BR, IF INTERRPUTS ARE DISABLED (FINISHED)
011502  000005                         RESET                           ;CLEAR EVERYTHING

011504  104107                         ERROR   107                     ;TRANSMIT INTERRUPT TIMEOUT IN MAIN. DATA TEST

011506  026767  000216  000212 3$:     CMP     XMTCNT,RCVCNT           ;COMPARE THE NUMBER OF INTERRUPTS
011514  001402                         BEQ     4$                      ;BR, IF EQUAL
011516  000005                         RESET                           ;CLEAR EVERYTHING

011520  104110                         ERROR   110                     ;RECEIVER DID NOT GET FULL TRANSMISSION
                                                                        ;    IF RCVCNT=0, NO DATA RECEIVED
                                                                        ;    IF RCVCNT=?, THEN (XMTCNT-RCVCNT)
                                                                        ;    EQUALS THE NO. OF INTERRUPTS LOST.
```

```
2156
2157
2158  011522  005767  170610          4$:     TST     CTSTFL          ;IS CONCOLE UNDER TEST?
2159  011526  001410                          BEQ     5$              ;IF NOT, SKIP CLOCK COUNT CHECK
2160  011530  032777  000100  167302          BIT     #BIT6,@SWR      ;IF YES, ARE CLOCK TESTS DISABLED?
2161  011536  001004                          BNE     5$              ;IF YES, SKIP CLOCK COUNT CHECK
2162  011540  005767  000166                  TST     CLKCNT          ;CHECK FOR AT LESST ONE CLOCK INTERRUPT
2163  011544  001001                          BNE     5$              ;BR IF INTERRUPTS OCCURRED
2164
2165  011546  104113                          ERROR   113             ;NO CLOCK INTERRUPTS IN EXERCISER
2166
2167  011550  000005          5$:             RESET                   ;CLEAR EVERYTHING
2168  011552  012700  011742                  MOV     #BUF,R0         ;LOAD RECEIVED DATA POINTER TO R0
2169  011556  005001                          CLR     R1              ;SET UP REGISTER FOR COMPARISON
2170  011560  022001          COMP:           CMP     (R0)+,R1        ;COMPARE XMIT & RCV DATA
2171  011562  001005                          BNE     6$              ;BR, IF NOT EQUAL
2172  011564  105201                          INCB    R1              ;INCREMENT COMPARE DATA
2173  011566  032701  000040                  BIT     #BIT5,R1        ;FINISHED CHECKING RECEIVED DATA?
2174  011572  001772                          BEQ     COMP            ;BR, IF NOT FINISHED
2175  011574  000405                          BR      7$              ;BR TO END OF TEST
2176
2177  011576  014067  167224          6$:     MOV     -(R0),$BDDAT    ;STORE BAD DATA FOR ERROR REPORT
2178  011602  010167  167216                  MOV     R1,$GDDAT       ;STORE GOOD DATA FOR ERROR REPORT
2179  011606  104111                          ERROR   111             ;DATA COMPARE ERROR IN EXERCISER
2180
2181  011610  010377  170546          7$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
2182  011614  010477  170536                  MOV     R4,@RVECT       ;RESTORE RECEIVE VECTOR
2183  011620  010577  170564                  MOV     R5,@RTCVT       ;RESTORE CLOCK VECTOR
2184  011624  016777  000104  170532          MOV     STPSW,@TPSW     ;RESTORE XMIT PSW VECTOR
2185  011632  016777  000100  170520          MOV     SRPSW,@RPSW     ;RESTORE RECEIVE PSW VECTOR
2186  011640  016777  000074  170544          MOV     SCPSW,@RTCPSW   ;RESTORE CLOCK PSW VECTOR
2187  011646  000501                          BR      ENDEV           ;BR TO END OF DEVICE TEST ROUTINE
2188
2189  011650  005267  000054          XMIT:   INC     XMTCNT          ;INCREMENT XMIT INTERRUPT COUNTER
2190  011654  105201                          INCB    R1              ;INCREMENT TEST DATA
2191  011656  032701  000040                  BIT     #BIT5,R1        ;SEND DATA PATTERN FROM 00 --> 37
2192  011662  001404                          BEQ     XCONT           ;BR, IF MORE DATA TO BE SENT
2193  011664  042777  000100  170460          BIC     #BIT6,@TCSR     ;CLEAR XMIT INTERRUPT ENABLE
2194  011672  000402                          BR      XRET            ;RETURN, WITHOUT SENDING ANY MORE DATA
2195  011674  110177  170454          XCONT:  MOVB    R1,@TBUF        ;SEND NEW CHARACTER
2196  011700  005000                  XRET:   CLR     R0              ;CLEAR TIMER
2197  011702  000002                          RTI                     ;RETURN
2198
2199  011704  017722  170440          RCV:    MOV     @RBUF,(R2)+     ;STORE RECEIVED DATA
2200  011710  005267  000012                  INC     RCVCNT          ;INCREMENT RCV INTERRUPT COUNTER
2201  011714  005000                          CLR     R0              ;CLEAR TIMER
2202  011716  000002                          RTI                     ;RETURN
2203
2204  011720  005267  000006          CLK:    INC     CLKCNT          ;INCREMENT CLOCK INTERRUPT COUNT
2205  011724  000002                          RTI                     ;RETURN
2206
2207  011726  000000          RCVCNT: .WORD   0
2208  011730  000000          XMTCNT: .WORD   0
2209  011732  000000          CLKCNT: .WORD   0
2210  011734  000000          STPSW:  .WORD   0
2211  011736  000000          SRPSW:  .WORD   0
```

```
 2212  011740  000000                SCPSW:  .WORD   0
 2213  011742  000044                BUF:    .BLKW   44
 2214
```

```
                                        ;END OF DEVICE PASS ROUTINE

012052  005067  166724          ENDEV:  CLR     $TSTNM          ;CLEAR TEST NO. COUNT FOR SCOPE ROUTINE
012056  005267  167014                  INC     $DEVCT          ;INCREMENT DEVICE COUNTER
012062  026767  170254  167006          CMP     TMP2,$DEVCT     ;ALL DEVICES TESTED
012070  001404                          BEQ     $EOP            ;BR, IF YES
012072  005067  170240                  CLR     CTSTFL          ;CLEAR CONSOLE UNDER TEST FLAG
012076  000167  171376                  JMP     TSTDEV          ;GO TEST NEXT DEVICE



                                        .SBTTL  END OF PASS ROUTINE

                                        ;;**************************************************************
                                        ;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;*IF THERES A MONITOR GO TO IT
                                        ;*IF THERE ISN'T JUMP TO GOAGIN

012102                                  $EOP:
012102  000004                                  SCOPE
012104  005067  166672                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
012110  005267  166760                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
012114  042767  100000  166752          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
012122  005327                          DEC     (PC)+           ;;LOOP?
012124  000001                  $EOPCT: .WORD   1
012126  003015                          BGT     $DOAGN          ;;YES
012130  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
012132  000001                  $ENDCT: .WORD   1
012134  012124                          $EOPCT
012136  104401  012172                  TYPE    ,ENDMG          ;;TYPE "END PASS"
012142  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
012146  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
012150  000005                          RESET                   ;;CLEAR THE WORLD
012152  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
012154  000240                          NOP                     ;;SAVE ROOM
012156  000240                          NOP                     ;;FOR
012160  000240                          NOP                     ;;ACT11
012162                          $DOAGN:
012162  000137                          JMP     @(PC)+          ;;RETURN
012164  012206                  $RTNAD: .WORD   GOAGIN
012166     377     377     000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
                012172                  .EVEN
012172  005015  047105  020104  ENDMG:  .ASCIZ  <CR><LF>/END PASS /
012200  040520  051523  000040
```

```
2260
2261  012206  005067  166664          GOAGIN: CLR     $DEVCT          ;CLEAR DEVICE COUNT
2262  012212  022767  000001  170122          CMP     #1,TMP2         ;IS THERE ONLY ONE DEVICE UNDER TEST?
2263  012220  001004                          BNE     RSTRT           ;BR, IF NOT
2264  012222  012706  001000                  MOV     #1000,SP                ;RESET STACK POINTER
2265  012226  000167  171370                  JMP     TST1            ;GO DO ANOTHER PASS
2266
2267  012232  005067  166642          RSTRT:  CLR     $UNIT           ;CLEAR UNIT NUMBER
2268  012236  000167  171162                  JMP     BEGIN
2269
2270  012242  011646                  WRPSW:  MOV(SP),-(SP)           ;COPY RETUN PC
2271  012244  013616                          MOV     @(SP)+,(SP)     ;MOVE NEW PSW TO STACK
2272  012246  062746  000002                  ADD     #2,-(SP)        ;ADJUST JSR RETURN
2273  012252  000002                          RTI                     ;POP RETURN PC & NEW PSW
2274
2275                                  ;SUBROUTINE TO REPORT UNEXPECTED OR ERRONEOUS TRAPS OR INTERRUPTS
2276
2277  012254  012600                  CATCH:  MOV     (SP)+,R0        ;GET ADDRESS OF TRAP VECTOR + 4
2278  012256  162700  000004                  SUB     #4,R0           ;ADJUST TO POINT TO TRAP ADDRESS
2279  012262  010067  000014                  MOV     R0,BDVECT       ;STORE TRAP OR INTERRUPT ADDRESS
2280  012266  016667  000002  000004          MOV     2(SP),OLDPC     ;GET PC WHERE TRAP OR INTERRUPT OCCURRED
2281  012274  104112                          ERROR   112             ;REPORT ERROR
2282
2283  012276  000000                          HALT                    ;PROGRAM MUST BE RESTARTED AT THIS POINT
2284  012300  000000                  OLDPC:  .WORD   0
2285  012302  000000                  BDVECT: .WORD   0
2286
```

```
                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                ;*SAVE THE ERROR ITEM NUMBER AND ADDRESS OF THE ERROR CALL
                                ;*AND GO TO $ERRTYP ON ERROR
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW15=1         HALT ON ERROR
                                ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                ;*SW09=1         LOOP IN ERROR
                                ;*CALL
                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
                                ;;*****************************************************************

012304                          $ERROR:
012304  105267  166473          7$:     INCB    $ERFLG          ;SET THE ERROR FLAG
012310  001775                          BEQ     7$              ;DON'T LET FLAG GO TO ZERO
012312  016777  166464  166522          MOV     $TSTNM,@DISPLAY ;DISPLAY TEST NUMBER AND ERROR FLAG
012320  005267  166466                  INC     $ERTTL          ;INCREMENT ERROR COUNT
012324  011667  166466                  MOV     (SP),$ERRPC     ;GET ADDRESS OF ERROR INSTRUCTION
012330  162767  000002  166460          SUB     #2,$ERRPC
012336  117767  166454  166450          MOVB    @$ERRPC,$ITEMB  ;STRIP AND SAVE THE ERROR ITEM CODE
012344  032777  020000  166466          BIT     #BIT13,@SWR     ;SKIP TYPEOUT IF SET
012352  001004                          BNE     20$             ;SKIP TYPEOUTS
012354  004767  000106                  JSR     PC,$ERRTYP      ;GO TO USER ERROR ROUTINE
012360  104401  001063                  TYPE    ,$CRLF
012364                          20$:
012364  122767  000001  166514          CMPB    #APTENV,$ENV    ;RUNNING IN APT MODE
012372  001007                          BNE     2$              ;NO, SKIP APT ERROR REPORT
012374  116767  166414  000004          MOVB    $ITEMB,21$      ;SET ITEM NUMBER AS ERROR NUMBER
012402  004767  000562                  JSR     PC,$ATY4        ;REPORT FATAL ERROR TO APT
012406     000                  21$:    .BYTE   0
012407     000                          .BYTE   0
012410  000777                  22$:    BR      22$             ;APT ERROR LOOP
012412  005777  166422          2$:     TST     @SWR            ;HALT ON ERROR
012416  100001                          BPL     3$              ;SKIP IF CONTINUE
012420  000000                          HALT                    ;HALT ON ERROR!
012422  104406                  3$:     CKSWR                   ;TEST FOR CHANGE IN SOFT-SWR
012424  032777  001000  166406          BIT     #BIT09,@SWR     ;LOOP ON ERROR SWITCH SET?
012432  001402                          BEQ     4$              ;BR IF NO
012434  016716  166350                  MOV     $LPERR,(SP)     ;FUDGE RETURN FOR LOOPING
012440  005767  166414          4$:     TST     $ESCAPE         ;CHECK FOR AN ESCAPE ADDRESS
012444  001402                          BEQ     5$              ;BR IF NONE
012446  016716  166406                  MOV     $ESCAPE,(SP)    ;FUDGE RETURN ADDRESS FOR ESCAPE
012452                          5$:
012452  022737  012152  000042          CMP     #$ENDAD,@#42    ;ACT-11 AUTO-ACCEPT?
012460  001001                          BNE     6$              ;BR IF NO
012462  000000                          HALT                    ;YES
012464                          6$:
012464  000002                          RTI                     ;RETURN
```

