DLV11

DIAGNOSTIC TEST

MD-11-DVKAE-B

EP DVKAE B DL A

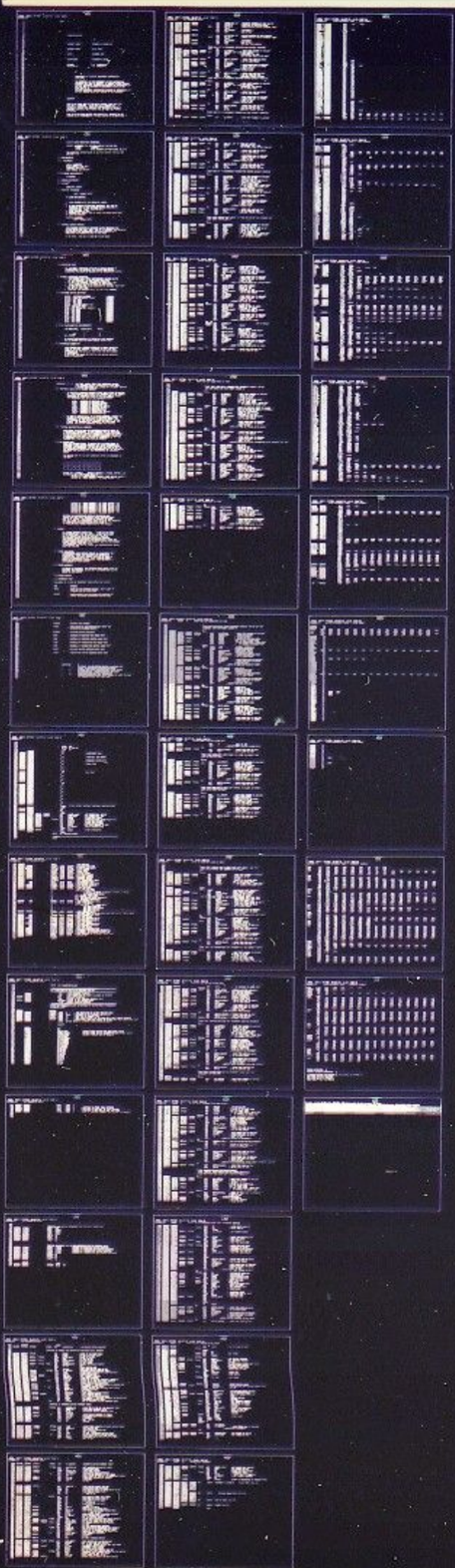
COPYRIGHT 1976

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.



```
                                .REM    %
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DVKAE-B-D
PRODUCT NAME:	DLV11 TEST
DATE:	OCTOBER 1976
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	R.D. FIORENTINO
REVISION B:	D.J. CASALETTO

COPYRIGHT (C) DIGITAL EQUIPMENT CORPORATION
1975,1976

THE MATERIAL IN THIS DOCUMENT IS FOR INFORMATION
PURPOSES ONLY AND IS SUBJECT TO CHANGE WITHOUT NOTICE.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR THE USE OF SOFTWARE ON EQUIPMENT WHICH IS NOT
SUPPLIED BY IT.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR ANY ERRORS WHICH MAY APPEAR IN THE DOCUMENT.

1. ABSTRACT

THIS IS A LOGIC TEST OF THE DLV11 SERIAL LINE UNIT FOR
THE LSI-11 COMPUTER. THIS TEST WILL OPERATE ON
THE DLV11 WITHOUT ANY SPECIAL TEST DEVICES BY DEFAULT.
HOWEVER, A SPECIAL WRAP MODULE CAN BE USED AND TESTED
BY OPTION.

THIS IS A MULTIPLE DEVICE TEST WHICH WILL OPERATE ON THE
CONSOLE DLV11 ADDRESSED AT 177560, AND UP TO 8 ADDITIONAL

```
                        DLV11'S WITH SPECIFIED ADDRESSES.

                        DEFAULT ADDRESSES FOR MULTIPLE DEVICE TESTING ARE:
                                177560          CONSOLE
                                176500          BASE ADDRESS FOR ADDITIONAL DLV11'S

                        FOR COMPLETE INSTRUCTIONS ON MULTIPLE DEVICE TESTING SEE
                        SECTION  5.5

2. REQUIREMENTS

2.1  EQUIPMENT

                        LSI-11 COMPUTER
                        DLV11 SERIAL LINE UNIT
                        TERMINAL FOR DLV11
                        TEST MODULE (BY OPTION)

2.2 STORAGE REQUIREMENTS

                        4K MEMORY

3. LOADING PROCEDURE

3.1 METHOD

                        ABSOLUTE LOADER

4. STARTING PROCEDURE

                                200 - NORMAL ENTRY

                        TO LOAD AND EXECUTE
                        -------------------

                        1. LOAD PROGRAM WITH THE ABSOLUTE LOADER.

                        2. IF ANY PROGRAM OPTIONS ARE REQUIRED, SET THE AP-
                           PROPRIATE BIT IN THE SOFTWARE SWITCH REGISTER AT
                           LOCATION 122.  (REF.  SECTION 5.1)
                        3. START PROGRAM AT 200.
                        4. PROGRAM WILL PRINT "END OF PASS" FOLLOWING EACH PASS.
                        TO LOAD AND EXECUTE

                        STARTING ADDRESS:
                                200 - NORMAL ENTRY


4.1 CONTROL SWITCH SETTING

                        THIS PROGRAM CONTAINS A SOFTWARE SWITCH REGISTER FOR
                        OPTION SELECTION (LOC  422). FOR IT TO OPERATE THE OPERATOR
                        MUST SELECT THE APPROPRIATE OPTION BY SETTING OR
                        RESETTING THE RESPECTIVE BIT IN THE WORD.

                        TO DO THIS , THE LSI-11 MUST BE IN ODT MODE.
```

4.2 EXECUTION TIME

AT 110 BAUD, EXECUTION TIME FOR A SINGLE DEVICE IS APPROXIMATELY 30 SECONDS. FOR HIGHER BAUD RATES, EXECUTION TIME MUST BE REDUCED BY AN APPROPRIATE FACTOR.

5. GENERAL OPERATING PROCEDURE.

1. THE PROGRAM WILL CYCLE CONTINUOUSLY UNLESS HALTED BY THE OPERATOR, OR SOME ERROR CONDITION.
2. TO HALT THE PROGRAM, DEPRESS THE BREAK KEY. ODT WILL DISPLAY THE PC AT WHICH IT WAS HALTED.
3. IF NEW OPTIONS ARE TO BE SELECTED IN THE SWR, THEY MUST BE SET AT THIS TIME.
4. CONTINUE THE PROGRAM VIA A "P" OR A "G" COMMAND.

5.1 SOFTWARE SWITCH SETTINGS

BIT15 - CONTINUE ON ERROR	(100000)
BIT14 - LOOP ON CURRENT ERROR	(040000)
BIT13 - NOT USED	(020000)
BIT12 - NOT USED	(010000)
BIT11 - NOT USED	(004000)
BIT10 - LOOP ON CURRENT TEST	(002000)
BIT9 - RUN WRAP TEST	(001000)
BIT8 - SET DEVICE MAP MANUALLY	(000400)
BIT7 - NOT USED	(000200)
BIT6 - NOT USED	(000100)
BIT5 - NOT USED	(000040)
BIT4 - NOT USED	(000020)
BIT3 - NOT USED	(000010)
BIT2 - NOT USED	(000004)
BIT1 - NOT USED	(000002)
BIT0 - NOT USED	(000001)

5.2 SLU CONFIGURATION REQUIREMENTS

FOR BAUD RATE	#STOP BITS	#BITS
110	2	8
ALL OTHERS	1	8

NO OTHER CONFIGURATION IS TESTED BY THIS PROGRAM
EIA IS TESTED WITH THE WRAP MODULE OPTION TEST.

5.3 OPERATION UNDER APT

THIS PROGRAM WILL OPERATE UNDER APT WITHOUT MODIFICATION TO THE SWR. HOWEVER, IN ORDER TO TEST THE FULL CAPABILITY OF THE DLV11, A TEST WITH THE TEST MODULE IS NECCESSARY.
TO DO THIS, APT MUST SET BIT9 IN THE SWITCH REGISTER PRIOR TO EXECUTION.

IMPORTANT ::::::: THE TEST TIMES, PASS TIME CONSTANTS WITHIN
THIS PROGRAM REFLECT OPERATION AT 9600 BAUD.
A SLOWER BAUD RATE WILL CAUSE APT TO ABORT TESTING

5.4 TESTING A DLV11 AT A UNIQUE ADDRESS

1. TO SPECIFY A CONSOLE ADDRESS OTHER THAN 177560 OR VECTOR 60, THE OPERATOR MUST SUPPLY THE PROGRAM WITH THE CORRECT ADDRESSES BY INSERTING THEM AT THE TAG LABELED "RCSR". THE ADDRESSES MUST BE IN THE FOLLOWING ORDER:

   RCSR: ADDRESS OF RECEIVER CSR
   RBUF: ADDRESS OF RECEIVER BUFFER
   TCSR: ADDRESS OF TRANSMITTER CSR
   TBUF: ADDRESS OF TRANSMITTER BUFFER
   RVECT: ADDRESS OF RECEIVER VECTOR
   RPSW: ADDRESS OF ASSOCIATED PSW
   TVECT: ADDRESS OF TRANSMITTER VECTOR
   TPSW: ADDRESS OF ASSOCIATED PSW

2. TO TEST A SINGLE DLV11 AT A UNIQUE ADDRESS, INSERT THE ADDRESS OF THE RECEIVER CSR IN LOCATION $BASE, AND THE ADDRESS OF THE VECTOR IN LOCATION $VECT1, IN THE ETABLE. THE PROGRAM WILL GENERATE THE ADDITIONAL CSR, BUFFER, AND VECTOR ADDRESSES NEEDED FOR TESTING.

5.5 TESTING MULTIPLE DLV11 MODULES

ADDITIONAL DLV11'S MUST BE ADDRESSED WITHIM A RANGE OF 8 SEQUENTIAL ADDRESSES. A BASE ADDRESS MAY BE SPECIFIED BY THE OPERATOR OR A DEFAULT ADDRESS OF 176500 AND DEFAULT VECTOR OF 300 ARE USED. THE PROGRAM GENERATES A TABLE OF 8 ADDRESSES WITH EACH CSR GIVEN IN INCREMENTS OF 10. THE PROGRAM WILL THEN SIZE FOR DEVICES PRESENT.

5.6 THE DEVICE MAP
WHEN THE PROGRAM SIZES, A DEVICE MAP IS ESTABLISHED TO REPRESENT THOSE DEVICES PRESENT AND THEIR CORRESPONDING RECEIVER CSR ADDRESSES. THE DEVICE MAP CAN ALSO BE SET BY THE OPERATOR BY SELECTING THE APPROPRIATE BIT IN THE SWITCH REGISTER (SEE SECTION 5.1) AND PROCEEDING AS INDICATED BELOW.

THE DEVICE MAP IS A 16 BIT WORD WITH BITS 0-8 ARRANGED AS FOLLOWS:

8	7	6	5	4	3	2	1	0

DEVICE #0 = CONSOLE DEFAULT ADDRESS = 177560; VECTOR = 60
DEVICES #1-#8 = ADDITIONAL DLV11'S
  DEVICE #1 = BASE ADDRESS ($BASE); BASE VECTOR ($VECT1)
  DEVICE #2 = BASE ADDRESS + 10; BASE VECTOR + 10

```
                        DEVICE #3 = BASE ADDRESS + 20; BASE VECTOR + 20
                        DEVICE #4 = BASE ADDRESS + 30; BASE VECTOR + 30
                        DEVICE #5 = BASE ADDRESS + 40; BASE VECTOR + 40
                        DEVICE #6 = BASE ADDRESS + 50; BASE VECTOR + 50
                        DEVICE #7 = BASE ADDRESS + 60; BASE VECTOR + 60
                        DEVICE #8 = BASE ADDRESS + 70; BASE VECTOR + 70

            SETTING THE APPROPRIATE BITS IN THE DEVICE MAP
            ($DEVM LOC 456) WILL DESIGNATE THOSE DEVICES WHICH
            ARE TO BE TESTED. THE PROGRAM IS STARTED AS USUAL,
            BUT WILL COME TO A PROGRAMMED HALT TO ALLOW THE OPERATOR
            TO LOAD THE DEVICE MAP WHILE IN ODT MODE. A
            "P" COMMAND WILL RESUME EXECUTION OF THE PROGRAM.


6. ERRORS

            ALL ERROR REPORTS WITHIN THIS TEST ARE IN THE FORM
            OF AN ERROR HALT. ON THE LSI-11, A HALT WILL FORCE
            ODT TO DISPLAY THE PC+2 OF THE HALT. THIS IS
            THE PRIMARY ERROR INDICATOR WITHIN THE PROGRAM.
            UPON DETECTION OF AN ERROR,THE PROGRAM WILL PLACE THE
            CURRENT ERROR NUMBER AND THE CURRENT TEST IN THE MAILBOX
            (SEE IMPORTANT TAGS SEC. 7.1)
            TO DETERMINE THE TYPE OF ERROR, THE OPERATOR MUST REFER-
            ENCE THE LISTING.


6.1 ERROR ISOLATION
            TO DETERMINE WHICH DEVICE FAILED IN A MULTIPLE TEST
            ENVIRONMENT, CHECK THE RECEIVER CSR ADDRESS AT
            LOCATION 5D6. THE UNIT NUMBER CAN ALSO BE USED TO
            LOCATE THE DEFECTIVE DLV11. THIS LOCATION KEEPS
            TRACK OF THE DEVICE # UNDER TEST. (SEE SECTION 5.6
            FOR ADDRESS OF THAT DEVICE).

6.1 ERROR RECOVERY

            IN ORDER TO CONTINUE, THE OPERATOR MUST ISSUE A "P" TO
            CONTINUE THE PROGRAM, OR MAY SET THE ERROR LOOP SWITCH
            PRIOR TO CONTINUING.

7.0 MISCELLANEOUS

7.1 IMPORTANT TAGS

FOLLOWING IS A LIST OF IMPORTANT TAGS WITHIN THE LISTING

TAG             COMMENT
---             -------
$MAIL           START OF THE PROGRAM MAILBOX. MANY CLUES TO
                PROBLEMS CAN BE FOUND HERE

$FATAL          ERROR NUMBER. USE THE TABLE OF CONTENTS TO
                LOCATE THE ERROR INFORMATION AND/OR CODE
```

```
                        $TESTN          CURRENT TEST NUMBER
                        $PASS           PASS COUNT OF THE PROGRAM WHEN ERROR WAS
                                        DETECTED OR PROGRAM HALTED
                        $SWREG          SOFTWARE SWITCH REGISTER
                        RCSR            ADDRESS OF RECEIVER CSR UNDER TEST
                        RBUF            ADDRESS OF RECEIVER BUFFER UNDER TEST
                        TCSR            ADDRESS OF TRANSMITTER CSR UNDER TEST
                        TBUF            ADDRESS OF TRANSMITTER BUFFER UNDER TEST
                        RVECT           RECEIVER VECTOR ADDRESS BEING USED
                        TVECT           TRANSMITTER VECTOR ADDRESS BEING USED


                                %


                                ;REVISION       B
                                ;       1.      IN TST12, FILLED SECOND BUFFER & CHECKED FOR
                                ;               "DONE" CLEAR BEFORE FLAGGING AN ERROR
                                ;       2.      REMOVED TEST THAT CHECKED"RESET" SETS "DONE"
                                ;               ON TRANSMITTER. (RESET DOES NOT SET DONE)
                                ;       3.      ADDED MULTIPLE SLU TEST CAPABILITY
                                ;       4.      CHANGED PROGRAM TO SKIP TST17 UNDER APT WHEN TESTING
                                ;               THE CONSOLE AFTER THE FIRST PASS
```

```
                                        .ENABL  ABS
                                        .NLIST  MC,CND,MD
                                        .LIST   ME
                                        .=0
000000  000002                          .+2                     ;UNASSIGNED TRAP
                                        0
000002  000004                          .+2                     ;TIME-OUT, BUS ERROR
                                        0
000004  000006                          .+2                     ;RESERVED INSTRUCTION
                                        0
000006  000010                          .+2                     ;TRACE TRAP
                                        0
000010  000012                          .+2
                                        0
000012  000000
                                        .=30
000030  000032                          .+2                     ;EMT TRAP
                                        0
000032  000034                          .+2
                                        0
000034  000036                          .+2
                                        0
000036  000040
                                        .=46
000046  000050                          .+2
                                        0
000050  000052                          .+2
                                        0
000052  000054                          .+2
                                        0
000054  000056                          .+2
                                        0
000056  000060                          .+2
                                        0
000060  000062                          .+2
                                        0
000062  000064                          .+2
                                        0
000064  000066                          .+2
                                        0
000066  000070                          .+2
                                        0
000070  000072                          .+2
                                        0
000072  000074                          .+2
                                        0
000074  000076
                                        .MCALL  .STYPE,.STRAP,.SAPTBLS,.SAPTHDR,.SAPTYPE,STARS
000076  000100
                                        .=100
000100  000102                          102
000102  000104                          RTI
000104  000200
                                        .=200
000200  012737  005426  000024          MOV     #PWRFL,@#24     ;SET POWER FAIL VECTOR
000206  012737  000340  000026          MOV     #340,@#26       ;SET POWER FAIL PSW
000214  005067  000166                  CLR     $PASS           ;CLEAR PASS COUNT
000220  005067  000156          PRCONT: CLR     $FATAL          ;STARTING POINT AFTER POWER FAIL
000224  005067  000154                  CLR     $TESTN          ;CLEAR TEST NUMBER
000230  005067  000154                  CLR     $DEVCT          ;CLEAR DEVICE COUNT
000234  005067  000152                  CLR     $UNIT           ;SET UNIT NUMBER TO ZERO
000240  004767  005062                  JSR     PC,MSGPRT       ;PRINT PROGRAM NAME
000244  006500                          .WORD   M1
000246  000167  000726                  JMP     START           ;START DIAGNOSTIC
000400                                  .=400
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;**************************************************************************
                                .EVEN
```

```
000400                          $MAIL:                  ;;APT MAILBOX
000400  000000                  $MSGTY: .WORD   AMSGTY  ;;MESSAGE TYPE CODE
000402  000000                  $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
000404  000000                  $TESTN: .WORD   ATESTN  ;;TEST NUMBER
000406  000000                  $PASS:  .WORD   APASS   ;;PASS COUNT
000410  000000                  $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
000412  000000                  $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
000414  000000                  $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
000416  000000                  $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
000420                          $ETABLE:                ;;APT ENVIRONMENT TABLE
000420     000                  $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
000421     000                  $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
000422  000000                  $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
000424  000000                  $USWR:  .WORD   AUSWR   ;;USER SWITCHES
000426  000000                  $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
000430     000                  $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
000431     000                  $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
                                ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
                                ;*                              900 NSEC CORE=001
                                ;*                              300 NSEC BIPOLAR=002
                                ;*                              500 NSEC MOS=003
000432  000000                  $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                                ;*                      MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
000434     000                  $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
000435     000                  $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
000436  000000                  $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
000440     000                  $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
000441     000                  $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
000442  000000                  $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
000444     000                  $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
000445     000                  $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
000446  000000                  $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
000450  000300                  $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
000452  000000                  $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
000454  176500                  $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
000456  000000                  $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
000460                          $ETEND:
                                .MEXIT
```