```
                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

                                ;;*****************************************************************
                                ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
                                ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
                                ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

012466                          $ERRTYP:
012466  104401  001063                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
012472  010046                        MOV     R0,-(SP)        ;;SAVE R0
012474  005000                        CLR     R0              ;;PICKUP THE ITEM INDEX
012476  153700  001014                BISB    @#$ITEMB,R0
012502  001004                        BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                                ;;TYPE THE PC OF THE ERROR
012504  016746  166306                MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                ;;ERROR ADDRESS
012510  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
012512  000426                        BR      6$              ;;GET OUT
012514  005300                1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
012516  006300                        ASL     R0              ;;      WORK FOR THE ERROR TABLE
012520  006300                        ASL     R0
012522  006300                        ASL     R0
012524  062700  001146                ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
012530  012067  000004                MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
012534  001404                        BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
012536  104401                        TYPE                    ;;TYPE THE "ERROR MESSAGE"
012540  000000                2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
012542  104401  001063                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
012546  012067  000004        3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
012552  001404                        BEQ     5$              ;;SKIP TYPEOUT IF 0
012554  104401                        TYPE                    ;;TYPE THE "DATA HEADER"
012556  000000                4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
012560  104401  001063                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
012564  011000                5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
012566  001004                        BNE     7$              ;;GO TYPE THE DATA
012570  012600                6$:     MOV     (SP)+,R0        ;;RESTORE R0
012572  104401  001063                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
012576  000207                        RTS     PC              ;;RETURN
012600                          7$:
012600  013046                        MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
012602  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
012604  005710                        TST     (R0)            ;;IS THERE ANOTHER NUMBER?
012606  001770                        BEQ     6$              ;;BR IF NO
012610  104401  012616                TYPE    ,8$             ;;TYPE TWO(2) SPACES
012614  000771                        BR      7$              ;;LOOP
012616  020040      000       8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
012622                                .EVEN
```

```
2386
2387                                    .SBTTL  POWER DOWN AND UP ROUTINES
2388                                    ;;**********************************************************************
2389                                    ;*POWER DOWN ROUTINE
2390                                    ;;**********************************************************************
2391  012622  012737  012762  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;SET FOR FAST UP
2392  012630  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;PRIO:7
2393  012636  010046                          MOV     R0,-(SP)        ;PUSH R0 ON STACK
2394  012640  010146                          MOV     R1,-(SP)        ;PUSH R1 ON STACK
2395  012642  010246                          MOV     R2,-(SP)        ;PUSH R2 ON STACK
2396  012644  010346                          MOV     R3,-(SP)        ;PUSH R3 ON STACK
2397  012646  010446                          MOV     R4,-(SP)        ;PUSH R4 ON STACK
2398  012650  010546                          MOV     R5,-(SP)        ;PUSH R5 ON STACK
2399  012652  017746  166162                  MOV     @SWR,-(SP)      ;PUSH @SWR ON STACK
2400  012656  010667  000104                  MOV     SP,$SAVR6       ;SAVE SP
2401  012662  012737  012674  000024          MOV     #$PWRUP,@#PWRVEC ;SET UP VECTOR
2402  012670  000000                          HALT
2403  012672  000776                          BR      .-2             ;HANG UP
2404
2405
2406                                    ;;**********************************************************************
2407                                    ;*POWER UP ROUTINE
2408                                    ;;**********************************************************************
2409  012674  012737  012762  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;SET FOR FAST DOWN
2410  012702  016706  000060                  MOV     $SAVR6,SP       ;GET SP
2411  012706  012677  166126                  MOV     (SP)+,@SWR      ;POP STACK INTO @SWR
2412  012712  012605                          MOV     (SP)+,R5
2413  012714  012604                          MOV     (SP)+,R4        ;POP STACK INTO R4
2414  012716  012603                          MOV     (SP)+,R3        ;POP STACK INTO R3
2415  012720  012602                          MOV     (SP)+,R2        ;POP STACK INTO R2
2416  012722  012601                          MOV     (SP)+,R1        ;POP STACK INTO R1
2417  012724  012600                          MOV     (SP)+,R0        ;POP STACK INTO R0
2418  012726  012737  012622  000024          MOV     #$PWRDN,@#PWRVEC ;SET UP THE POWER DOWN VECTOR
2419  012734  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;PRIO:7
2420  012742  005067  000020                  CLR     $SAVR6          ;WAIT LOOP FOR THE TTY
2421  012746  005267  000014          1$:     INC     $SAVR6          ;WAIT FOR THE INC
2422  012752  001375                          BNE     1$              ;OF WORD
2423  012754  104401                          TYPE                    ;REPORT THE POWER FAILURE
2424  012756  012770                  $PWRMG: .WORD   $POWER          ;POWER FAIL MESSAGE POINTER
2425  012760  000002                          RTI
2426  012762  000000                  $ILLUP: HALT                    ;THE POWER UP SEQUENCE WAS STARTED
2427  012764  000776                          BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE
2428  012766  000000                  $SAVR6: 0                       ;PUT THE SP HERE
2429  012770  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
2430  012776  000122
2431
```

```
2432
2433                                    .SBTTL  SCOPE HANDLER ROUTINE
2434
2435                                    ;;**********************************************************************
2436                                    ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
2437                                    ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
2438                                    ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
2439                                    ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
2440                                    ;*SW14=1         LOOP ON TEST
2441                                    ;*SW09=1         LOOP ON ERROR
2442                                    ;*CALL
2443                                    ;*      SCOPE           ;;SCOPE=IOT
2444
2445  013000                            $SCOPE:
2446  013000  104406                           CKSWR                           ;;TEST  FOR CHANGE IN SOFT-SWR
2447  013002  032777  040000  166030    1$:     BIT     #BIT14,@SWR             ;;LOOP ON PRESENT TEST?
2448  013010  001052                           BNE     $OVER                   ;;YES IF SW14=1
2449                                    ;#####START OF CODE FOR THE XOR TESTER#####
2450  013012  000416                    $XTSTR: BR      6$                      ;;IF RUNNING ON THE "XOR" TESTER CHANGE
2451                                                                            ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
2452  013014  013746  000004                   MOV     @#ERRVEC,-(SP)          ;;SAVE THE CONTENTS OF THE ERROR VECTOR
2453  013020  012737  013040  000004           MOV     #5$,@#ERRVEC            ;;SET FOR TIMEOUT
2454  013026  005737  177060                   TST     @#177060                ;;TIME OUT ON XOR?
2455  013032  012637  000004                   MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
2456  013036  000421                           BR      $SVLAD                  ;;GO TO THE NEXT TEST
2457  013040  022626                    5$:     CMP     (SP)+,(SP)+             ;;CLEAR THE STACK AFTER A TIME OUT
2458  013042  012637  000004                   MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
2459  013046  000407                           BR      7$                      ;;LOOP ON THE PRESENT TEST
2460  013050                            6$:;#####END OF CODE FOR THE XOR TESTER#####
2461  013050  105767  165727            2$:     TSTB    $ERFLG                  ;;HAS AN ERROR OCCURRED?
2462  013054  001412                           BEQ     $SVLAD                  ;;BR IF NO
2463  013056  032777  001000  165754           BIT     #BIT09,@SWR             ;;LOOP ON ERROR?
2464  013064  001404                           BEQ     4$                      ;;BR IF NO
2465  013066  016767  165716  165712    7$:     MOV     $LPERR,$LPADR           ;;SET LOOP ADDRESS TO LAST SCOPE
2466  013074  000420                           BR      $OVER
2467  013076  105067  165701            4$:     CLRB    $ERFLG                  ;;ZERO THE ERROR FLAG
2468  013102  105267  165674            $SVLAD: INCB    $TSTNM                  ;;COUNT TEST NUMBERS
2469  013106  116767  165670  165756           MOVB    $TSTNM,$TESTN           ;;SET TEST NUMBER IN APT MAILBOX
2470  013114  011667  165666                   MOV     (SP),$LPADR             ;;SAVE SCOPE LOOP ADDRESS
2471  013120  011667  165664                   MOV     (SP),$LPERR             ;;SAVE ERROR LOOP ADDRESS
2472  013124  005067  165730                   CLR     $ESCAPE                 ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
2473  013130  112767  000001  165657           MOVB    #1,$ERMAX               ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
2474  013136  016777  165640  165676    $OVER:  MOV     $TSTNM,@DISPLAY         ;;DISPLAY TEST NUMBER
2475  013144  016716  165636                   MOV     $LPADR,(SP)             ;;FUDGE RETURN ADDRESS
2476  013150  000002                           RTI                             ;;FIXES PS
2477
```

```
2478
2479                                    ;;*****************************************************************
2480                                    .SBTTL  APT COMMUNICATIONS ROUTINE
2481                                    ;;*****************************************************************
2482
2483  013152  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG        ;TO REPORT FATAL ERROR
2484  013160  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG        ;TO TYPE A MESSAGE
2485  013166  000403                   BR      $ATYC
2486  013170  112767  000001  000220  $ATY4:  MOVB    #1,$FFLG        ;TO ONLY REPORT FATAL ERROR
2487  013176                           $ATYC:
2488  013176  010046                   MOV     R0,-(SP)        ;PUSH R0 ON STACK
2489  013200  010146                   MOV     R1,-(SP)        ;PUSH R1 ON STACK
2490  013202  105767  000206          TSTB    $MFLG           ;SHOULD TYPE A MESSAGE?
2491  013206  001450                   BEQ     5$              ;IF NOT: BR
2492  013210  122767  000001  165670  CMPB    #APTENV,$ENV    ;OPERATING UNDER APT?
2493  013216  001031                   BNE     3$              ;IF NOT: BR
2494  013220  132767  000100  165661  BITB    #APTSPOOL,$ENVM ;SHOULD SPOOL MESSAGE?
2495  013226  001425                   BEQ     3$              ;IF NOT: BR
2496  013230  017600  000004          MOV     @4(SP),R0       ;GET MESSAGE ADDRESS
2497  013234  062766  000002  000004  ADD     #2,4(SP)        ;BUMP RETURN ADDRESS
2498  013242  005767  165620          1$:     TST     $MSGTYPE                ;SEE IF DONE W/ LAST XMISSION?
2499  013246  001375                   BNE     1$              ;IF NOT: WAIT
2500  013250  010067  165626          MOV     R0,$MSGAD       ;PUT ADDRESS IN MAILBOX
2501  013254  105720                   2$:     TSTB    (R0)+           ;FIND END OF MESSAGE
2502  013256  001376                   BNE     2$
2503  013260  166700  165616          SUB     $MSGAD,R0       ;SUB START OF MESSAGE
2504  013264  006200                   ASR     R0              ;GET MESSAGE LENGTH IN WORDS
2505  013266  010067  165612          MOV     R0,$MSGLGT      ;PUT LENGTH IN MAILBOX
2506  013272  012767  000004  165566  MOV     #4,$MSGTYPE     ;TELL APT TO TAKE MESSAGE
2507  013300  000413                   BR      5$
2508  013302  017667  000004  000016  3$:     MOV     @4(SP),4$       ;PUT MSG ADDR IN JSR LINKAGE
2509  013310  062766  000002  000004  ADD     #2,4(SP)        ;BUMP RETURN ADDRESS
2510  013316  016746  164454          MOV     177776,-(SP)    ;PUSH 177776 ON STACK
2511  013322  004767  000072          JSR     PC,$TYPE        ;CALL TYPE MACRO
2512  013326  000000                   4$:     .WORD   0
2513  013330                           5$:
2514  013330  105767  000062          10$:    TSTB    $FFLG           ;SHOULD REPORT FATAL ERROR?
2515  013334  001413                   BEQ     12$             ;IF NOT: BR
2516  013336  005767  165544          TST     $ENV            ;RUNNING UNDER APT?
2517  013342  001410                   BEQ     12$             ;IF NOT: BR
2518  013344  005767  165516          11$:    TST     $MSGTYPE        ;FINISHED LAST MESSAGE?
2519  013350  001375                   BNE     11$             ;IF NOT: WAIT
2520  013352  017667  000004  165510  MOV     @4(SP),$FATAL   ;GET ERROR #
2521  013360  005267  165502          INC     $MSGTYPE        ;TELL APT TO TAKE ERROR
2522  013364  062766  000002  000004  12$:    ADD     #2,4(SP)        ;BUMP RETURN ADDRESS
2523  013372  105067  000020          CLRB    $FFLG           ;CLEAR FATAL FLAG
2524  013376  105067  000013          CLRB    $LFLG           ;CLEAR LOG FLAG
2525  013402  105067  000006          CLRB    $MFLG           ;CLEAR MESSAGE FLAG
2526  013406  012601                   MOV     (SP)+,R1        ;POP STACK INTO R1
2527  013410  012600                   MOV     (SP)+,R0        ;POP STACK INTO R1
2528  013412  000207                   RTS     PC              ;RETURN
2529  013414     000                   $MFLG:  .BYTE   0
2530  013415     000                   $LFLG:  .BYTE   0               ;LOG FLAG
2531  013416     000                   $FFLG:  .BYTE   0               ;FATAL FLAG
2532
2533  013420                           .EVEN
```