```
                                .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   10      ;;RUN TIM OF LONGEST TEST
                                $PASTM: .WORD   10      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
                                $UNITM: .WORD   0       ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
                                        .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)

                                ;       EQUATES
                                ;
                                        X=1
                                        N=1
                                        AVECT1=300              ;DEFAULT BASE VECTOR FOR ADDITIONAL UNITS
                                        ABASE=176500            ;DEFAULT BASE DEVICE ADDRESS FOR ADDITIONAL UNITS
                                        STKPTR=1000             ;DEFAULT CONSOLE DEVICE ADDRESS
                                        CSR=177560
                                        BIT15=100000
                                        BIT14=40000
                                        BIT13=20000
                                        BIT12=10000
                                        BIT11=4000
                                        BIT10=2000
                                        BIT9=1000
                                        BIT8=400
                                        BIT7=200
                                        BIT6=100
                                        BIT5=40
                                        BIT4=20
                                        BIT3=10
                                        BIT2=4
                                        BIT1=2
                                        BIT0=1
```

```
  461  000474  000000                    TMP0:   .WORD   0
  462  000476  000000                    TMP1:   .WORD   0       ;LOCATION TO STORE TABLE OFFSETS
  463  000500  000000                    TMP2:   .WORD   0       ;LOCATION TO KEEP NO. OF TEST DEVICES
  464  000502  000000                    TMP3:   .WORD   0       ;LOCATION TO KEEP DEVICE MAP TEST MASK
  465  000504  000000                    CTSTFL: .WORD   0       ;FLAG TO INDICATE CURRENT TEST DEVICE IS CONSOLE
  466
  467
  468
```

```
                                        ;
                                        ;REGISTER AND VECTOR ADDRESSES FOR THE DLV11 UNDER TEST
000506  177560                          RCSR:   CSR
000510  177562                          RBUF:   CSR+2
000512  177564                          TCSR:   CSR+4
000514  177566                          TBUF:   CSR+6
000516  000060                          RVECT:  60
000520  000062                          RPSW:   62
000522  000064                          TVECT:  64
000524  000066                          TPSW:   66
                                        ;
                                        ;INITIAL REGISTER AND VECTOR ADDRESSES FOR THE CONSOLE DLV11
000526  177560                          DRCSR:  177560          ;ADDRESS OF RECEIVER STATUS REGISTER
000530  177562                          DRBUF:  177562          ;ADDRESS OF RECEIVER DATA BUFFER REGISTER
000532  177564                          DTCSR:  177564          ;ADDRESS OF TRANSMITTER STATUS REGISTER
000534  177566                          DTBUF:  177566          ;ADDRESS OF TRANSMITTER DATA BUFFER REGISTER
000536  000060                          DRVECT: 60
000540  000062                          DRPSW:  62
000542  000064                          DTVECT: 64
000544  000066                          DTPSW:  66
                                        .=600
000600  000020                  ADRTBL: .BLKW   20
000640  000010                  VCTTBL: .BLKW   10
```

```
                001200                          .=1200
001200  012706  001000          START:  MOV     #STKPTR,SP      ; INITIALIZE STACK POINTER
                                        ;
                                        ;ROUTINE TO HANDLE THE TESTING OF MULTIPLE DLV11 MODULES
                                        ;
001204  005767  177176                  TST     $PASS           ;CHECK IF ON FIRST PASS
001210  001161                          BNE     SIZED           ;IF NOT ON FIRST PASS
001212  132767  000001  177200          BITB    #BIT0,$ENV      ;CHECK IF ON APT
001220  001404                          BEQ     MANL            ;IF NOT ON APT
001222  132767  000200  177171          BITB    #BIT7,$ENVM     ;DID APT SIZE?
001230  001067                          BNE     APTSIZ          ;IF APT SIZED
001232  032767  000400  177162  MANL:   BIT     #BIT8,$SWREG    ;MANUAL SETTING OF $DEVM?
001240  001402                          BEQ     SETTBL          ;NO
001242  000000                          HALT                    ;INPUT DEVICE MAP
001244  000461                          BR      APTSIZ          ;CONT WHEN DEVICE MAP IS SET
001246  004767  000104          SETTBL: JSR     PC,DEVADR       ;GENERATE DEVICE ADDRESS TABLE
001252  005067  177222                  CLR     TMP2            ;CLEAR TEMP LOCATION TO COUNT DEVICES
001256  005067  177174                  CLR     $DEVM           ;CLEAR DEVICE MAP
001262  012737  001316  000004          MOV     #2$,@#4         ;SET TIMEOUT POINTER
001270  016700  177160                  MOV     $BASE,R0        ;LOAD BASE ADDRESS
001274  062700  000100                  ADD     #100,R0         ;POINT R0 TO UNIT#8 (UNIT#0 = CONSOLE)
001300  162700  000010          1$:     SUB     #10,R0          ;POINT R0 TO THE NEXT UNIT ADDRESS
001304  005710                          TST     (R0)            ;CHECK FOR DEVICE EXISTANCE
001306  005267  177144                  INC     $DEVM           ;NO TIMEOUT - INDICATE DEVICE IN DEVICE MAP
001312  005267  177162                  INC     TMP2            ;INCREMENT DEVICE COUNT
001316  006[illegible] 177134          2$:     ASL     $DEVM           ;ADJUST DEVICE MAP FOR NEXT UNIT CHECK
001322  026700  177126                  CMP     $BASE,R0        ;FINISHED SIZING?
001326  002764                          BLT     1$              ;BR IF NOT
001330  016700  177172                  MOV     DRCSR,R0        ;LOAD CONSOLE DEVICE ADRESS
001334  012737  001354  000004          MOV     #3$,@#4         ;SET TIMEOUT VECTOR
001342  005710                          TST     (R0)            ;TEST FOR CONSOLE EXISTANCE
001344  005267  177106                  INC     $DEVM           ;NO TIMEOUT - INDICATE CONSOLE IN DEVICE MAP
001350  005267  177124                  INC     TMP2            ;INCREMENT DEVICE COUNT
001354  000432                  3$:     BR      VCTADR          ;BR TO GENERATE VECTOR ADDRESS TABLE

                                        ;ROUTINE TO GENERATE DEVICE ADDRESS TABLE

001356  012702  000600          DEVADR: MOV     #ADRTBL,R2      ;POINT R2 TO TOP OF THE ADDRESS TABLE
001362  016700  177066                  MOV     $BASE,R0        ;LOAD BASE ADDRESS IN R0
001366  010001                          MOV     R0,R1           ;CALCULATE DEVICE ADDRESS LIMIT
001370  062701  000100                  ADD     #100,R1
001374  010022          1$:     MOV     R0,(R2)+        ;MOVE DEVICE ADDRESS TO TABLE
001376  062700  000010                  ADD     #10,R0          ;CALCULATE NEXT DEVICE ADDRESS
001402  020001                          CMP     R0,R1           ;FINISHED?
001404  002773                          BLT     1$              ;BR, IF NOT
001406  000207                          RTS     PC              ;RETURN

001410  005067  177064          APTSIZ: CLR     TMP2            ;CLEAR TEMP LOCATION TO KEEP DEVICE COUNT
001414  016702  177036                  MOV     $DEVM,R2        ;GET DEVICE MAP
001420  005702          TSTDVM: TST     R2              ;TEST MSB OF DEVICE MAP
001422  100002                          BPL     1$              ;BR, IF CLEAR
001424  005267  177050                  INC     TMP2            ;IF SET, INCREMENT DEVICE COUNT
001430  006302          1$:     ASL     R2              ;SHIFT NEXT BIT INTO MSB POSITION
001432  001401                          BEQ     DVADT           ;FINISHED (NO BITS LEFT SET IN MAP)?
001434  000771                          BR      TSTDVM          ;BR, IF NOT
```

```
                                DVADT:  JSR     PC,DEVADR       ;GENERATE DEVICE ADDRESS TABLE
                                
                                VCTADR: MOV     #VCTTBL,R2      ;GET LOCATION OF VECTOR TABLE
                                        CLR     R1              ;CLEAR DATA WORD
                                        MOVB    $VECT1,R0       ;COPY BASE VECTOR
                                        BIC     #177400,R0      ;CLEAR BYTE SIGN EXTENSION
                                1$:     MOV     R0,(R2)+        ;PUT VECTOR ADDRESS IN THE TABLE
                                        INC     R1              ;KEEP COUNT OF THE NO. OF ADDRESSES STORED
                                        ADD     #10,R0          ;FORM NEXT SEQUENTIAL ADDRESS
                                        CMP     #10,R1          ;FINISHED?
                                        BGE     1$              ;IF NO
                                
                                        MOV     TMP2,R0         ;COPY DEVICE COUNT INTO R0
                                        CLR     R1              ;CLEAR AUXILIARY REGISTER
                                        SWAB    R0              ;PUT DEVICE COUNT IN UPPER BYTE OF R0
                                        ASL     R0              ;MOVE MSB OF COUNT INTO
                                        ASL     R0              ;   MSB OF R0
                                