```
                000200                  APTSIZE=200
                000001                  APIENV=001
                000100                  APISPOOL=100
                000040                  APTCSUP=040
```

```
                                .SBTTL  TYPE ROUTINE

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

013420  105767  165433          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
013424  100002                          BPL     1$              ;;BR IF YES
013426  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
013430  000430                          BR      3$              ;;LEAVE
013432  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
013434  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
013440  122767  000001  165440          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
013446  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
013450  132767  000100  165431          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
013456  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
013460  010067  000004                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
013464  004767  177470                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
013470  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
013472  132767  000040  165407  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
013500  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
013502  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
013504  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
013506  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
013510  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
013512  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
013516  000002                          RTI                     ;;RETURN
013520  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
013524  001430                          BEQ     8$
013526  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
013532  001006                          BNE     5$
013534  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
013536  104401                          TYPE                    ;;TYPE A CR AND LF
013540  001063                          $CRLF
013542  105067  000130                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
013546  000755                          BR      2$              ;;GET NEXT CHARACTER
013550  004767  000056          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
013554  126726  165276          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
013560  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
013562  016746  165266                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
013566  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
013572  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
013574  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
```

```
2595  013600  105367  000072            DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
2596  013604  000770                    BR      7$              ;;LOOP
2597
2598                                    ;HORIZONTAL TAB PROCESSOR
2599
2600  013606  112716  000040            8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
2601  013612  004767  000014            9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
2602  013616  132767  000007  000052            BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
2603  013624  001372                            BNE     9$              ;;TAB STOP
2604  013626  005726                            TST     (SP)+           ;;POP SPACE OFF STACK
2605  013630  000724                            BR      2$              ;;GET NEXT CHARACTER
2606  013632  105777  165212            $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
2607  013636  100375                            BPL     $TYPEC
2608  013640  116677  000002  165204            MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
2609  013646  122766  000015  000002            CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
2610  013654  001003                            BNE     1$              ;;BRANCH IF NO
2611  013656  105067  000014                    CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
2612  013662  000406                            BR      $TYPEX          ;;EXIT
2613  013664  122766  000012  000002  1$:       CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
2614  013672  001402                            BEQ     $TYPEX          ;;BRANCH IF YES
2615  013674  105227                            INCB    (PC)+           ;;COUNT THE CHARACTER
2616  013676  000000            $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
2617  013700  000207            $TYPEX: RTS     PC
2618
2619                                    .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
2620
2621                                    ;;**********************************************************************
2622                                    ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
2623                                    ;*OCTAL (ASCII) NUMBER AND TYPE IT.
2624                                    ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
2625                                    ;*CALL:
2626                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
2627                                    ;*      TYPOS                   ;;CALL FOR TYPEOUT
2628                                    ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
2629                                    ;*      .BYTE   M               ;;M=1 OR 0
2630                                    ;*                                      ;;1=TYPE LEADING ZEROS
2631                                    ;*                                      ;;0=SUPPRESS LEADING ZEROS
2632                                    ;*
2633                                    ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
2634                                    ;*$TYPOS OR $TYPOC
2635                                    ;*CALL:
2636                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
2637                                    ;*      TYPON                   ;;CALL FOR TYPEOUT
2638                                    ;*
2639                                    ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
2640                                    ;*CALL:
2641                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
2642                                    ;*      TYPOC                   ;;CALL FOR TYPEOUT
2643
2644  013702  017646  000000            $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
2645  013706  116667  000001  000211            MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
2646  013714  112667  000207                    MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
2647  013720  062716  000002                    ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
2648  013724  000406                            BR      $TYPON
2649  013726  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
2650  013734  112767  000006  000165            MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
```

```
 2651  013742  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 2652  013750  010346                          MOV     R3,-(SP)        ;;SAVE R3
 2653  013752  010446                          MOV     R4,-(SP)        ;;SAVE R4
 2654  013754  010546                          MOV     R5,-(SP)        ;;SAVE R5
 2655  013756  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 2656  013762  005404                          NEG     R4
 2657  013764  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 2658  013770  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 2659  013774  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 2660  014000  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 2661  014004  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
 2662  014006  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 2663  014010  000404                          BR      3$              ;;GO DO MSB
 2664  014012  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
 2665  014014  006105                          ROL     R5
 2666  014016  006105                          ROL     R5
 2667  014020  010503                          MOV     R5,R3
 2668  014022  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 2669  014024  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
 2670  014030  100016                          BPL     7$              ;;BR IF NO
 2671  014032  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
 2672  014036  001002                          BNE     4$              ;;TEST FOR 0
 2673  014040  005704                          TST     R4              ;;SUPPRESS THIS 0?
 2674  014042  001403                          BEQ     5$              ;;BR IF YES
 2675  014044  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 2676  014046  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 2677  014052  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 2678  014056  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
 2679  014062  104401  014122                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
 2680  014066  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
 2681  014072  003347                          BGT     2$              ;;BR IF MORE TO DO
 2682  014074  002402                          BLT     6$              ;;BR IF DONE
 2683  014076  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 2684  014100  000744                          BR      2$              ;;GO DO THE LAST DIGIT
 2685  014102  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
 2686  014104  012604                          MOV     (SP)+,R4        ;;RESTORE R4
 2687  014106  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 2688  014110  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 2689  014116  012616                          MOV     (SP)+,(SP)
 2690  014120  000002                          RTI                     ;;RETURN
 2691  014122     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 2692  014123     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 2693  014124     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 2694  014125     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 2695  014126  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
 2696
```

```
                                .SBTTL  TTY INPUT ROUTINE

                                ;;**********************************************************************
                                .ENABL  LSB

                                ;;**********************************************************************
                                ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                ;*WHEN OPERATING IN TTY FLAG MODE.
014130  022767  000176  164702  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
014136  001074                         BNE     15$             ;;BRANCH IF NO
014140  105777  164700                 TSTB    @$TKS           ;;CHAR THERE?
014144  100071                         BPL     15$             ;;IF NO, DON'T WAIT AROUND
014146  117746  164674                 MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
014152  042716  177600                 BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
014156  022726  000007                 CMP     #7,(SP)+        ;;IS IT A CONTROL G?
014162  001062                         BNE     15$             ;;NO, RETURN TO USER
014164  126727  164644  000001         CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
014172  001456                         BEQ     15$             ;;BRANCH IF YES

014174  104401  014655                 TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
014200  104401  014662         $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
014204  016746  163766                 MOV     $WREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
014210  104402                         TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
014212  104401  014673                 TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
014216  005046                 19$:    CLR     -(SP)           ;;CLEAR COUNTER
014220  005046                         CLR     -(SP)           ;;THE NEW SWR
014222  105777  164616         7$:     TSTB    @$TKS           ;;CHAR THERE?
014226  100375                         BPL     7$              ;;IF NOT TRY AGAIN

014230  117746  164612                 MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
014234  042716  177600                 BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII



014240  021627  000025         9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
014244  001005                         BNE     10$             ;;BRANCH IF NOT
014246  104401  014650                 TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
014252  062706  000006         20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
014256  000757                         BR      19$             ;;LET'S TRY IT AGAIN


014260  021627  000015         10$:    CMP     (SP),#15        ;;IS IT A <CR>?
014264  001022                         BNE     16$             ;;BRANCH IF NO
014266  005766  000004                 TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
014272  001403                         BEQ     11$             ;;BRANCH IF YES
014274  016677  000002  164536         MOV     2(SP),@SWR      ;;SAVE NEW SWR
014302  062706  000006         11$:    ADD     #6,SP           ;;CLEAR UP STACK
014306  104401  001063         14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
014312  126727  164517  000001         CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
014320  001003                         BNE     15$             ;;BRANCH IF NOT
014322  012777  000100  164514         MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
014330  000002                 15$:    RTI                     ;;RETURN
014332  004767  177274         16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
```

```
2753  014336  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
2754  014342  002420                          BLT     18$             ;;BRANCH IF YES
2755  014344  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
2756  014350  003015                          BGT     18$             ;;BRANCH IF YES
2757  014352  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
2758  014356  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
2759  014362  001403                          BEQ     17$             ;;BRANCH IF YES
2760  014364  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
2761  014366  006316                          ASL     (SP)            ;;    CHAR OVER TO MAKE
2762  014370  006316                          ASL     (SP)            ;;    ROOM FOR NEW ONE.
2763  014372  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
2764  014376  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
2765  014402  000707                          BR      7$              ;;GET THE NEXT ONE
2766  014404  104401  001062          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
2767  014410  000720                          BR      20$             ;;SIMULATE CONTROL-U
2768                                .DSABL  LSB
2769
2770
2771                                ;;*****************************************************************
2772                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
2773                                ;*CALL:
2774                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
2775                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
2776                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
2777                                ;
2778
2779  014412  011646                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
2780  014414  016666  000004  000002        MOV     4(SP),2(SP)     ;;SAVE THE PS
2781  014422  105777  164416          1$:   TSTB    @$TKS           ;;WAIT FOR
2782  014426  100375                        BPL     1$              ;;A CHARACTER
2783  014430  117766  164412  000004        MOVB    @$TKB,4(SP)     ;;READ THE TTY
2784  014436  042766  177600  000004        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
2785  014444  026627  000004  000023        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
2786  014452  001013                        BNE     3$              ;;BRANCH IF NO
2787  014454  105777  164364          2$:   TSTB    @$TKS           ;;WAIT FOR A CHARACTER
2788  014460  100375                        BPL     2$              ;;LOOP UNTIL ITS THERE
2789  014462  117746  164360                MOVB    @$TKB,-(SP)     ;;GET CHARACTER
2790  014466  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
2791  014472  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
2792  014476  001366                        BNE     2$              ;;IF NOT DISCARD IT
2793  014500  000750                        BR      1$              ;;YES, RESUME
2794  014502  026627  000004  000140  3$:   CMP     4(SP),#140      ;;IS IT UPPER CASE?
2795  014510  002407                        BLT     4$              ;;BRANCH IF YES
2796  014512  026627  000004  000175        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
2797  014520  003003                        BGT     4$              ;;BRANCH IF YES
2798  014522  042766  000040  000004        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
2799  014530  000002                  4$:   RTI                     ;;GO BACK TO USER
2800                                ;;*****************************************************************
2801                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
2802                                ;*CALL:
2803                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
2804                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
2805                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
2806
2807  014532  010346                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
2808  014534  012703  014640          1$:   MOV     #$TTYIN,R3      ;;GET ADDRESS
```

```
 2809  014540  022703  014650          2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
 2810  014544  101405                          BLOS    4$              ;;BR IF YES
 2811  014546  104407                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
 2812  014550  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
 2813  014552  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
 2814  014556  001003                          BNE     3$              ;;SKIP IF NOT
 2815  014560  104401  001062          4$:     TYPE    ,$QUES          ;;TYPE A '?'
 2816  014564  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
 2817  014566  111367  000044          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
 2818  014572  104401  014636                  TYPE    ,9$
 2819  014576  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 2820  014602  001356                          BNE     2$              ;;LOOP IF NOT RETURN
 2821  014604  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 2822  014610  104401  001064                  TYPE    ,$LF            ;;TYPE A LINE FEED
 2823  014614  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 2824  014616  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 2825  014620  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 2826  014626  012766  014640  000004          MOV     #$TTYIN,4(SP)
 2827  014634  000002                          RTI                     ;;RETURN
 2828  014636     000          9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
 2829  014637     000                  .BYTE   0                       ;;TERMINATOR
 2830  014640  000010          $TTYIN: .BLKB   8.                      ;;RESERVE 8 BYTES FOR TTY INPUT
 2831  014650  052536  005015     000  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
 2832  014655     136  006507  000012  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
 2833  014662  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
 2834  014670  020075     000
 2835  014673     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
 2836  014700  036440  000040
 2837
```

```
2838
2839
2840                                    .SBTTL  TRAP DECODER
2841
2842                                    ;;**********************************************************************
2843                                    ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
2844                                    ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
2845                                    ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
2846                                    ;*GO TO THAT ROUTINE.
2847
2848  014704  010046                    $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
2849  014706  016600  000002                   MOV     2(SP),R0        ;;GET TRAP ADDRESS
2850  014712  005740                           TST     -(R0)           ;;BACKUP BY 2
2851  014714  111000                           MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
2852  014716  006300                           ASL     R0              ;;POSITION FOR INDEXING
2853  014720  016000  014740                   MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
2854  014724  000200                           RTS     R0              ;;GO TO ROUTINE
2855
2856
2857                                    ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
2858
2859  014726  011646                    $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
2860  014730  016666  000004  000002            MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
2861  014736  000002                           RTI                     ;;RESTORE THE PSW
2862
2863                                    .SBTTL  TRAP TABLE
2864
2865                                    ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
2866                                    ;*BY THE "TRAP" INSTRUCTION.
2867
2868                                    ;       ROUTINE
2869                                    ;       -------
2870  014740  014726                    $TRPAD: .WORD   $TRAP2
2871  014742  013420                            $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
2872  014744  013726                            $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
2873  014746  013702                            $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
2874  014750  013742                            $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
2875
2876  014752  014200                            $GTSWR  ;;CALL=GTSWR    TRAP+5(104405)  GET SOFT-SWR SETTING
2877
2878  014754  014130                            $CKSWR  ;;CALL=CKSWR    TRAP+6(104406)  TEST FOR CHANGE IN SOFT-SWR
2879  014756  014412                            $RDCHR  ;;CALL=RDCHR    TRAP+7(104407)  TTY TYPEIN CHARACTER ROUTINE
2880  014760  014532                            $RDLIN  ;;CALL=RDLIN    TRAP+10(104410) TTY TYPEIN STRING ROUTINE
2881
```

```
                                        .SBTTL  ECHO TEST
                                ;;**********************************************************************
                                        ;*THIS ROUTINE WILL ECHO ANY CHARACTER TYPED
                                        ;*AT THE CONSOLE
                                        ;*THE TEST IS HALTED BY TYPING A CONTROL-C
                                        ;*TEST CAN BE RESTARTED AFTER HALTING BY JUST CONTINUING
                                ;;**********************************************************************
014762                          ECHO:
014762  000005                        RESET                   ;CLEAR EVERYTHING
014764  112777  000052  165402        MOVB    #'*,@CTBUF      ;TRANSMIT PROMPT "*"
014772  105777  165370        2$:     TSTB    @CRCSR          ;WAIT FOR INPUT
014776  100375                        BPL     2$
015000  117777  165364  165366        MOVB    @CRBUF,@CTBUF   ;ECHO INPUT
015006  017700  165356                MOV     @CRBUF,R0       ;STORE INPUT
015012  100023                        BPL     5$              ;BR IF "ERROR"NOT SET
015014  052701  010000                BIS     #BIT12,R1       ;SET PARITY ERROR TEST MASK
015020  030100                        BIT     R1,R0           ;CHECK  FOR PARITY ERROR FLAG
015022  001403                        BEQ     3$              ;BR IF NOT SET
015024  004767  000056                JSR     PC,MSG          ;REPORT PARITY ERROR
015030  015134                        MPAR
015032  006301                3$:     ASL     R1              ;SHIFT MASK TO TEST "FR" FLAG
015034  030100                        BIT     R1,R0           ;TEST FOR FRAMING ERROR FLAG
015036  001403                        BEQ     4$              ;BR IF NOT SET
015040  004767  000042                JSR     PC,MSG          ;REPSORT FRAMING ERROR
015044  015145                        MFR
015046  006301                4$:     ASL     R1              ;SHIFT MASK TO TEST "OR" FLAG
015050  030100                        BIT     R1,R0           ;TEST FOR OVERFLOW ERROR
015052  001403                        BEQ     5$              ;BR IF NOT SET
015054  004767  000026                JSR     PC,MSG          ;REPORT OVERFLOW ERROR
015060  015157                        MOR
015062  042700  000200        5$:     BIC     #BIT7,R0        ;CLEAR ANY PARITY BIT
015066  022700  000003                CMP     #3,R0           ;WAS INPUT CONTROL-C
015072  001337                        BNE     2$              ;BR IF NOT
015074  004767  000006                JSR     PC,MSG          ;REPORT PROGRAM STOP
015100  015172                        MSTOP
015102  000000                        HALT                    ;END OF TEST HALT
015104  000726                        BR      ECHO            ;AFTER END OF TEST HALT
                                                            ;   PRESS CONTINUE TO RESTART ECHO TEST

015106  013600                MSG:    MOV     @(SP)+,R0       ;PICK UP MESSAGE POINTER
015110  062746  000002                ADD     #2,-(SP)        ;ADJUST RETURN PC
015114  105777  165252        WAIT:   TSTB    @CTCSR          ;WAIT FOR XMIT DONE
015120  100375                        BPL     WAIT
015122  112077  165246                MOVB    (R0)+,@CTBUF    ;SEND CHARACTER
015126  105710                        TSTB    (R0)            ;IS THIS END OF MESSAGE?
015130  001371                        BNE     WAIT            ;BR IF NOT
015132  000207                        RTS     PC              ;RETURN

015134  005015  040520  044522  MPAR:   .ASCIZ  <CR><LF>/PARITY/
015142  054524     000
015145     015  043012  040522  MFR:    .ASCIZ  <CR><LF>/FRAMING/
015152  044515  043516     000
015157     015  047412  042526  MOR:    .ASCIZ  <CR><LF>/OVERFLOW/
015164  043122  047514  000127
015172  005015  052123  050117  MSTOP:  .ASCIZ  <CR><LF>/STOP/
015200     000
```

```
                                .EVEN

                        .SBTTL  TERMINAL OUTPUT TEST

                        ;;**********************************************************************
                                ;*THIS ROUTINE WILL OUTPUT ALL WRITABLE CHARACTERS FOR THE
                                ;*THE OCTAL CODE 040 --> 377
                                ;*32 CHARACTERS ARE PRINTED ON EACH LINE
                                ;*THE PATTERN IS REPEATED EVERY THREE LINES
                                ;*
                        ;;**********************************************************************

015202  012701  000040  OUTTST: MOV     #40,R1          ;LOAD FIRST WRITABLE CHARACTER
015206  012700  000040  1$:     MOV     #40,R0          ;LOAD CHAR COUNT PER LINE
015212  105777  165154  2$:     TSTB    @XCTCSR         ;WAIT FOR DONE
015216  100375                  BPL     2$
015220  010177  165150          MOV     R1,@XCTBUF      ;TRANSMIT A CHARACTER
015224  105201                  INCB    R1              ;INCREMENT CHARACTER CODE
015226  005300                  DEC     R0              ;DECREMENT CHAR COUNT
015230  001370                  BNE     2$              ;BR IF LINE NOT COMPLETE
015232  004767  177650          JSR     PC,MSG          ;SSUE CR,LINE FEED

015236  001063                  $CRLF
015240  105777  165122          TSTB    @RCRCSR         ;ANY CHARACTER RECEIVED?
015244  100404                  BMI     3$              ;BR IF YES
015246  032701  000200          BIT     #BIT7,R1        ;FINISHED ONE PASS OF WRITABLE CHARACTERS?
015252  001353                  BNE     OUTTST          ;BR IF YES
015254  000754                  BR      1$              ;IF NOT WRITE NEXT LINE

015256  005077  165106  3$:     CLR     @RCRBUF         ;CLEAR RECEIVER
015262  000000                  HALT                    ;STOP TEST
015264  000746                  BR      OUTTST          ;RESTART TEST IF CONTINUED
```

```
                                                
                                                
015266  040503  020116  047516  EM1:    .ASCIZ  /CAN NOT ACCESS TCSR/
015274  020124  041501  042503
015302  051523  052040  051503
015310  000122
015312  040503  020116  047516  EM2:    .ASCIZ  /CAN NOT ACCESS TBUF/
015320  020124  041501  042503
015326  051523  052040  052502
015334  000106
015336  041524  051123  042040  EM3:    .ASCIZ  /TCSR DONE NOT CLEARED WITH TBUF FULL/
015344  047117  020105  047516
015352  020124  046103  040505
015360  042522  020104  044527
015366  044124  052040  052502
015374  020106  052506  046114
015402      000
015403      124  051503  020122  EM4:    .ASCIZ  /TCSR DONE NOT SET/
015410  047504  042516  047040
015416  052117  051440  052105
015424      000
015425      124  051503  020122  EM5:    .ASCIZ  /TCSR DONE NOT SET WITH RESET/
015432  047504  042516  047040
015440  052117  051440  052105
015446  053440  052111  020110
015454  042522  042523  000124
015462  040503  020116  047516  EM6:    .ASCIZ  /CAN NOT ACCESS RCSR/
015470  020124  041501  042503
015476  051523  051040  051503
015504  000122
015506  040503  020116  047516  EM7:    .ASCIZ  /CAN NOT ACCESS RBUF/
015514  020124  041501  042503
015522  051523  051040  052502
015530  000106
015532  040503  020116  047516  EM10:   .ASCIZ  /CAN NOT ACCESS LKS/
015540  020124  041501  042503
015546  051523  046040  051513
015554      000
015555      102  052111  020060  EM11:   .ASCIZ  /BITO OF TCSR NOT CLEAR AFTER RESET/
015562  043117  052040  051503
015570  020122  047516  020124
015576  046103  040505  020122
015604  043101  042524  020122
015612  042522  042523  000124
015620  040503  020116  047516  EM12:   .ASCIZ  /CAN NOT SET BITO OF TCSR/
015626  020124  042523  020124
015634  044502  030124  047440
015642  020106  041524  051123
015650      000
015651      103  047101  047040  EM13:   .ASCIZ  /CAN NOT CLEAR BITO OF TCSR/
015656  052117  041440  042514
015664  051101  041040  052111
015672  020060  043117  052040
015700  051503  000122
015704  042522  042523  020124  EM14:   .ASCIZ  /RESET DID NOT CLEAR BITO OF TCSR/
015712  044504  020104  047516
```

```
 3028  015720  020124  046103  040505
 3029  015726  020122  044502  030124
 3030  015734  047440  020106  041524
 3031  015742  051123     000
 3032  015745     102  052111  020062  EM15:   .ASCIZ  /BIT2 OF TCSR NOT CLEAR AFTER RESET/
 3033  015752  043117  052040  051503
 3034  015760  020122  047516  020124
 3035  015766  046103  040505  020122
 3036  015774  043101  042524  020122
 3037  016002  042522  042523  000124
 3038  016010  040503  020116  047516  EM16:   .ASCIZ  /CAN NOT SET BIT2 OF TCSR/
 3039  016016  020124  042523  020124
 3040  016024  044502  031124  047440
 3041  016032  020106  041524  051123
 3042  016040     000
 3043  016041     103  047101  047040  EM17:   .ASCIZ  /CAN NOT CLEAR BIT2 OF TCSR/
 3044  016046  052117  041440  042514
 3045  016054  051101  041040  052111
 3046  016062  020062  043117  052040
 3047  016070  051503  000122
 3048  016074  042522  042523  020124  EM20:   .ASCIZ  /RESET DID NOT CLEAR BIT2 OF TCSR/
 3049  016102  044504  020104  047516
 3050  016110  020124  046103  040505
 3051  016116  020122  044502  031124
 3052  016124  047440  020106  041524
 3053  016132  051123     000
 3054  016135     102  052111  020066  EM21:   .ASCIZ  ;BIT6 OF TCSR NOT CLEAR AFTER RESET/
 3055  016142  043117  052040  051503
 3056  016150  020122  047516  020124
 3057  016156  046103  040505  020122
 3058  016164  043101  042524  020122
 3059  016172  042522  042523  027524
 3060  016200     000
 3061  016201     130  044515  020124  EM22:   .ASCIZ  /XMIT INTERRUPT AT PRIORITY 7/
 3062  016206  047111  042524  051122
 3063  016214  050125  020124  052101
 3064  016222  050040  044522  051117
 3065  016230  052111  020131  000067
 3066  016236  040503  020116  047516  EM23:   .ASCIZ  /CAN NOT SET BIT6 OF TCSR/
 3067  016244  020124  042523  020124
 3068  016252  044502  033124  047440
 3069  016260  020106  041524  051123
 3070  016266     000
 3071  016267     103  047101  047040  EM24:   .ASCIZ  /CAN NOT CLEAR BIT6 OF TCSR/
 3072  016274  052117  041440  042514
 3073  016302  051101  041040  052111
 3074  016310  020066  043117  052040
 3075  016316  051503  000122
 3076  016322  042522  042523  020124  EM25:   .ASCIZ  /RESET DID NOT CLEAR BIT6 OF TCSR/
 3077  016330  044504  020104  047516
 3078  016336  020124  046103  040505
 3079  016344  020122  044502  033124
 3080  016352  047440  020106  041524
 3081  016360  051123     000
 3082  016363     102  052111  020066  EM26:   .ASCIZ  /BIT6 OF RCSR NOT CLEAR AFTER RESET/
 3083  016370  043117  051040  051503
```