                                SHIFT:  ASL     R0              ;PUT MSB OF COUNT INTO CARRY
                                        ROLB    R1              ;MOVE MSB OF COUNT INTO R1
                                        ASL     R0              ;MOVE NEXT BIT OF COUNT
                                        ROLB    R1              ;   INTO R1
                                        ASL     R0              ;MOVE LAST BIT OF DIGIT
                                        ROLB    R1              ;   INTO R1
                                        ADD     #60,R1          ;CONVERT TO ASCII
                                        SWAB    R1              ;MOVE DIGIT TO UPPER BYTE
                                        BIT     #BIT4,R1        ;HAVE BOTH DIGITS BEEN MAVED TOR1?
                                        BEQ     SHIFT           ;BR, IF NOT
                                        MOV     R1,M2           ;MOVE DEVICE COUNT TO OUTPUT MESSAGE
                                        JSR     PC,MSGPRT       ;PRINT MESSAGE DEFINING THE
                                        .WORD   M2              ;NO. OF DEVICES SIZED
                                SIZED:  CLR     TMP1            ;CLEAR TEMP LOCATION FOR TABLE OFFSETS
                                        MOV     #BIT1,TMP3      ;SET UP BIT MASK FOR DEVICE MAP
                                TSTTY:  BIT     #BIT0,$DEVM     ;IS CONSOLE DEVICE TO BE TESTED?
                                        BNE     TCONSL          ;IF YES
                                        BR      SETUP           ;GO TEST ADDITIONAL DEVICES
                                TCONSL: MOV     #RCSR,R0        ;SET UPREGISTER ADDRESSES FOR CONSOLE
                                        MOV     #DRCSA,R1       ;POINT R1 TO CONSOLE ADDRESS TABLE
                                        INC     CTSTFL          ;INDICATE THAT CONSOLE IS UNDER TEST
                                1$:     MOV     (R1)+,(R0)+     ;FORM CONSOLE ADDRESSES
                                        CMP     #TPSW,R0        ;FINISHED?
                                        BGE     1$              ;IF NO
                                        JMP     TST1            ;GO TEST CONSOLE DEVICE
                                
                                SETUP:  BIT     TMP3,$DEVM      ;TEST DEVICE BIT IN DEVICE MAP
                                        BNE     SETADR          ;BR, IF SET AND TEST DEVICE
                                        ASL     TMP3            ;IF, CLEAR, UPDATE DEVICE MAP TEST MASK
                                        ADD     #2,TMP1         ;INCREMENT TABLE OFFSET
                                        INC     $UNIT           ;INCREMENT UNIT NUMBER
                                        BR      SETUP           ;BR, TO TEST NEXT DEVICE MAP BIT
                                SETADR: INC     $UNIT           ;INCREMENT UNIT NUMBER BEFORE TESTING DEVICE
                                        ASL     TMP3            ;UPDATE DEVICE MAP TEST MASK
                                        MOV     TMP1,R2         ;MOVE TABLE OFFSET TO R2
                                        ADD     #2,TMP1         ;UPDATE TABLE OFFSET FOR NEXT DEVICE
```

```
                                                MOV     ADRTBL(R2),R0   ;REGISTER ADDRESS OF NEXT DEVICE
                                                MOV     #RCSR,R1        ;LOCATION OF REGISTER ADDR.
                                        ADR:    MOV     R0,(R1)+        ;CREATE ACTIVE DEVICE ADDRESS TABLE
                                                TST     (R0)+
                                                BIT     R0,#000006      ;FINISHED?
                                                BNE     ADR             ;BR, IF NOT

                                                MOV     VCTTBL(R2),R0   ;VECTOR ADDRESS OF NEXT DEVICE
                                        VECT:   MOV     R0,(R1)+        ;CREATE ACTIVE VECTOR ADDRESS TABLE
                                                TST     (R0)+
                                                BIT     R0,#000006      ;FINISHED?
                                                BNE     VECT            ;BR, IF NOT
                                                JMP     TST1            ;GO TEST THIS DEVICE

                                        ;
                                        ;TEST ABILITY TO REFERENCE RECEIVER CSR WITHOUT TRAPPING
                                        ;
                                        TST1:
                                        LP1:    MOV     #1,STESTN       ; MOVE TEST NUMBER TO MAILBOX
                                                MOV     #2$,@#4 ;SET TIME OUT VECTOR
                                                TST     @RCSR           ;REFERENCE THE CSR
                                                BR      1$              ;GO TO NEXT TEST IF NO TRAP
                                        2$:
                                                BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
                                                BNE     LP1             ; GO TO LOOP ERROR
                                                MOV     #1,$FATAL
                                                MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
                                                TST     $SWREG          ; CHECK FOR HALT ON ERROR
                                                BMI     1$              ; HALT IF SET
                                                HALT                    ;<CANNOT ACCESS RCSR>
                                        1$:
                                                BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
                                                BNE     TST1            ; GO TO LOOP ON TEST
                                                MOV     #6,@#4          ;RESTORE TIMEOUT VECTOR
                                        ;TEST ABILITY TO REFERENCE RECEIVER BUFFER WITHOUT TRAPPING
                                        ;
                                        TST2:
                                                MOV     #2,STESTN       ; MOVE TEST NUMBER TO MAILBOX
                                        LP2:    MOV     #2$,@#4 ;SET TIME OUT VECTOR
                                                TST     @RBUF           ;REFERENCE THE BUFFER
                                                BR      1$              ;GO TO NEXT TEST
                                        2$:
                                                BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
                                                BNE     LP2             ; GO TO LOOP ERROR
                                                MOV     #2,$FATAL
                                                MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
                                                TST     $SWREG          ; CHECK FOR HALT ON ERROR
                                                BMI     1$              ; HALT IF SET
                                                HALT                    ;<CANNOT ACCESS RBUF>
                                        1$:
                                                BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
                                                BNE     TST2            ; GO TO LOOP ON TEST
                                                MOV     #6,@#4          ;RESTORE TIMEOUT VECTOR
                                        ;TEST ABILITY TO REFERENCE TRANSMITTER CSR WITHOUT TRAPPING
                                        ;
                                        TST3:
                                                MOV     #3,STESTN       ; MOVE TEST NUMBER TO MAILBOX
```

```
                                LP3:    MOV     #2$,@#4 ;SET UP TIME OUT VECTOR
                                        TST     @TCSR           ;REFERENCE THE T CSR
                                        BR      1$              ;GO TO NEXT TEST IF NO TIME-OUT
                                2$:
                                        BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
                                        BNE     LP3             ; GO TO LOOP ERROR
                                        MOV     #3,$FATAL
                                        MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
                                        TST     $SWREG          ; CHECK FOR HALT ON ERROR
                                        BMI     1$              ; HALT IF SET
                                        HALT            ;<CANNOT ACCESS TCSR>
                                1$:
                                        BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
                                        BNE     TST3            ; GO TO LOOP ON TEST
                                        MOV     #6,@#4          ;RESTORE TIMEOUT VECTOR

                                ;TEST ABILITY TO REFERENCE TRANSMITTER BUFFER WITHOUT TRAPPING

                                TST4:
                                        MOV     #4,$TESTN       ; MOVE TEST NUMBER TO MAILBOX
                                        BIT     #BIT0,$ENV      ; CHECK IF ON APT
                                        BEQ     LP4             ; DO THIS TEST IF NOT ON APT
                                        TST     $PASS           ; CHECK IF FIRST PASS
                                        BEQ     LP4             ; BR IF FIRST PASS
                                        TST     CTSTFL          ;IF NOT FIRST PASS - SKIP TEST
                                        BNE     TST5            ;   IF CONSOLE IS UNDER TEST
                                LP4:    MOV     #2$,@#4 ;SET UP TIME-OUT VECTOR
                                        TST     @TBUF           ;REFERENCE THE T BUFFER
                                        BR      1$              ;GO TO NEXT TEST IF NO TIME-OUT
                                2$:
                                        BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
                                        BNE     LP4             ; GO TO LOOP ERROR
                                        MOV     #4,$FATAL
                                        MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
                                        TST     $SWREG          ; CHECK FOR HALT ON ERROR
                                        BMI     1$              ; HALT IF SET
                                        HALT            ;<CANNOT ACCESS TBUF>
                                1$:
                                        BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
                                        BNE     TST4            ; GO TO LOOP ON TEST
                                        MOV     #6,@#4          ;RESTORE TIMEOUT VECTOR
                                        NOP

                                ;TEST THAT "BREAK" BIT (BIT0) IN TCSR CAN BE SET AND CLEARED AND
                                ;THAT RESET CAN CLEAR IT.

                                TST5:
                                        MOV     #5,$TESTN       ; MOVE TEST NUMBER TO MAILBOX
                                        BIT     #BIT0,$ENV      ; CHECK IF ON APT
                                        BEQ     LP5A            ; DO THIS TEST IF NOT ON APT
                                        TST     $PASS           ; CHECK IF FIRST PASS
                                        BEQ     LP5A            ; BR IF FIRST PASS
                                        TST     CTSTFL          ;IF NOT FIRST PASS - SKIP TEST
                                        BNE     TST6            ;   IF CONSOLE IS UNDER TEST
                                LP5A:   CLR     R4              ; SET UP 300 MS COUNT
                                LP5B:   INC     R4
```

```
                                                BNE     LPSB
                                LP5:            RESET                   ;CLEAR EVERYTHING
                                                BIT     #BIT0,@TCSR     ;CHECK BIT 0 (BREAK)
                                                BEQ     TSCON           ;CONTINUE IF NOT SET
                                                BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
                                                BNE     LP5             ; GO TO LOOP ERROR
                                                MOV     #5,$FATAL
                                                MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
                                                TST     $SWREG          ; CHECK FOR HALT ON ERROR
                                                BMI     1$              ; HALT IF SET
                                                HALT            ;<BIT 0 IN TCSR SHOULD BE 0 AFTER RESET>
                                1$:
                                TSCON:          BIS     #BIT0,@TCSR     ;SET BIT0 (BREAK)
                                                BIT     #BIT0,@TCSR     ;CHECK IF SET
                                                BNE     TSCON1          ;CONTINUE IF SET
                                                BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
                                                BNE     TSCON           ; GO TO LOOP ERROR
                                                MOV     #6,$FATAL
                                                MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
                                                TST     $SWREG          ; CHECK FOR HALT ON ERROR
                                                BMI     1$              ; HALT IF SET
                                                HALT            ;<CANNOT SET BIT0 IN TCSR>
                                1$:
                                TSCON1:         BIC     #BIT0,@TCSR     ;CLEAR BIT0
                                                BIT     #BIT0,@TCSR     ;CHECK TO SEE IF CLEAR
                                                BEQ     TSCON2          ;CONTINUE IF CLEAR
                                                BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
                                                BNE     TSCON1          ; GO TO LOOP ERROR
                                                MOV     #7,$FATAL
                                                MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
                                                TST     $SWREG          ; CHECK FOR HALT ON ERROR
                                                BMI     1$              ; HALT IF SET
                                                HALT            ;<CANNOT CLEAR BIT0 IN TCSR>
                                1$:
                                TSCON2:         MOV     #10000,R4                       ; SET UP DELAY
                                TSDEC:          DEC     R4              ; DELAY
                                                BNE     TSDEC
                                                BIS     #BIT0,@TCSR     ;SET IT AGAIN
                                                RESET                   ;CHECK RESET CLEAR AGAIN
                                                BIT     #BIT0,@TCSR     ;CHECK BIT 0
                                                BEQ     1$              ;CONTINUE IF RESET
                                                BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
                                                BNE     TSCON2          ; GO TO LOOP ERROR
                                                MOV     #10,$FATAL
                                                MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
                                                TST     $SWREG          ; CHECK FOR HALT ON ERROR
                                                BMI     1$              ; HALT IF SET
                                                HALT            ;<RESET DID NOT CLEAR BIT0 IN TCSR>
                                1$:
                                                BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
                                                BNE     TST5            ; GO TO LOOP ON TEST
```

DVKAEB.P11 ERROR 10 RESET DID NOT CLEAR BIT0 IN TCSR

```
                                        ;TEST THAT TCSR BIT6 (TRANSMITTER INTERRUPT ENABLE) CAN BE
                                        ;SET AND RESET AND THAT RESET CLEARS IT.
                                        ;
003702                                  TST6:
003702  012767  000006  175474                MOV     #6,$TESTN       ; MOVE TEST NUMBER TO MAILBOX
003710  032767  000001  175502                BIT     #BIT0,$ENV      ; CHECK IF ON APT
003716  001406                                BEQ     TST6A           ; IF NOT ON APT
003720  005767  175462                        TST     $PASS           ; CHECK IF ON FIRST PASS
003724  001403                                BEQ     TST6A           ;BR, IF FIRST PASS
003726  005767  175552                        TST     CTSTFL          ;IF NOT FIRST PASS - SKIP TEST
003732  001152                                BNE     TST7            ;   IF CONSOLE IS UNDER TEST
003734                                  TST6A:
003734  000005                          LP6:  RESET                   ;SEND THE INITIAL RESET TO CLEAR
003736  012777  003014  175556                MOV     #ERR12,@TVECT   ;SET UP VECTOR IN CASE OF INTERRUPT
003744  106427  000200                        MTPS    #200            ;TURN OFF INTERRUPTS
003750  032777  000100  175534                BIT     #BIT6,@TCSR     ;CHECK IF BIT 6 IS ON
003756  001434                                BEQ     T6CON1          ;CONTINUE IF NOT SET
003760  032767  040000  175434                BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
003766  001362                                BNE     LP6             ; GO TO LOOP ERROR
003770  012767  000011  175404                MOV     #11,$FATAL
003776  012767  000001  175374                MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
004004  005767  175412                        TST     $SWREG          ; CHECK FOR HALT ON ERROR
004010  100401                                BMI     1$              ; HALT IF SET
004012  000000                                HALT            ;<BIT6 IN TCSR NOT CLEAR AFTER RESET>
004014                                  1$:
004014                                  ERR12:
004014  032767  040000  175400                BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
004022  001344                                BNE     LP6             ; GO TO LOOP ERROR
004024  012767  000012  175360                MOV     #12,$FATAL
004032  012767  000001  175340                MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
004040  005767  175356                        TST     $SWREG          ; CHECK FOR HALT ON ERROR
004044  100401                                BMI     1$              ; HALT IF SET
004046  000000                                HALT            ;<DEVICE SHOULD NOT HAVE INTERRUPTED WITH PRIORITY BIT SET>
004050                                  1$:
004050  052777  000100  175434        T6CON1: BIS     #BIT6,@TCSR     ;SET BIT6
004056  032777  000100  175426                BIT     #BIT6,@TCSR     ;CHECK IF SET
004064  001016                                BNE     T6CON2          ;CONTINUE IF SET.
004066  032767  040000  175326                BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
004074  001365                                BNE     T6CON1          ; GO TO LOOP ERROR
004076  012767  000013  175276                MOV     #13,$FATAL
004104  012767  000001  175266                MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
004112  005767  175304                        TST     $SWREG          ; CHECK FOR HALT ON ERROR
004116  100401                                BMI     1$              ; HALT IF SET
004120  000000                                HALT            ;<CANNOT SET BIT6 IN TCSR>
004122                                  1$:
004122  042777  000100  175362        T6CON2: BIC     #BIT6,@TCSR     ;CLEAR BIT6
004130  032777  000100  175354                BIT     #BIT6,@TCSR     ;CHECK IF CLEAR
004136  001416                                BEQ     T6CON3          ;CONTINUE IF CLEAR
004140  032767  040000  175254                BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
004146  001365                                BNE     T6CON2          ; GO TO LOOP ERROR
004150  012767  000014  175224                MOV     #14,$FATAL
004156  012767  000001  175214                MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
004164  005767  175232                        TST     $SWREG          ; CHECK FOR HALT ON ERROR
004170  100401                                BMI     1$              ; HALT IF SET
004172  000000                                HALT            ;<CANNOT CLEAR BIT6 IN TCSR>
```

```
.MAIN.  MACY11 27(732)  04-OCT-76  14:23  PAGE 19
DVKAEB.P11      ERROR   14      CANNOT CLEAR BIT6 IN TCSR

                                        1$:
003174                          175310  T6CON3: BIS     #BIT6,@TCSR     ;SET BIT6 AGAIN
003174  052777  000100                        RESET                   ;TRY RESET AGAIN
003202  000005
003204  032777  000100          175300        BIT     #BIT6,@TCSR     ;CHECK TO SEE IF CLEAR
003212  001416                                BEQ     1$              ;CONTINUE IF CLEAR
003214  032767  040000          175200        BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
003222  001364                                BNE     T6CON3          ; GO TO LOOP ERROR
003224  012767  000015          175150        MOV     #15,$FATAL
003232  012767  000001          175140        MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
003240  005767  175156                        TST     $SWREG          ; CHECK FOR HALT ON ERROR
003244  100401                                BMI     1$              ; HALT IF SET
003246  000000                                HALT                    ;<RESET DID NOT CLEAR BIT6 IN TCSR>
003250                                  1$:
003250  032767  002000          175144        BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
003256  001211                                BNE     TST6            ; GO TO LOOP ON TEST

```