```
 3084  016376  020122  047516  020124
 3085  016404  046103  040505  020122
 3086  016412  043101  042524  020122
 3087  016420  042522  042523  000124
 3088  016426  041522  051126  044440  EM27:   .ASCIZ  /RCVR INTERRUPT WITH PRIORITY 7/
 3089  016434  052116  051105  052522
 3090  016442  052120  053440  052111
 3091  016450  020110  051120  047511
 3092  016456  044522  054524  033440
 3093  016464     000
 3094  016465     103  047101  047040  EM30:   .ASCIZ  /CAN NOT SET BIT6 OF RCSR/
 3095  016472  052117  051440  052105
 3096  016500  041040  052111  020066
 3097  016506  043117  051040  051503
 3098  016514  000122
 3099  016516  040503  020116  047516  EM31:   .ASCIZ  /CAN NOT CLEAR BIT6 OF RCSR/
 3100  016524  020124  046103  040505
 3101  016532  020122  044502  033124
 3102  016540  047440  020106  041522
 3103  016546  051123     000
 3104  016551     103  047101  047040  EM32:   .ASCIZ  /CAN NOT CLEAR BIT6 OF RCSR WITH RESET/
 3105  016556  052117  041440  042514
 3106  016564  051101  041040  052111
 3107  016572  020066  043117  051040
 3108  016600  051503  020122  044527
 3109  016606  044124  051040  051505
 3110  016614  052105     000
 3111  016617     102  052111  020066  EM33:   .ASCIZ  /BIT6 OF LKS NOT CLEAR AFTER RESET/
 3112  016624  043117  046040  051513
 3113  016632  047040  052117  041440
 3114  016640  042514  051101  040440
 3115  016646  052106  051105  051040
 3116  016654  051505  052105     000
 3117  016661     114  051513  044440  EM34:   .ASCIZ  /LKS INTERRUPT WITH PRIORITY 7/
 3118  016666  052116  051105  052522
 3119  016674  052120  053440  052111
 3120  016702  020110  051120  047511
 3121  016710  044522  054524  033440
 3122  016716     000
 3123  016717     103  047101  047040  EM35:   .ASCIZ  /CAN NOT SET BIT6 OF LKS/
 3124  016724  052117  051440  052105
 3125  016732  041040  052111  020066
 3126  016740  043117  046040  051513
 3127  016746     000
 3128  016747     103  047101  047040  EM36:   .ASCIZ  /CAN NOT CLEAR BIT6 OF LKS/
 3129  016754  052117  041440  042514
 3130  016762  051101  041040  052111
 3131  016770  020066  043117  046040
 3132  016776  051513     000
 3133  017001     122  051505  052105  EM37:   .ASCIZ  /RESET DID NOT CLEAR BIT6 OF LKS/
 3134  017006  042040  042111  047040
 3135  017014  052117  041440  042514
 3136  017022  051101  041040  052111
 3137  017030  020066  043117  046040
 3138  017036  051513     000
 3139  017041     104  040525  020114  EM40:   .ASCIZ  /DUAL ADDRESSING ERROR/
```

```
3140  017046  042101  051104  051505
3141  017054  044523  043516  042440
3142  017062  051122  051117     000
3143  017067     102  052111  020066  EM41:   .ASCIZ  /BIT6 OF LKS NOT SET AFTER RESET/
3144  017074  043117  046040  051513
3145  017102  047040  052117  051440
3146  017110  052105  040440  052106
3147  017116  051105  051040  051505
3148  017124  052105     000
3149  017127     103  047101  047040  EM42:   .ASCIZ  /CAN NOT CLEAR BIT7 OF LKS/
3150  017134  052117  041440  042514
3151  017142  051101  041040  052111
3152  017150  020067  043117  046040
3153  017156  051513     000
3154  017161     102  052111  020067  EM43:   .ASCIZ  /BIT7 OF LKS DOES NOT SET/
3155  017166  043117  046040  051513
3156  017174  042040  042517  020123
3157  017202  047516  020124  042523
3158  017210  000124
3159  017212  052122  020103  047111  EM44:   .ASCIZ  /RTC INTERRUPT AT PRIORITY 7/
3160  017220  042524  051122  050125
3161  017226  020124  052101  050040
3162  017234  044522  051117  052111
3163  017242  020131  000067
3164  017246  052122  020103  047111  EM45:   .ASCIZ  /RTC INTERRUPTS WHEN DISABLED/
3165  017254  042524  051122  050125
3166  017262  051524  053440  042510
3167  017270  020116  044504  040523
3168  017276  046102  042105     000
3169  017303                          EM47:
3170  017303     122  041524  044440  EM46:   .ASCIZ  /RTC INTERRUPT DID NOT OCCUR/
3171  017310  052116  051105  052522
3172  017316  052120  042040  042111
3173  017324  047040  052117  047440
3174  017332  041503  051125     000
3175  017337     122  041524  042040  EM50:   .ASCIZ  /RTC DOUBLE INTERRUPT/
3176  017344  052517  046102  020105
3177  017352  047111  042524  051122
3178  017360  050125  000124
3179  017364  042522  042523  020124  EM51:   .ASCIZ  /RESET DID NOT INTERRUPT/
3180  017372  044504  020104  047516
3181  017400  020124  047111  042524
3182  017406  051122  050125  000124
3183  017414  052122  020103  047111  EM52:   .ASCIZ  /RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS/
3184  017422  042524  051122  050125
3185  017430  020124  044504  020104
3186  017436  047516  020124  046103
3187  017444  040505  020122  044527
3188  017452  044124  041040  052111
3189  017460  020067  043117  046040
3190  017466  051513     000
3191  017471     103  047514  045503  EM53:   .ASCIZ  /CLOCK REPEATABILITY/
3192  017476  051040  050105  040505
3193  017504  040524  044502  044514
3194  017512  054524     000
3195  017515     130  044515  020124  EM54:   .ASCIZ  /XMIT INTERRUPTS WHEN DISABLED/
```

```
3196  017522  047111  042524  051122
3197  017530  050125  051524  053440
3198  017536  042510  020116  044504
3199  017544  040523  046102  042105
3200  017552     000
3201  017553     130  044515  020124  EM56:   .ASCIZ  /XMIT INTERRUPTS AT PRIORITY 7/
3202  017560  047111  042524  051122
3203  017566  050125  051524  040440
3204  017574  020124  051120  047511
3205  017602  044522  054524  033440
3206  017610     000
3207  017611     130  044515  020124  EM57:   .ASCIZ  /XMIT INTERRUPTS WITH ENABLE CLEAR/
3208  017616  047111  042524  051122
3209  017624  050125  051524  053440
3210  017632  052111  020110  047105
3211  017640  041101  042514  041440
3212  017646  042514  051101     000
3213  017653                  EM55:
3214  017653     130  044515  020124  EM60:   .ASCIZ  /XMIT DID NOT INTERRUPT/
3215  017660  044504  020104  047516
3216  017666  020124  047111  042524
3217  017674  051122  050125  000124
3218  017702  046530  052111  051040  EM61:   .ASCIZ  /XMIT RE-INTERRUPTED/
3219  017710  026505  047111  042524
3220  017716  051122  050125  042524
3221  017724  000104
3222  017726  047514  042101  047111  EM62:   .ASCIZ  /LOADING TBUF DID NOT CLEAR INTERRUPT/
3223  017734  020107  041124  043125
3224  017742  042040  042111  047040
3225  017750  052117  041440  042514
3226  017756  051101  044440  052116
3227  017764  051105  052522  052120
3228  017772     000
3229  017773     122  053103  020122  EM63:   .ASCIZ  /RCVR ACTIVE NOT SET/
3230  020000  041501  044524  042526
3231  020006  047040  052117  051440
3232  020014  052105     000
3233  020017     122  053103  020122  EM64:   .ASCIZ  /RCVR DONE NEVER SET/
3234  020024  047504  042516  047040
3235  020032  053105  051105  051440
3236  020040  052105     000
3237  020043     122  053103  020122  EM65:   .ASCIZ  /RCVR ACTIVE NOT CLEARED WITH DONE SET/
3238  020050  041501  044524  042526
3239  020056  047040  052117  041440
3240  020064  042514  051101  042105
3241  020072  053440  052111  020110
3242  020100  047504  042516  051440
3243  020106  052105     000
3244  020111     122  051505  052105  EM66:   .ASCIZ  /RESET DID NOT CLEAR RCVR DONE/
3245  020116  042040  042111  047040
3246  020124  052117  041440  042514
3247  020132  051101  051040  053103
3248  020140  020122  047504  042516
3249  020146     000
3250  020147     122  051104  042440  EM67:   .ASCIZ  /RDR ENABLE DID NOT CLEAR RCVR DONE/
3251  020154  040516  046102  020105
```

```
3252  020162  044504  020104  047516
3253  020170  020124  046103  040505
3254  020176  020122  041522  051126
3255  020204  042040  047117  000105
3256  020212  042522  042101  047111  EM70:   .ASCIZ  /READING RBUF DID NOT CLEAR RCVR DONE/
3257  020220  020107  041122  043125
3258  020226  042040  042111  047040
3259  020234  052117  041440  042514
3260  020242  051101  051040  053103
3261  020250  020122  047504  042516
3262  020256     000
3263  020257     122  053103  020122  EM71:   .ASCIZ  /RCVR INTERRUPTS WITH ENABLE CLEAR/
3264  020264  047111  042524  051122
3265  020272  050125  051524  053440
3266  020300  052111  020110  047105
3267  020306  041101  042514  041440
3268  020314  042514  051101     000
3269  020321     122  053103  020122  EM73:   .ASCIZ  /RCVR INTERRUPTS AT PRIORITY 7/
3270  020326  047111  042524  051122
3271  020334  050125  051524  040440
3272  020342  020124  051120  047511
3273  020350  044522  054524  033440
3274  020356     000
3275  020357     122  053103  020122  EM74:   .ASCIZ  /RCVR INT RQST PASSED WITH ENABLE CLEAR/
3276  020364  047111  020124  050522
3277  020372  052123  050040  051501
3278  020400  042523  020104  044527
3279  020406  044124  042440  040516
3280  020414  046102  020105  046103
3281  020422  040505  000122
3282  020426                          EM72:
3283  020426  041522  051126  042040  EM75:   .ASCIZ  /RCVR DID NOT INTERRUPT/
3284  020434  042111  047040  052117
3285  020442  044440  052116  051105
3286  020450  052522  052120     000
3287  020455     122  053103  020122  EM76:   .ASCIZ  /RCVR RE-INTERRUPTED/
3288  020462  042522  044455  052116
3289  020470  051105  052522  052120
3290  020476  042105     000
3291  020501     122  040505  044504  EM77:   .ASCIZ  /READING RBUF DID NOT CLEAR INTERRUPT/
3292  020506  043516  051040  052502
3293  020514  020106  044504  020104
3294  020522  047516  020124  046103
3295  020530  040505  020122  047111
3296  020536  042524  051122  050125
3297  020544  000124
3298  020546  042522  042523  020124  EM100:  .ASCIZ  /RESET DID NOT CLEAR RCVR INTERRUPT/
3299  020554  044504  020104  047516
3300  020562  020124  046103  040505
3301  020570  020122  041522  051126
3302  020576  044440  052116  051105
3303  020604  052522  052120     000
3304  020611     047  051117  020047  EM101:  .ASCIZ  /'OR' FLAG DID NOT SET/
3305  020616  046106  043501  042040
3306  020624  042111  047040  052117
3307  020632  051440  052105     000
```

```
3308  020637     047  051105  047522  EM102:  .ASCIZ  /'ERROR' NOT SET WITH 'OR' FLAG/
3309  020644  023522  047040  052117
3310  020652  051440  052105  053440
3311  020660  052111  020110  047447
3312  020666  023522  043040  040514
3313  020674  000107
3314  020676  051102  040505  020113  EM103:  .ASCIZ  /BREAK DID NOT XMIT ALL ZEROES/
3315  020704  044504  020104  047516
3316  020712  020124  046530  052111
3317  020720  040440  046114  055040
3318  020726  051105  042517  000123
3319  020734  051102  040505  020113  EM104:  .ASCIZ  /BREAK DID NOT SET 'FR' ERROR/
3320  020742  044504  020104  047516
3321  020750  020124  042523  020124
3322  020756  043047  023522  042440
3323  020764  051122  051117     000
3324  020771     104  052101  020101  EM105:  .ASCIZ  /DATA COMPARE ERROR/
3325  020776  047503  050115  051101
3326  021004  020105  051105  047522
3327  021012  000122
3328  021014  040504  040524  041440  EM106:  .ASCIZ  /DATA COMPARE ERROR WITH CABLE/
3329  021022  046517  040520  042522
3330  021030  042440  051122  051117
3331  021036  053440  052111  020110
3332  021044  040503  046102  000105
3333  021052  044524  042515  052517  EM107:  .ASCIZ  /TIMEOUT IN EXERCISER TEST/
3334  021060  020124  047111  042440
3335  021066  042530  041522  051511
3336  021074  051105  052040  051505
3337  021102  000124
3338  021104  047111  047503  051122  EM110:  .ASCIZ  /INCORRECT RECEIVE COUNT/
3339  021112  041505  020124  042522
3340  021120  042503  053111  020105
3341  021126  047503  047125  000124
3342  021134  040504  040524  041440  EM111:  .ASCIZ  /DATA COMPARE ERROR IN EXERCISER/
3343  021142  046517  040520  042522
3344  021150  042440  051122  051117
3345  021156  044440  020116  054105
3346  021164  051105  044503  042523
3347  021172  000122
3348  021174  051124  050101  041440  EM112:  .ASCIZ  /TRAP CATCHER/
3349  021202  052101  044103  051105
3350  021210     000
3351  021211     116  020117  046103  EM113:  .ASCIZ  /NO CLK INTERRUPT IN EXERCISER/
3352  021216  020113  047111  042524
3353  021224  051122  050125  020124
3354  021232  047111  042440  042530
3355  021240  041522  051511  051105
3356  021246     000
3357  021247     042  051105  047522  EM114:  .ASCIZ  /"ERROR" NOT SET WITH "FR" FLAG/
3358  021254  021122  047040  052117
3359  021262  051440  052105  053440
3360  021270  052111  020110  043042
3361  021276  021122  043040  040514
3362  021304  000107
3363  021306  041522  020126  041501  EM115:  .ASCIZ  /RCV ACTIVE NOT CLEAR WITH INIT/
```