```
                                        ;
                                        ;TEST THAT RCSR BIT6 (RCVR INTERRUPT ENABLE) CAN BE SET AND
                                        ;CLEARED, AND THAT RESET CLEARS IT.
                                        ;
003260                          TST7:
003260  012767  000007  175116          MOV     #7,$TESTN       ; MOVE TEST NUMBER TO MAILBOX
003266  032767  000001  175124          BIT     #BIT0,$ENV      ; CHECK IF ON APT
003274  001406                          BEQ     TST7A           ; IF NOT ON APT
003276  005767  175104                  TST     $PASS           ; CHECK IF IN FIRST PASS
003302  001403                          BEQ     TST7A           ;BR, IF NOT  FIRST PASS
003304  005767  175174                  TST     CTSTFL          ;IF NOT FIRST PASS - SKIP TEST
003310  001131                          BNE     TST10           ;   IF CONSOLE IS UNDER TEST
003312                          TST7A:
003312  106427  000200          LP8:    MTPS    #200            ;DISABLE INTERRUPTS
003316  000005                          RESET                   ;CLEAR EVERYTHING
003320  032777  000100  175160          BIT     #BIT6,@RCSR     ;CHECK BIT6 FOR 0
003326  001416                          BEQ     TBCON1          ;CONTINUE IF RESET
003330  032767  040000  175064          BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
003336  001365                          BNE     LP8             ; GO TO LOOP ERROR
003340  012767  000016  175034          MOV     #16,$FATAL
003346  012767  000001  175024          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
003354  005767  175042                  TST     $SWREG          ; CHECK FOR HALT ON ERROR
003360  100401                          BMI     1$              ; HALT IF SET
003362  000000                          HALT            ;<BIT6 IN RCSR NOT CLEAR BY RESET>
003364                          1$:
003364  052777  000100  175114  TBCON1: BIS     #BIT6,@RCSR     ;SET BIT6
003372  032777  000100  175106          BIT     #BIT6,@RCSR     ;CHECK IF BIT SET
003400  001016                          BNE     TBCON2          ;CONTINUE IF SET
003402  032767  040000  175012          BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
003410  001365                          BNE     TBCON1          ; GO TO LOOP ERROR
003412  012767  000017  174762          MOV     #17,$FATAL
003420  012767  000001  174752          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
003426  005767  174770                  TST     $SWREG          ; CHECK FOR HALT ON ERROR
003432  100401                          BMI     1$              ; HALT IF SET
003434  000000                          HALT            ;<CANNOT SET BIT6 IN RCSR>
003436                          1$:
003436  042777  000100  175042  TBCON2: BIC     #BIT6,@RCSR     ;CLEAR BIT6
003444  032777  000100  175034          BIT     #BIT6,@RCSR     ;CHECK IF CLEAR
003452  001416                          BEQ     TBCON3          ;CONTINUE IF RESET
003454  032767  040000  174740          BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
003462  001365                          BNE     TBCON2          ; GO TO LOOP ERROR
003464  012767  000020  174710          MOV     #20,$FATAL
003472  012767  000001  174700          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
003500  005767  174716                  TST     $SWREG          ; CHECK FOR HALT ON ERROR
003504  100401                          BMI     1$              ; HALT IF SET
003506  000000                          HALT            ;<CANNOT CLEAR BIT6 IN RCSR>
003510                          1$:
003510  052777  000100  174770  TBCON3: BIS     #BIT6,@RCSR     ;SET BIT6 AGAIN
003516  000005                          RESET                   ;ISSUE ANOTHER RESET
003520  032777  000100  174760          BIT     #BIT6,@RCSR     ;CHECK IF RESET CLEARED BIT6.
003526  001416                          BEQ     1$              ;CONTINUE IF CLEAR
003530  032767  040000  174664          BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
003536  001364                          BNE     TBCON3          ; GO TO LOOP ERROR
003540  012767  000021  174634          MOV     #21,$FATAL
003546  012767  000001  174624          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
003554  005767  174642                  TST     $SWREG          ; CHECK FOR HALT ON ERROR
```

```
                                                BMI     1$                      ; HALT IF SET
                                                HALT                    ;<SECOND RESET DID NOT CLEAR BIT6>
                                        1$:
003564  032767  002000  174630                  BIT     #BIT10,@#SWREG          ; CHECK FOR LOOP ON TEST
003572  001232                                  BNE     TST7                    ; GO TO LOOP ON TEST
                                        ;
                                        ;TEST THAT RCSR BIT7 (RCVR DONE) IS CLEAR (BY RESET) AND
                                        ;CAN BE READ RELIABLY.
                                        ;
003574                                  TST10:
003574  012767  000010  174602                  MOV     #10,@#TESTN             ; MOVE TEST NUMBER TO MAILBOX
003602  000005                          LP9:    RESET                           ;CLEAR EVERYTHING
003604  032777  030000  174674                  BIT     #BIT7,@RCSR             ;CHECK IF BIT7 CLEAR.
003612  001416                                  BEQ     1$                      ;CONTINUE IF RESET
003614  032767  040000  174600                  BIT     #BIT14,@#SWREG          ; CHECK FOR LOOP ON ERROR
003622  001367                                  BNE     LP9                     ; GO TO LOOP ERROR
003624  012767  000022  174550                  MOV     #22,@#FATAL
003632  012767  000001  174540                  MOV     #1,@#MSGTY              ; MOVE ERROR NUM TO MAILBOX
003640  005767  174556                          TST     @#SWREG                 ; CHECK FOR HALT ON ERROR
003644  100401                                  BMI     1$                      ; HALT IF SET
003646  000000                                  HALT                    ;<BIT7 IN RCSR NOT CLEAR BY RESET>
003650                                  1$:
003650  032767  002000  174544                  BIT     #BIT10,@#SWREG          ; CHECK FOR LOOP ON TEST
003656  001346                                  BNE     TST10                   ; GO TO LOOP ON TEST
                                        ;
                                        ;TEST THAT RCSR BIT15 (DATA SET STATUS) IS CLEAR (BY RESET)
                                        ;AND CAN BE READ RELIABLY.
                                        ;
003660                                  TST11:
003660  012767  000011  174516                  MOV     #11,@#TESTN             ; MOVE TEST NUMBER TO MAILBOX
003666  000005                          LP10:   RESET                           ;CLEAR EVERYTHING
003670  032777  100000  174610                  BIT     #BIT15,@RCSR            ;CHECK IF BIT15 IS CLEAR
003676  001416                                  BEQ     1$                      ;CONTINUE IF CLEAR
003700  032767  040000  174514                  BIT     #BIT14,@#SWREG          ; CHECK FOR LOOP ON ERROR
003706  001367                                  BNE     LP10                    ; GO TO LOOP ERROR
003710  012767  000023  174464                  MOV     #23,@#FATAL
003716  012767  000001  174454                  MOV     #1,@#MSGTY              ; MOVE ERROR NUM TO MAILBOX
003724  005767  174472                          TST     @#SWREG                 ; CHECK FOR HALT ON ERROR
003730  100401                                  BMI     1$                      ; HALT IF SET
003732  000000                                  HALT                    ;<BIT15 IN RCSR NOT CLEAR BY RESET>
003734                                  1$:
003734  032767  002000  174460                  BIT     #BIT10,@#SWREG          ; CHECK FOR LOOP ON TEST
003742  001346                                  BNE     TST11                   ; GO TO LOOP ON TEST
```

```
                                        ;
                                        ;TEST THAT LOADING TBUF CLEARS TCSR BIT7 (READY) AND THAT
                                        ;BIT7 IS SET AFTER THE TRANSMITTER IS DONE.
                                        ;
003744                                  TST12:
003744  012767  000012  174432                  MOV     #12,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
003752  032767  000001  174440                  BIT     #BIT0,$ENV      ;CHECK IF ON APT
003760  001406                                  BEQ     NOAPT           ;IF NOT
003762  005767  174420                          TST     $PASS           ;CHECK IF IN FIRST PASS
003766  001403                                  BEQ     NOAPT           ;BR IF NOT FIRST PASS
003770  005767  174510                          TST     CTSTFL          ;IF NOT FIRST PASS - SKIP TEST
003774  001066                                  BNE     TST13           ;   IF CONSOLE IS UNDER TEST
003776                                  NOAPT:
003776  000005                          LP11:   RESET                   ;INITIALIZE (SET TCSR BIT7)
004000  005077  174510                          CLR     @TBUF           ;MOVE ZEROS TO TBUFFER
004004  032777  000200  174500                  BIT     #BIT7,@TCSR     ;CHECK IF BIT7 CLEAR.
004012  001424                                  BEQ     T30CN1          ;CONTINUE IF CLEAR
                                        ; DO IT ONE MORE TIME IN CASE REFRESH BOMBED THE FIRST TEST
004014  005077  174474                          CLR     @TBUF           ;DROP READY WITH THIS CHAR.
004020  032777  000200  174464                  BIT     #BIT7,@TCSR     ;CHECK FOR READY LOW
004026  001416                                  BEQ     T30CN1          ;THIS IS A REAL ERROR IF NOT 0
004030  032767  040000  174364                  BIT     #BIT14,$SWREG   ;CHECK FOR LOOP ON ERROR
004036  001357                                  BNE     LP11            ;GO TO LOOP ERROR
004040  012767  000024  174334                  MOV     #24,$FATAL
004046  012767  000001  174324                  MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
004054  005767  174342                          TST     $SWREG          ;CHECK FOR HALT ON ERROR
004060  100401                                  BMI     1$              ;HALT IF SET
004062  000000                                  HALT            ;<TCSR BIT DID NOT CLEAR AFTER DATA XMIT>
004064                                  1$:
004064  005001                          T30CN1: CLR     R1              ;CLEAR COUNT REG.
004066  052701  010000                          BIS     #10000,R1       ;SET COUNT
004072  005301                          DECR:   DEC     R1              ;DECREMENT COUNT
004074  001376                                  BNE     DECR            ;CONTINUE IF NOT 0
004076  032777  000200  174406                  BIT     #BIT7,@TCSR     ;CHECK IF READY IS BACK ON.
004104  001016                                  BNE     1$              ;CONTINUE IF SET.
004106  032767  040000  174306                  BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
004114  001330                                  BNE     LP11            ; GO TO LOOP ERROR
004116  012767  000025  174256                  MOV     #25,$FATAL
004124  012767  000001  174246                  MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
004132  005767  174264                          TST     $SWREG          ;CHECK FOR HALT ON ERROR
004136  100401                                  BMI     1$              ;HALT IF SET
004140  000000                                  HALT            ;<TCSR BIT7 DID NOT SET AFTER XMIT>
004142                                  1$:
004142  032767  002000  174252                  BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
004150  001275                                  BNE     TST12           ; GO TO LOOP ON TEST
                                        ;
                                        ;TEST THAT THE TRANSMITTER CAN INTERRUPT AT THE CORRECT VECTOR
                                        ;
004152                                  TST13:
004152  012767  000013  174224                  MOV     #13,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
004160  032767  000001  174232                  BIT     #BIT0,$ENV      ;CHECK IF ON APT
004166  001406                                  BEQ     TST13A          ;IF NOT ON APT
004170  005767  174212                          TST     $PASS           ;CHECK IF FIRST PASS
004174  001403                                  BEQ     TST13A          ;BR, IF NOT FIRST PASS
004176  005767  174302                          TST     CTSTFL          ;IF NOT FIRST PASS - SKIP TEST
```

```
.MAIN.  MACY11 27(732)  04-OCT-76  14:23  PAGE 23
DVKAEB.P11      ERROR   25      TCSR BIT7 DID NOT SET AFTER XMIT

  996  004202  001043                          BNE     TST14           ;   IF CONSOLE IS UNDER TEST
  997  004204                          TST13A:
  998  004204  042777  000100  174300  LP12:   BIC     #BIT6,@TCSR     ;DISABLE XMITTER INTERRUPT
  999  004212  012777  004304  174302          MOV     #T31CN1,@TVECT  ;SET VECTOR ADDRESS
 1000  004220  005077  174300                  CLR     @TPSW           ;SET PSW FOR INTERRUPT
 1001  004224  106427  000000                  MTPS    #0              ;ALLOW INTERRUPTS
 1002  004230  052777  000100  174254          BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
 1003  004236  000240                          NOP                     ;GIVE IT SOME TIME
 1004  004240  032767  040000  174154          BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
 1005  004246  001356                          BNE     LP12            ; GO TO LOOP ERROR
 1006  004250  012767  000026  174124          MOV     #26,$FATAL
 1007  004256  012767  000001  174114          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
 1008  004264  005767  174132                  TST     $SWREG          ; CHECK FOR HALT ON ERROR
 1009  004270  100401                          BMI     1$              ; HALT IF SET
 1010  004272  000000                          HALT            ;<NO INTERRUPT FROM DEVICE>
 1011  004274                          1$:
 1012  004274  032767  002000  174120          BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
 1013  004302  001323                          BNE     TST13           ; GO TO LOOP ON TEST
 1014  004304  042777  000100  174200  T31CN1: BIC     #BIT6,@TCSR     ;DISABLE INTERRUPT
 1015                                  ;
 1016                                  ;TEST THAT TRANSMITTER DOES NOT INTERRUPT WHEN PSW DISABLES
 1017                                  ;
 1018  004312                          TST14:
 1019  004312  012767  000014  174064          MOV     #14,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
 1020  004320  032767  000001  174072          BIT     #BIT0,$ENV      ; CHECK IF ON APT
 1021  004326  001406                          BEQ     TST14A          ; IF NOT ON APT
 1022  004330  005767  174052                  TST     $PASS           ; CHECK IF IN FIRST PASS
 1023  004334  001403                          BEQ     TST14A          ;BR, IF NOT FIRST PASS
 1024  004336  005767  174142                  TST     CTSTFL          ;IF NOT FIRST PASS - SKIP TEST
 1025  004342  001041                          BNE     TST15           ;   IF CONSOLE IS UNDER TEST
 1026  004344                          TST14A:
 1027  004344  042777  000100  174140  LP13:   BIC     #BIT6,@TCSR     ;CLEAR INT ENABLE
 1028  004352  106427  000200                  MTPS    #200            ;DISABLE INTERRUPTS
 1029  004356  012777  004402  174136          MOV     #2$,@TVECT      ;VECTOR POINT TO ERROR
 1030  004364  005077  174134                  CLR     @TPSW           ;CLEAR PSW FOR INTERRUPT
 1031  004370  052777  000100  174114          BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
 1032  004376  000240                          NOP                     ;GIVE IT SOME TIME
 1033  004400  000416                          BR      1$              ;CONTINUE IF NO INTERRUPT.
 1034  004402                          2$:
 1035  004402  032767  040000  174012          BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
 1036  004410  001355                          BNE     LP13            ; GO TO LOOP ERROR
 1037  004412  012767  000027  173762          MOV     #27,$FATAL
 1038  004420  012767  000001  173752          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
 1039  004426  005767  173770                  TST     $SWREG          ; CHECK FOR HALT ON ERROR
 1040  004432  100401                          BMI     1$              ; HALT IF SET
 1041  004434  000000                          HALT            ;<DEVICE GAVE INTERRUPT WITH PSW DISABLE>
 1042  004436                          1$:
 1043  004436  032767  002000  173756          BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
 1044  004444  001322                          BNE     TST14           ; GO TO LOOP ON TEST
 1045                                  ;
 1046                                  ;TEST THAT XMITTER DOES NOT RE-INTERRUPT AFTER THE FIRST
 1047                                  ;INTERRUPT IS SERVICED.
 1048                                  ;
 1049  004446                          TST15:
 1050  004446  012767  000015  173730          MOV     #15,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
 1051  004454  032767  000001  173736          BIT     #BIT0,$ENV      ; CHECK IF ON APT
```

```
.MAIN.  MACY11 27(732)  04-OCT-76  14:23  PAGE 24
DVKAEB.P11      ERROR   27      DEVICE GAVE INTERRUPT WITH PSW DISABLE

 1052  004462  001406                        BEQ     TST15A          ; IF NOT ON APT
 1053  004464  005767  173716                TST     $PASS           ; CHECK IF IN FIRST PASS
 1054  004470  001403                        BEQ     TST15A          ;BR, IF FIRST PASS
 1055  004472  005767  174006                TST     CTSTFL          ;IF NOT FIRST PASS - SKIP TEST
 1056  004476  001066                        BNE     TST16           ;    IF CONSOLE IS UNDER TEST
 1057  004500                        TST15A:
 1058  004500  042777  000100  174004 LP14:  BIC     #BIT6,@TCSR     ;CLEAR INTERRUPT DISABLE
 1059  004506  012777  004570  174006        MOV     #T33CN1,@TVECT  ;SET UP VECTOR ADDRESS
 1060  004514  005077  174004                CLR     @TPSW           ;PSW FOR VECTOR
 1061  004520  106427  000000                MTPS    #0              ;ALLOW INTERRUPTS
 1062  004524  052777  000100  173760        BIS     #BIT6,@TCSR     ;ENABLE INTERRUPT
 1063  004532  000240                        NOP                     ;TIME
 1064  004534  032767  040000  173660        BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
 1065  004542  001356                        BNE     LP14            ; GO TO LOOP ERROR
 1066  004544  012767  000030  173630        MOV     #30,$FATAL
 1067  004552  012767  000001  173620        MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
 1068  004560  005767  173636                TST     $SWREG          ; CHECK FOR HALT ON ERROR
 1069  004564  100401                        BMI     1$              ; HALT IF SET
 1070  004566  000000                        HALT            ;<DEVICE FAILED TO INTERRUPT>
 1071  004570                        1$:
 1072  004570  012777  004610  173724 T33CN1: MOV     #ERR26,@TVECT   ;SET UP VECTOR FOR RE-INTERRUPT
 1073  004576  005016                        CLR     (SP)
 1074  004600  012746  004654                MOV     #TST16,-(SP)    ;POINT TO NEXT TEST FOR RTI
 1075  004604  000002                        RTI                     ;SERVICE INTERRUPT
 1076  004606  000240                        NOP
 1077  004610                        ERR26:
 1078  004610  032767  040000  173604        BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
 1079  004616  001364                        BNE     T33CN1          ; GO TO LOOP ERROR
 1080  004620  012767  000031  173554        MOV     #31,$FATAL
 1081  004626  012767  000001  173544        MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
 1082  004634  005767  173562                TST     $SWREG          ; CHECK FOR HALT ON ERROR
 1083  004640  100401                        BMI     1$              ; HALT IF SET
 1084  004642  000000                        HALT            ;<DEVICE RE-INTERRUPTED AFTER FIRST INTERRUPT SERVICE>
 1085  004644                        1$:
 1086  004644  032767  002000  173550        BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
 1087  004652  001275                        BNE     TST15           ; GO TO LOOP ON TEST
 1088                                ;
 1089                                ; TEST TRANSMITTER AND RECIEVER BUFFERS.
 1090                                ; NOTE: SPECIAL WRAP MODULE MUST BE INSTALLED TO RUN
 1091                                ; THIS TEST.THIS TEST WILL RUN IF BIT9 OF SWREG =1.
 1092                                ;
 1093  004654                        TST16:
 1094  004654  012767  000016  173522        MOV     #16,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
 1095  004662  032767  000001  173530        BIT     #BIT0,$ENV      ; CHECK IF ON APT
 1096  004670  001406                        BEQ     NOAPT1          ; IF NOT
 1097  004672  005767  173510                TST     $PASS           ; CHECK IF IN FIRST PASS
 1098  004676  001403                        BEQ     NOAPT1          ;BR, IF FIRST PASS
 1099  004700  005767  173600                TST     CTSTFL          ;IF NOT FIRST PASS - SKIP TEST
 1100  004704  001147                        BNE     TST999          ;    IF CONSOLE IS UNDER TEST
 1101  004706                        NOAPT1:
 1102  004706  032767  001000  173506        BIT     #BIT9,$SWREG    ; CHECK IF THIS TEST IS ENABLED
 1103  004714  001537                        BEQ     TST17           ; SKIP IF NOT
 1104  004716  000005                LP15:   RESET           ; CLEAR EVERYTHING
 1105  004720  106427  000200                MTPS    #200            ; DISABLE INTERRUPTS
 1106  004724  012703  000020                MOV     #20,R3          ; SET SYNC COUNTER
 1107  004730  005777  173554                TST     @RBUF           ; CLEAR BUFFER
```