```
3364  021314  044524  042526  047040
3365  021322  052117  041440  042514
3366  021330  051101  053440  052111
3367  021336  020110  047111  052111
3368  021344     000
3369  021345     122  053103  040440  EM116:  .ASCIZ  /RCV ACTIVE WITHOUT "START" BIT/
3370  021352  052103  053111  020105
3371  021360  044527  044124  052517
3372  021366  020124  051442  040524
3373  021374  052122  020042  044502
3374  021402  000124
3375  021404  042122  020122  047105  EM117:  .ASCIZ  /RDR ENABLE NOT CLEAR WITH RCV ACTIVE/
3376  021412  041101  042514  047040
3377  021420  052117  041440  042514
3378  021426  051101  053440  052111
3379  021434  020110  041522  020126
3380  021442  041501  044524  042526
3381  021450     000
3382
3383  021451     124  051505  021524  DH1:    .ASCIZ  /TEST#   ERROR#  TCSR/
3384  021456  020040  042440  051122
3385  021464  051117  020043  052040
3386  021472  051503  000122
3387  021476  042524  052123  020043  DH2:    .ASCIZ  /TEST#   ERR PC  TBUF/
3388  021504  020040  051105  020122
3389  021512  041520  020040  041124
3390  021520  043125     000
3391  021523     124  051505  021524  DH6:    .ASCIZ  /TEST#   ERR PC  RCSR/
3392  021530  020040  042440  051122
3393  021536  050040  020103  051040
3394  021544  051503  000122
3395  021550  042524  052123  020043  DH7:    .ASCIZ  /TEST#   ERR PC  RBUF/
3396  021556  020040  051105  020122
3397  021564  041520  020040  041122
3398  021572  043125     000
3399  021575     124  051505  021524  DH10:   .ASCIZ  /TEST#   ERR PC  LKS/
3400  021602  020040  042440  051122
3401  021610  050040  020103  046040
3402  021616  051513     000
3403  021621     124  051505  021524  DH40:   .ASCIZ  /TEST#   ERR PC  GOOD    BAD/
3404  021626  020040  042440  051122
3405  021634  050040  020103  043440
3406  021642  047517  020104  020040
3407  021650  041040  042101     000
3408  021655     124  051505  021524  DH53:   .ASCIZ  /TEST#   ERR PC  LKS     CNT1    CNT2/
3409  021662  020040  042440  051122
3410  021670  050040  020103  046040
3411  021676  051513  020040  020040
3412  021704  041440  052116  020061
3413  021712  020040  041440  052116
3414  021720  000062
3415  021722  042524  052123  020043  DH103:  .ASCIZ  /TEST#   ERR PC  RCSR    DATA/
3416  021730  020040  051105  020122
3417  021736  041520  020040  041522
3418  021744  051123  020040  020040
3419  021752  040504  040524     000
```

```
3420  021757    124  051505  021524  DH105:  .ASCIZ  /TEST#   ERR PC  RCSR    GOOD    BAD/
3421  021764 020040  042440  051122
3422  021772 050040  020103  051040
3423  022000 051503  020122  020040
3424  022006 043440  047517  020104
3425  022014 020040  041040  042101
3426  022022    000
3427  022023    124  051505  021524  DH110:  .ASCIZ  /TEST#   ERR PC  RCSR    TRANS   RCV/
3428  022030 020040  042440  051122
3429  022036 050040  020103  051040
3430  022044 051503  020122  020040
3431  022052 052040  040522  051516
3432  022060 020040  051040  053103
3433  022066    000
3434  022067    124  051505  021524  DH112:  .ASCIZ  /TEST#   ERR PC  RCSR    OLDPC   TRAP ADR/
3435  022074 020040  042440  051122
3436  022102 050040  020103  051040
3437  022110 051503  020122  020040
3438  022116 047440  042114  041520
3439  022124 020040  052040  040522
3440  022132 020120  042101  000122
3441                                  .EVEN
3442  022140 005015  040515  047111  M1:     .ASCIZ  <CR><LF>/MAINDEC-DZDLD-B/
3443  022146 042504  026503  055104
3444  022154 046104  026504  000102
3445                                  .EVEN
3446  022162 005015                 M2:     .ASCII  <CR><LF>
3447  022164 020040  042040  053105  M2A:    .ASCIZ  /   DEVICES UNDER TEST  /
3448  022172 041511  051505  052440
3449  022200 042116  051105  052040
3450  022206 051505  020124  000040
3451
3452                                  .EVEN
3453  022214 001072  001016  002352  DT1:    .WORD   $TESTN,$ERRPC,TCSR,0
3454  022222 000000
3455  022224 001072  001016  002354  DT2:    .WORD   $TESTN,$ERRPC,TBUF,0
3456  022232 000000
3457  022234 001072  001016  002346  DT6:    .WORD   $TESTN,$ERRPC,RCSR,0
3458  022242 000000
3459  022244 001072  001016  002350  DT7:    .WORD   $TESTN,$ERRPC,RBUF,0
3460  022252 000000
3461  022254 001072  001016  002406  DT10:   .WORD   $TESTN,$ERRPC,LKS,0
3462  022262 000000
3463  022264 001072  001016  001020  DT40:   .WORD   $TESTN,$ERRPC,$GDADR,$BDADR,0
3464  022272 001022  000000
3465  022276 001072  001016  002406  DT53:   .WORD   $TESTN,$ERRPC,LKS,FIRST,SECND,0
3466  022304 006600  006602  000000
3467  022312 001072  001016  002346  DT103:  .WORD   $TESTN,$ERRPC,RCSR,$BDDAT,0
3468  022320 001026  000000
3469  022324 001072  001016  002346  DT105:  .WORD   $TESTN,$ERRPC,RCSR,$GDDAT,$BDDAT,0
3470  022332 001024  001026  000000
3471  022340 001072  001016  002346  DT110:  .WORD   $TESTN,$ERRPC,RCSR,XMTCNT,RCVCNT,0
3472  022346 011730  011726  000000
3473  022354 001072  001016  002346  DT112:  .WORD   $TESTN,$ERRPC,RCSR,OLDPC,BDVECT,0
3474  022362 012300  012302  000000
3475         000001                   .END
```

```
ABASE = 176500            119#    232     273
ACDW1 = 000000            232
ACDW2 = 000000            232
ACPUOP= 000000            232     247
ADDW0 = 000000            232
ADDW1 = 000000            232
ADDW10= 000000            232
ADDW11= 000000            232
ADDW12= 000000            232
ADDW13= 000000            232
ADDW14= 000000            232
ADDW15= 000000            232
ADDW2 = 000000            232
ADDW3 = 000000            232
ADDW4 = 000000            232
ADDW5 = 000000            232
ADDW6 = 000000            232
ADDW7 = 000000            232
ADDW8 = 000000            232
ADDW9 = 000000            232
ADEVCT= 000000            232     238
ADEVM = 000000            232     274
ADR     003562            896#    899
ADRTBL  002414            721#    727     894
AENV  = 000000            232     243
AENVM = 000000            232     244
AFATAL= 000000            232     235
AMADR1= 000000            232     260
AMADR2= 000000            232     264
AMADR3= 000000            232     267
AMADR4= 000000            232     270
AMAMS1= 000000            232     254
AMAMS2= 000000            232     262
AMAMS3= 000000            232     265
AMAMS4= 000000            232     268
AMSGAD= 000000            232     240
AMSGLG= 000000            232     241
AMSGTY= 000000            232     234
AMTYP1= 000000            232     255
AMTYP2= 000000            232     263
AMTYP3= 000000            232     266
AMTYP4= 000000            232     269
APASS = 000000            232     237
APRIOR= 000000            232
APTCSU= 000040           2537#   2570
APTENV= 000001           2313    2492    2535#   2563
APTSIZ= 000200            789    2534#
APTSPO= 000100           2494    2536#   2565
APTSZD  003266            800     802     827#
ASWREG= 000000            232     245
ATESTN= 000000            232     236
AUNIT = 000000            232     239
AUSWR = 000400            121#    232     246
AVECT1= 000300            120#    232     271
AVECT2= 000000            232     272
BDVECT  012302           2279#   2285#   3473
```

```
BEGIN   003424          869#   2268
BIT0  = 000001          104#    797    871   1063   1074   1075   1080   1081   1092   1094   1096   1749   1754
                       1759    1990   1997   2002   2006   2019   2023
BIT00 = 000001           94#    104
BIT01 = 000002           93#    103
BIT02 = 000004           92#    102
BIT03 = 000010           91#    101
BIT04 = 000020           90#    100
BIT05 = 000040           89#     99
BIT06 = 000100           88#     98
BIT07 = 000200           87#     97
BIT08 = 000400           86#     96
BIT09 = 001000           85#     95   2324   2463
BIT1  = 000002          103#    869   1305
BIT10 = 002000           94#   1955   2013
BIT11 = 004000           83#   1691   1700   1710   1726
BIT12 = 010000           82#   2897
BIT13 = 020000           81#   2024   2308
BIT14 = 040000           80#   1966   2447
BIT15 = 100000           79#   1971   2029
BIT2  = 000004          102#   1105   1118   1119   1124   1125   1137   1139   1142   1688   1695   1706   1713
                       1722    1728   1736   1740   1750   1762   1773   1778   1791   1799   1802   1810   1815
                       1829    1836   1846   1849   1867   1871   1904   1908   1933   1958   1965   1985   1997
                       2002    2006   2018   2023   2044   2061   2064   2126
BIT3  = 000010          101#
BIT4  = 000020          100#    793    864   1002   1306   1559
BIT5  = 000040           99#    801   2173   2191
BIT6  = 000100           98#   1156   1166   1167   1173   1174   1179   1181   1198   1208   1209   1215   1216
                       1222    1224   1239   1257   1264   1274   1275   1281   1282   1286   1288   1317   1318
                       1342    1374   1383   1406   1414   1428   1438   1456   1467   1474   1481   1496   1503
                       1510    1528   1530   1571   1584   1589   1598   1605   1612   1629   1638   1648   1661
                       1666    1674   1789   1790   1808   1814   1833   1841   1872   1882   1903   1913   1932
                       1941    2121   2125   2127   2128   2141   2160   2193
BIT7  = 000200           97#    799   1350   1351   1353   1354   1382   1405   1437   1473   1502   1506   2072
                       2912    2965
BIT8  = 000400           96#   1060   1982   2015
BIT9  = 001000           95#
BPT   = 000003          124#
BPTVEC= 000014          111#
BUF     011742         2133    2168   2213#
CATCH   012254          135    2277#
CKSWR = 104406         2323    2446   2878#
CLK     011720         2123    2204#
CLKCNT  011732         2162    2204*  2209#
CMPARE  006536         1546    1549#
COMP    011560         2170#   2174
CONT    005566         1360#   1363
CONT41  010724         1993    2000#
CR    = 000015           19#   2258   2609   2619   2930   2932   2934   2936   3442   3446
CRBUF   002370          708#   2894   2895   2969*
CRCSR   002366          707#    819    875   2892   2963
CRLF  = 000200           20#   2580   2619
CRPSW   002400          712#
CRVECT  002376          711#
CTBUF   002374          710#   2891*  2894*  2925*  2956*
CTCSR   002372          709#   2923   2954
```

```
CTPSW   002404           714#
CTSTFL  002336           690#   742*   795*   874*   921    943    967    981   1065   1083   1108   1128   1237
                        1255   1340   1372   1426   1465   1494   1526   2119   2158   2222*
CTVECT  002402           713#
DDISP = 177570            26#   215    777
DEVADR  002514           727#   804    835
DH1     021461           295    305    310    315    335    340    345    350    355    360    365    370    375
                         380    385    390    395    510    515    520    525    530    535    540   3383#
DH10    021575           330    425    430    435    440    445    455    460    465    470    475    480    485
                         490    495    500    666   3399#
DH103   021722           626   3415#
DH105   021757           636    641    656   3420#
DH110   022023           651   3427#
DH112   022067           661   3434#
DH2     021476           300   3387#
DH40    021621           450   3403#
DH53    021655           505   3408#
DH6     021523           320    400    405    410    415    420    545    550    555    560    565    570    575
                         580    585    590    595    600    605    610    616    621    631    646    671    676
                         681    686   3391#
DH7     021550           325   3395#
DISPLA  001042           215#   777*   785*  2303*  2474*
DISPRE  000174           145#   785
DSWR  = 177570            25#   214    776
DT1     022214           296    306    311    316    336    341    346    351    356    361    366    371    376
                         381    386    391    396    511    516    521    526    531    536    541   3453#
DT10    022254           331    426    431    436    441    446    456    461    466    471    476    481    486
                         491    496    501    667   3461#
DT103   022312           627   3467#
DT105   022324           637    642    657   3469#
DT110   022340           652   3471#
DT112   022354           662   3473#
DT2     022224           301   3455#
DT40    022264           451   3463#
DT53    022276           506   3465#
DT6     022234           321    401    406    411    416    421    546    551    556    561    566    571    576
                         581    586    591    596    601    606    611    617    622    632    647    672    677
                         682    687   3457#
DT7     022244           326   3459#
DVADT   003314           833    835#
ECHO    014762           150   2889#  2918
EMTVEC= 000030           114#   761*   762*
EM1     015266           294   2974#
EM10    015532           329   3006#
EM100   020546           609   3298#
EM101   020611           615   3304#
EM102   020637           620   3308#
EM103   020676           625   3314#
EM104   020734           630   3319#
EM105   020771           635   3324#
EM106   021014           640   3328#
EM107   021052           645   3333#
EM11    015555           334   3010#
EM110   021104           650   3338#
EM111   021134           655   3342#
EM112   021174           660   3348#
```

```
EM113   021211            665    3351#
EM114   021247            670    3357#
EM115   021306            675    3363#
EM116   021345            680    3369#
EM117   021404            685    3375#
EM12    015620            339    3016#
EM13    015651            344    3021#
EM14    015704            349    3026#
EM15    015745            354    3032#
EM16    016010            359    3038#
EM17    016041            364    3043#
EM2     015312            299    2978#
EM20    016074            369    3048#
EM21    016135            374    3054#
EM22    016201            379    3061#
EM23    016236            384    3066#
EM24    016267            389    3071#
EM25    016322            394    3076#
EM26    016363            399    3082#
EM27    016426            404    3088#
EM3     015336            304    2982#
EM30    016465            409    3094#
EM31    016516            414    3099#
EM32    016551            419    3104#
EM33    016617            424    3111#
EM34    016661            429    3117#
EM35    016717            434    3123#
EM36    016747            439    3128#
EM37    017001            444    3133#
EM4     015403            309    2989#
EM40    017041            449    3139#
EM41    017067            454    3143#
EM42    017127            459    3149#
EM43    017161            464    3154#
EM44    017212            469    3159#
EM45    017246            474    3164#
EM46    017303            479    3170#
EM47    017303            484    3169#
EM5     015425            314    2993#
EM50    017337            489    3175#
EM51    017364            494    3179#
EM52    017414            499    3183#
EM53    017471            504    3191#
EM54    017515            509    3195#
EM55    017653            514    3213#
EM56    017553            519    3201#
EM57    017611            524    3207#
EM6     015462            319    2998#
EM60    017653            529    3214#
EM61    017702            534    3218#
EM62    017726            539    3222#
EM63    017773            544    3229#
EM64    020017            549    3233#
EM65    020043            554    3237#
EM66    020111            559    3244#
EM67    020147            564    3250#
```

```
EM7     015506            324    3002#
EM70    020212            569    3256#
EM71    020257            574    3263#
EM72    020426            579    3282#
EM73    020321            584    3269#
EM74    020357            589    3275#
EM75    020426            594    3283#
EM76    020455            599    3287#
EM77    020501            604    3291#
ENDEV   012052           2187    2218#
ENDMG   012172           2245    2258#
ERRVEC= 000004            107#    774     775*    786*    2452    2453*   2455*   2458*
FIRST   006600           1547*   1549    1563#   3465
GNS     = ****** U       2871    2872    2873    2874    2876    2878    2879    2880
GOAGIN  012206           2255    2261#
GTSWR   = 104405         2876#
HT      = 000011            17#  2578    2619
ID      004056            796     977     984     992#
INIT    003112            794     797#
IOTVEC= 000020            112#    759*    760*
LF      = 000012            18#  2258    2613    2619    2930    2932    2934    2936    3442    3446
LKS     002406            717#   1243    1264    1274*   1275    1281*   1282    1286*   1288    1308    1345    1350*   1351
                         1353*   1354    1360    1382*   1383*   1384    1392*   1396    1405*   1406*   1407    1414*   1437*
                         1438*   1439    1456*   1473*   1474*   1475    1481*   1502*   1503*   1504    1506*   1510*   1530*
                         1535*   1536    1538*   1539    1542*   2125*   3461    3465
MANL    003132            798     801#
MFR     015145           2906    2932#
MOP     015157           2911    2934#
MPAR    015134           2901    2930#
MSG     015106           2900    2905    2910    2915    2921#   2960
MSTOP   015172           2916    2936#
M1      022140            999    3442#
M2      022162           1001    3446#
M2A     022164            866*   3447#
OLDPC   012300           2280*   2284#   3473
OUTTST  015202            151    2952#   2966    2971
PIRQ    = 177772            24#
PIRQVE= 000240            118#
PR0     = 000000             41#
PR1     = 000040             42#
PR2     = 000100             43#
PR3     = 000140             44#
PR4     = 000200             45#
PR5     = 000240             46#
PR6     = 000300             47#
PR7     = 000340             48#
PS      = 177776             21#    22
PSW     = 177776             22#
PWRVEC= 000024            113#    765*    766*    2391*   2392*   2401*   2409*   2418*   2419*
RBUF    002350            697#   1044    1777    1909*   1966    1971    1989*   2000    2024    2029    2052    2089    2199
                         3459
RCSR    002346            696#    876     895    1030    1198    1208*   1209    1215*   1216    1222*   1224    1310    1691
                         1700    1710    1719    1726    1733    1740*   1749*   1752    1754    1759*   1760    1775    1779
                         1790*   1797    1808*   1814*   1831    1833*   1841*   1869    1872*   1883*   1903*   1906    1913*
                         1932*   1935    1941*   1987    1992    2021    2050    2082    2128*   3457    3467    3469    3471
                         3473
```

```
RCV     011704          2117    2199#
RCVCNT  011726          2130*   2147    2200*   2207#   3471
RDCHR = 104407          2811    2879#
RDLIN = 104410          2880#
RESVEC= 000010           108#
RPSW    002360           701#   1862    1864*   1885*   2113    2118*   2185*
RSTRT   012232          2263    2267#
RTCPSW  002412           719#   1379    1381*   1416*   1432    1434*   1458*   2114    2124*   2186*
RTCVT   002410           718#   1260    1261*   1293*   1378    1380*   1393*   1404*   1415*   1431    1433*   1446*   1457*
                        1471    1472*   1487*   1500    1501*   1517*   2111    2123*   2183*
RVECT   002356           700#   1194    1195*   1229*   1792    1793*   1807*   1817*   1827    1828*   1842*   1853*   1861
                        1863*   1879*   1884*   1891*   1901    1902*   1919*   1930    1931*   1948*   2110    2117*   2182*
SCPSW   011740          2114*   2186    2212#
SECND   006602          1556*   1564#   3465
SETADR  003530           884     890#
SHIFT   003370           856#    865
SIZE    003142           804#
SRPSW   011736          2113*   2185    2211#
STACK = 001100            12#
START   002546           149     739#
STKLMT= 177774            23#
STPSW   011734          2112*   2184    2210#
SWR     001040           214#    755     776*    778     784*    791*    793     801     1002    1060    1239    1257    1306
                        1342    1374    1428    1467    1496    1528    1559    1955    1982    2013    2015    2072    2121
                        2160    2308    2320    2324    2399    2411*   2447    2463    2708    2745*
SWREG   000176           146#    784    2708    2721
SW0   = 000001            76#
SW00  = 000001            66#
SW01  = 000002            65#
SW02  = 000004            64#
SW03  = 000010            63#
SW04  = 000020            62#
SW05  = 000040            61#
SW06  = 000100            60#
SW07  = 000200            59#
SW08  = 000400            58#
SW09  = 001000            57#
SW1   = 000002            75#
SW10  = 002000            56#
SW11  = 004000            55#
SW12  = 010000            54#
SW13  = 020000            53#
SW14  = 040000            52#
SW15  = 100000            51#
SW2   = 000004            74#
SW3   = 000010            73#
SW4   = 000020            72#
SW5   = 000040            71#
SW6   = 000100            70#
SW7   = 000200            69#
SW8   = 000400            68#
SW9   = 001000            67#
TBITVE= 000014           109#
TBUF    002354           699#    939     958*    963*    1011*   1014*   1667*   1670*   1690*   1717*   1751*   1774*   1796*
                        1830*   1868*   1905*   1934*   1960*   1986*   2020*   2049*   2080*   2134*   2195*   3455
TCLOCK  005056          1004    1232#
```

```
TCONS   003450          872    874*
TCSR    002352          698*   917    959    964    976   1012   1017   1063   1074*  1075   1080*  1081   1092*
                       1094   1096*  1105   1118*  1119   1124*  1125   1137*  1139   1142*  1156   1166*  1167
                       1173*  1174   1179*  1181   1571*  1574   1584*  1589*  1598*  1603   1605*  1612*  1629*
                       1636   1638*  1648*  1661*  1666*  1668   1674*  1688*  1695*  1706*  1713*  1722*  1728*
                       1736*  1750*  1762*  1773*  1778*  1789*  1791*  1799*  1802*  1810*  1815*  1829*  1836*
                       1846*  1849*  1867*  1871*  1904*  1908*  1933*  1958*  1961   1965*  1985*  1990*  1997*
                       2002*  2006*  2018*  2019*  2023*  2044*  2061*  2064*  2126*  2127*  2141   2193*  3453
TKVEC = 000060          116#
TMP1    002340          691#   870*   886*   892    893*
TMP2    002342          692#   805*   813*   823*   827*   831*   851    2220   2262
TMP3    002344          693#   869*   883    885*   891*
TOLER   006550         1551   1553#
TPSW    002364          703#   878    1631   1633*  1655*  2112   2116*  2184*
TPVEC = 000064          117#
TRAPVE= 000034          115#   763*   764*
TRTVEC= 000014          110#
TSTDEV  003500          873    883#   888    2223
TSTDVM  003276          829#   834
TST1    003622          880    906    914#   2265
TST10   004424         1103#
TST11   004572         1061   1095   1150#
TST12   004724         1140   1192#
TST13   005064         1236#
TST14   005134         1238   1240   1254#
TST15   005304         1256   1258   1299#
TST16   005474         1339#
TST17   005602         1341   1343   1361   1371#
TST2    003674          936#
TST20   006020         1373   1375   1425#
TST21   006166         1427   1429   1464#
TST22   006302         1466   1468   1493#
TST23   006430         1495   1497   1525#
TST24   006604         1527   1529   1560   1570#
TST25   006704         1597#
TST26   007016         1628#
TST27   007144         1660#
TST3    003746          957#
TST30   007250         1686#
TST31   007470         1741   1747#
TST32   007560         1761   1771#
TST33   007626         1780   1788#
TST34   007776         1823#
TST35   010140         1859#
TST36   010306         1897#
TST37   010422         1926#
TST4    004126          989   1003   1010#
TST40   010530         1954#
TST41   010626         1956   1981#
TST42   010750         1983   2012#
TST43   011054         2014   2016   2040#
TST44   011154         2071#
TST45   011252         2073   2094   2105#
TST5    004162         1018   1027#
TST6    004214         1041#
TST7    004246         1059#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
TVECT   002362            702#  1152  1153*  1186*  1572  1573*  1583*  1591*  1601  1602*  1613*  1622*  1630
                         1632*  1644*  1654*  1664  1665*  1680*  2109  2115*  2181*
TYPE  = 104401            998  1000  2245  2311  2346  2363  2365  2368  2370  2374  2381  2423  2583
                         2679  2719  2720  2723  2736  2747  2766  2815  2818  2822  2871#
TYPOC = 104402           2354  2378  2722  2872#
TYPON = 104404           2874#
TYPOS = 104403           2873#
VCTADR  003320            825  839#
VCTTBL  002454            722#  839  901
VECT    003602            902#  905
WACTV   007270           1691#  1694
WAIT    015114           2923#  2924  2927
WDONE   007376           1718#  1721
WRPSW   012242           1154  1196  1262  1376  1394  1435  1447  1469  1478  1498  1507  1518  1576
                         1599  1614  1634  1645  1662  1671  1794  1825  1843  1865  1880  1899  1910
                         1928  1938  2107  2135  2270#
WT      007340           1707#  1709
XCONT   011674           2192  2195#
XMIT    011650           2115  2189#
XMTCNT  011730           2129*  2147  2189*  2208#  3471
XRET    011700           2194  2196#
$APTHD  000500            173  179#
$ASTAT= ****** U          926  948  972  986  1070  1088  1114  1133
$ATYC   013176           2485  2487#
$ATY1   013152           2483#
$ATY3   013160           2484#  2568
$ATY4   013170            926  948  972  986  1070  1088  1114  1133  2316  2486#
$AUTOB  001034            211#  2716  2837
$BASE   001142            273#  728  809  817
$BDADR  001022            206#  1311*  1312*  1313  1315*  1316  1317*  1320*  3463
$BDDAT  001026            208#  1992*  2060*  2097*  2177*  3467  3469
$CHARC  013676           2585*  2595*  2602  2611*  2616#
$CKSWR  014130           2708#  2878
$CMTAG  001000            194#  752  753  761  767  768
$CM3  = 000000            224#
$CNTLG  014655           2719  2832#
$CNTLU  014650           2736  2831#
$CPUOP  001114            247#
$CRLF   001063            226#  2311  2346  2365  2370  2374  2584  2619  2747  2831  2962
$DEVCT  001076            238#  743*  996  2219*  2220  2261*
$DEVM   001144            274#  806*  812*  815*  822*  828  871  883
$DOAGN  012162           2241  2247  2253#
$ENDAD  012152            160  2249#  2331
$ENDCT  012132            767  2243#
$ENULL  012166           2256#
$ENV    001106            243#  797  2313  2492  2516  2563
$ENVM   001107            244#  789  799  2494  2565  2570
$EOP    012102           1561  2221  2234#
$EOPCT  012124            767*  2240#  2244
$ERFLG  001003            197#  2301*  2438  2461  2467*  2477
$ERMAX  001015            203#  769*  2473*  2477
$ERROR  012304            761  2300#
$ERRPC  001016            204#  2305*  2306*  2307  2352  3453  3455  3457  3459  3461  3463  3465  3467
                         3469  3471  3473
$ERRTB  001146            291#  2360
$ERRTY  012466           2310  2345#
```

```
$ERTTL  001012            201#   2304*
$ESCAP  001060            224#    768*   2327    2329    2472*
$ETABL  001106            242#
$ETEND  001146            185     275#
$FATAL  001070            235#    739*   2520*
$FFLG   013416           2483*   2486*   2514    2523*   2531#
$FILLC  001056            222#   2588    2619
$FILLS  001055            221#   2619
$GDADR  001020            205#   1308*   1310*   1311    1313    1318    3463
$GDDAT  001024            207#   1316*   1320    2059*   2096*   2178*   3469
$GET42  012142           2246#
$GTSWR  014200           2720#   2876
$HIBTS  000500            180#
$ICNT   001004            198#
$ILLUP  012762           2391    2409    2426#
$INTAG  001035            212#   2748    2837
$ITEMB  001014            202#   2307*   2315    2349
$LF     001064            227#   2619    2822    2831
$LFLG   013415           2524*   2530#
$LPADR  001006            199#    770*   2465*   2470*   2475    2477
$LPERR  001010            200#    771*   2326    2465    2471*   2477
$MADR1  001120            260#
$MADR2  001124            264#
$MADR3  001130            267#
$MADR4  001134            270#
$MAIL   001066            181     185     233#    788     2469    2563
$MAMS1  001116            254#
$MAMS2  001122            262#
$MAMS3  001126            265#
$MAMS4  001132            268#
$MBADR  000502            181#
$MFLG   013414           2484*   2490    2525*   2529#
$MNEW   014673           2723    2835#
$MSGAD  001102            240#   2500*   2503
$MSGLG  001104            241#   2505*
$MSGTY  001066            234#    740*   2498    2506*   2518    2521*
$MSWR   014662           2720    2833#
$MTYP1  001117            255#
$MTYP2  001123            263#
$MTYP3  001127            266#
$MTYP4  001133            269#
$NULL   001054            220#   2590    2619
$NWTST= 000001            911#    933#    954#   1007#   1024#   1038#   1056#   1100#   1147#   1189#   1233#   1251#   1296#
                         1336#   1368#   1422#   1461#   1490#   1522#   1567#   1594#   1625#   1657#   1683#   1744#   1768#
                         1785#   1820#   1856#   1894#   1923#   1951#   1978#   2009#   2037#   2068#   2102#
$OCNT   014124           2651*   2680*   2693#
$OMODE  014126           2646*   2650*   2655    2658*   2669*   2695#
$OVER   013136           2448    2466    2474#
$PASS   001074            237#    788*    994    2237*   2238*   2256
$PASTM  000506            183#
$POWER  012770           2424    2429#
$PWRDN  012622            765    2391#   2418
$PWRMG  012756           2424#
$PWRUP  012674           2401    2409#
$QUES   001062            225#   2619    2766    2815    2831
$RDCHR  014412           2779#   2879
```

```
$RDDEC= ****** U        2881
$RDLIN  014532          2807#   2880
$RDOCT= ****** U        2881
$RDSZ = 000010          2800#
$RTNAD  012164          2255#
$R2A  = ****** U        2881
$SAVRE= ****** U        2881
$SAVR6  012766          2400*   2410    2420*   2421*   2428#
$SCOPE  013000           759    2445#
$SETUP= 000137           142#    758     759     761     763     765     767     768     770    2236    2446    2703    2837
$STUP = 177777           142#
$SVLAD  013102          2456    2462    2468#
$SVPC = 000500           158#    163
$SWR  = 161000           123#    224     225     768     770     771     915     937     958    1011    1028    1042    1060
                        1104    1151    1193    1237    1255    1300    1340    1372    1425    1465    1494    1526    1571
                        1598    1629    1661    1687    1748    1772    1789    1824    1860    1898    1927    1955    1982
                        2013    2041    2072    2106    2231    2237    2248    2254    2256    2439    2440    2441    2442
                        2447    2459    2461    2462    2463    2468    2471    2474    2477
$SWREG  001110           245#    791
$SWRMK= 000000          2442
$TESTN  001072           236#    741*    2469*   3453    3455    3457    3459    3461    3463    3465    3467    3469    3471
                        3473
$TKB    001046           217#   2701    2712    2729    2783    2789
$TKS    001044           216#   2701    2710    2726    2750*   2781    2787
$TN   = 000046           122#    911     915#    933     937#    954     958#   1007    1011#   1024    1028#   1038    1042#
                        1056    1060#   1100    1104#   1147    1151#   1189    1193#   1232    1233    1237#   1251    1255#
                        1296    1300#   1336    1340#   1368    1372#   1422    1426#   1461    1465#   1490    1494#   1522
                        1526#   1567    1571#   1594    1598#   1625    1629#   1657    1661#   1683    1687#   1744    1748#
                        1768    1772#   1785    1789#   1820    1824#   1856    1860#   1894    1898#   1923    1927#   1951
                        1955#   1978    1982#   2009    2013#   2037    2041#   2068    2072#   2102    2106#
$TPB    001052           219#   2608*   2619
$TPFLG  001057           223#   2557    2619
$TPS    001050           218#   2606    2619
$TRAP   014704           763    2848#
$TRAP2  014726          2859#   2870
$TRP  = 000011          2863#   2872#   2873#   2874#   2875#   2876    2877#   2878    2879#   2880#   2881#
$TRPAD  014740          2853    2870#
$TSTM   000504           182#
$TSTNM  001002           196#   1232*   2218*   2236*   2303    2438    2468*   2469    2474    2477
$TTYIN  014640          2808    2809    2826    2830#
$TYPBN= ****** U        2875
$TYPDS= ****** U        2875
$TYPE   013420          2511    2557#   2863    2871
$TYPEC  013632          2587    2594    2601    2606#   2607    2752
$TYPEX  013700          2612    2614    2617#
$TYPOC  013726          2649#   2872
$TYPON  013742          2648    2651#   2874
$TYPOS  013702          2644#   2873
$UNIT   001100           239#    744*    887*    890*   2267*
$UNITM  000510           184#
$USWR   001112           246#    745     747*   2053    2090
$VECT1  001136           271#    840
$VECT2  001140           272#
$XTSTR  013012          2450#
$$GET4= 000000          2248#
$OFILL  014125          2645*   2649*   2659    2694#
```

```
$40CAT= ****** U          2447
.     = 022370             127#    133     134#    138#    144#    148#    153#    158     159#    161#    163#    169     170#
                           172#    174#    193#    228     721#    722#    756     770     771     2213#   2256    2257#   2384#
                          2403    2427    2477    2533#   2619    2701    2830#   2831    2837    2940#
.$ERRT  001146             293#
.$X     000500             169#    174
```

```
CLKTST      6#   1237   1255   1340   1372   1426   1465   1494   1526
COMMEN    119#
CTESTS      8#   1232
ENDCOM    119#
ENDPAS      7#   2245
ERROR      13#    923    945    969    983   1020   1034   1048   1067   1078   1085   1097   1110   1122   1130
         1143   1158   1163   1170   1176   1184   1200   1205   1212   1219   1227   1247   1266   1271   1278
         1284   1291   1328   1348   1357   1365   1390   1402   1411   1443   1453   1485   1515   1557   1581
         1587   1610   1620   1641   1652   1678   1697   1703   1714   1723   1729   1737   1757   1763   1781
         1804   1811   1838   1851   1875   1889   1917   1946   1968   1973   1998   2004   2027   2032   2062
         2087   2099   2145   2151   2165   2179   2281
ESCAPE    119#
GETPRI    119#
GETSWR    119#
MULT      119#
NEWTST    119#    911    933    954   1007   1024   1038   1056   1100   1147   1189   1233   1251   1296   1336
         1368   1422   1461   1490   1522   1567   1594   1625   1657   1683   1744   1768   1785   1820   1856
         1894   1923   1951   1978   2009   2037   2068   2102
POP         9#    119#
PUSH        9#    119#
REPORT    119#    925    947    971    985   1069   1087   1113   1132
SCOPE      14#    914    936    957   1010   1027   1041   1059   1103   1150   1192   1236   1254   1299   1339
         1371   1425   1464   1493   1525   1570   1597   1628   1660   1686   1747   1771   1788   1823   1859
         1897   1926   1954   1981   2012   2040   2071   2105   2235
SETPRI    119#
SETTRA   2863#   2872   2873   2874   2876   2878   2879   2880
SETUP       9#    119#    751
SKIP      119#
SLASH     119#
SPACE     119#
STARS       9#    119#    128    156    166    168    175    189    228    231    911    913    933    935    954
          956   1007   1009   1024   1026   1038   1040   1056   1058   1100   1102   1147   1149   1189   1191
         1233   1235   1251   1253   1296   1298   1336   1338   1368   1370   1422   1424   1461   1463   1490
         1492   1522   1524   1567   1569   1594   1596   1625   1627   1657   1659   1683   1685   1744   1746
         1768   1770   1785   1787   1820   1822   1856   1858   1894   1896   1923   1925   1951   1953   1979
         1980   2009   2011   2037   2039   2068   2070   2102   2104   2229   2288   2298   2340   2388   2390
         2406   2408   2435   2479   2481   2542   2621   2700   2703   2771   2800   2842   2883   2888   2944
         2950
SWRSU     119#    772#
TRMTRP   2863#
TYPBIN    119#
TYPDEC    119#
TYPNAM    119#
TYPNUM    119#
TYPOCS    119#
TYPOCT    119#   2352   2376   2721
TYPTXT    119#
$CLKTS      5#   1237   1255   1340   1372   1426   1465   1494   1526
$$CMRE    187#
$$CMTM    187#
$$ESCA    119#
$$NEWT    119#    911    933    954   1007   1024   1038   1056   1100   1147   1189   1233   1251   1296   1336
         1368   1422   1461   1490   1522   1567   1594   1625   1657   1683   1744   1768   1785   1820   1856
         1894   1923   1951   1978   2009   2037   2068   2102
$$SET    2863#   2872   2873   2874   2876   2878   2879   2880
$$SETM    788#
```

```
$$SKIP    119#
.EQUAT      9#
.SETUP      9#     142
.$ACT1      9#     154
.$APTB      9#     229#
.$APTH      9#     164
.$CMTA      9#     187
.$EOP       9#    2227
.$ERRT      9#    2338
.$READ      9#    2698
.$SCOP      9#    2433
.$TRAP      9#    2840
.$TYPE      9#    2540
.$TYPO      9#    2619


. ABS.  022370     000

 ERRORS DETECTED:  0

 DSKZ:DZDLDB,DZDLDB/SOL/CRF=DZDLDB.P11
 RUN-TIME: 21 11 1 SECONDS
 RUN-TIME RATIO: 178/33=5.3
 CORE USED:  21K  (41 PAGES)
```