```
 1108  004734  005004                          CLR     R4              ; CLEAR DELAY COUNTER
 1109  004736  005204                  INCR4:  INC     R4              ; DELAY 300 MS
 1110  004740  001376                          BNE     INCR4
 1111  004742  052777  000011  173536          BIS     #11,@RCSR       ; SET READER ENABLE
 1112  004750  000240                          NOP
 1113  004752  000240                          NOP
 1114  004754  105777  173532          CHK1:   TSTB    @TCSR           ; CHECK FOR XMITTER DONE
 1115  004760  100375                          BPL     CHK1            ; WAIT TIL READY
 1116  004762  012777  000125  173524          MOV     #125,@TBUF      ; SEND DATA
 1117  004770  105777  173512          CHK2:   TSTB    @RCSR           ; WAIT FOR RCVR DONE
 1118  004774  100375                          BPL     CHK2
 1119  004776  122777  000125  173504  CHKSYN: CMPB    #125,@RBUF      ; CHECK FOR SYNC BYTE
 1120  005004  001426                          BEQ     T34CON          ; CONTINUE IF FOUND
 1121  005006  005303                          DEC     R3              ; DECREMENT SYNC COUNTER
 1122  005010  001367                          BNE     CHK2            ; CONTINUE TIL SYNC UP
 1123  005012  012777  000001  173466          MOV     #BIT0,@RCSR     ; TURN OFF WRAP MODULE
 1124  005020  000240                          NOP
 1125  005022  000240                          NOP
 1126  005024  000005                          RESET
 1127  005026  032767  040000  173366          BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
 1128  005034  001330                          BNE     LP15            ; GO TO LOOP ERROR
 1129  005036  012767  000032  173336          MOV     #32,$FATAL
 1130  005044  012767  000001  173326          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
 1131  005052  005767  173344                  TST     $SWREG          ; CHECK FOR HALT ON ERROR
 1132  005056  100401                          BMI     1$              ; HALT IF SET
 1133  005060  000000                          HALT                    ;<CANNOT GET TEST MODULE IN SYNC>
 1134  005062                          1$:
 1135  005062  005001                  T34CON: CLR     R1              ; INITIALIZE DATA REG
 1136  005064  012702  177777                  MOV     #-1,R2          ; INITIALIZE R2
 1137  005070  005777  173414                  TST     @RBUF           ; FORCE BUFFER CLEAR
 1138  005074  105777  173412          CHKT:   TSTB    @TCSR           ; CHECK XMITTER DONE
 1139  005100  100375                          BPL     CHKT            ; WAIT TIL DONE
 1140  005102  010177  173406          SDATA:  MOV     R1,@TBUF        ;SEND DATA
 1141  005106  105777  173374          CHKR:   TSTB    @RCSR           ; CHECK FOR RCVR DONE
 1142  005112  100375                          BPL     CHKR            ; WAIT TIL DONE
 1143  005114  017702  173370                  MOV     @RBUF,R2        ; GET WRAPPED DATA
 1144  005120  020102                          CMP     R1,R2           ; CHECK DATA
 1145  005122  001012                          BNE     2$
 1146  005124  005201                          INC     R1              ; INCREMENT DATA
 1147  005126  105701                          TSTB    R1              ; CHECK FOR END OF DATA
 1148  005130  001364                          BNE     SDATA           ; CONTINUE LOOP
 1149  005132  012777  000001  173346          MOV     #BIT0,@RCSR     ; TURN OFF LOOP
 1150  005140  000240                          NOP
 1151  005142  000240                          NOP
 1152  005144  000005                          RESET
 1153  005146  000422                          BR      TST17
 1154  005150                          2$:
 1155  005150  032767  040000  173244          BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
 1156  005156  001353                          BNE     CHKR            ; GO TO LOOP ERROR
 1157  005160  012767  000033  173214          MOV     #33,$FATAL
 1158  005166  012767  000001  173204          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
 1159  005174  005767  173222                  TST     $SWREG          ; CHECK FOR HALT ON ERROR
 1160  005200  100401                          BMI     1$              ; HALT IF SET
 1161  005202  000000                          HALT                    ;<DATA DID NOT COMPARE ON SLU WRAP>
 1162  005204                          1$:
 1163  005204  032767  002000  173210          BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
```

```
1164  005212  001220                          BNE     TST16           ; GO TO LOOP ON TEST
1165  005214                          TST17:
1166  005214  000005                          RESET
1167  005216  005004                          CLR     R4
1168  005220  005304                  T35DEC: DEC     R4              ;DEC TIMER
1169  005222  001376                          BNE     T35DEC
1170  005224                          TST999:
1171  005224  005267  173160                  INC     $DEVCT          ;INC DEVICE COUNTER
1172  005230  026767  173244  173152  T999A:  CMP     TMP2,$DEVCT     ;ALL DEVICES TESTED?
1173  005236  001404                          BEQ     T9998           ;IF YES
1174  005240  005067  173240                  CLR     CTSTFL          ;CLEAR CONSOLE UNDER TEST FLAG
1175  005244  000167  174360                  JMP     SETUP           ;GO TEST NEXT DEVICE
1176  005250  005237  000406          T9998:  INC     @#$PASS         ;INCREMENT PASS COUNT
1177  005254  004767  000046                  JSR     PC,MSGPRT       ;PRINT END OF PASS
1178  005260  005572                          .WORD   M3
1179  005262  013700  000042          ACT:    MOV     @#42,R0         ;CHECK ACT
1180  005266  001405                          BEQ     GOAGIN          ;KEEP GOING
1181  005270  000005                          RESET
1182  005272  004710                  $ENDAD: JSR     PC,(R0)         ;ACT HOOKS
1183  005274  000240                          NOP
1184  005276  000240                          NOP
1185  005300  000240                          NOP
1186  005302  005067  173102          GOAGIN: CLR     $DEVCT          ;RESET DEVICE COUNT
1187  005306  022767  000001  173164          CMP     #1,TMP2         ;IS THIS THE ONLY DEVICE UNDER TEST?
1188  005314  001026                          BNE     RSTRT           ;IF NO
1189
1190  005316  012706  001000                  MOV     #STKPTR,SP      ;INITIALIZE STACK POINTER
1191  005322  000167  174420                  JMP     TST1            ;GO DO ANOTHER PASS
1192  005326  132767  000040  173065  MSGPRT: BITB    #40,$ENVM       ;WILL APT ALLOW PRINTING?
1193  005334  001403                          BEQ     TYPE            ;YES
1194  005336  062716  000002                  ADD     #2,(SP)         ;ADJUST RETURN
1195  005342  000207                          RTS     PC              ;RETURN
1196  005344  011600                  TYPE:   MOV     (SP),R0         ;GET MESSAGE ADDRESS
1197  005346  011000                          MOV     (R0),R0         ;GET READY TO PRINT
1198  005350  105777  000030          WAIT:   TSTB    @TPS            ;CHECK IF TTY READY
1199  005354  100375                          BPL     WAIT            ;IF NOT
1200  005356  112077  000020                  MOVB    (R0)+,@TPB      ;PRINT THE CHARACTER
1201  005362  001372                          BNE     WAIT            ;NEXT IF NOT DONE
1202  005364  062716  000002                  ADD     #2,(SP)         ;ADJUST RETURN
1203  005370  000207                          RTS     PC              ;RETURN
1204  005372  005067  173014          RSTRT:  CLR     $UNIT
1205  005376  000167  173576                  JMP     START           ;START OVER
1206  005402  177566                  TPB:    .WORD   177566
1207  005404  177564                  TPS:    .WORD   177564
1208  005406  000000                  ENDTST: 0
1209  005410                          UINT:
1210  005410  012767  000034  172764          MOV     #34,$FATAL
1211  005416  012767  000001  172754          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
1212  005424  000000                          HALT                    ;<UNEXPECTED INTERRUPT>
1213  005426                          1$:
1214
1215  005426  012737  005446  000024  PWRFL:  MOV     #PWRUP,@#24     ;SET POWER UP VECTOR
1216  005434  012737  000340  000026          MOV     #340,@#26       ;SET POWER UP PSW
1217  005442  000000                          HALT
1218  005444  000776                          BR      .-2
1219
```

```
 1220
 1221  005446  012737  005426  000024  PWRUP:  MOV     #PWRFL,@#24     ;SET POWER FAIL VECTOR
 1222  005454  005037  000026          CLR     @#26            ;SET POWER UP PSW
 1223  005460  005000                  CLR     R0              ;CLEAR TTY WAIT TIMER
 1224  005462  005200                  INC     R0              ;INCREMENT TIMER
 1225  005464  001376                  BNE     TTYWT           ;  WAIT FOR TTY POWER UP
 1226  005466  004767  177634          JSR     PC,MSGPRT       ;TYPE "POWER" MESSAGE
 1227  005472  005610                  .WORD   M4
 1228  005474  000167  172520          JMP     PRCONT          ;START AT BEGINNING OF PRESENT PASS
 1229
 1230  005500  040515  047111  042504  M1:     .ASCIZ  .MAINDEC-11-DVKAE-B     DLVII TEST.<15><12>
 1231  005506  026503  030461  042055
 1232  005514  045526  042501  041055
 1233  005522  020040  042011  053114
 1234  005530  044511  052040  051505
 1235  005536  006524  000012
 1236  005542  020040  042040  053105  M2:     .ASCIZ  .   DEVICES UNDER TEST.<15><12>
 1237  005550  041511  051505  052440
 1238  005556  042116  051105  052040
 1239  005564  051505  006524  000012
 1240  005572  047105  020104  043117  M3:     .ASCIZ  .END OF PASS.<15><12>
 1241  005600  050040  051501  006523
 1242  005606  000012
 1243  005610  047520  042527  006522  M4:     .ASCIZ  .POWER.<15><12>
 1244  005616  000012
 1245          000001                  .END
```

B03


```
ABASE = 176500      369     410     442#
ACDW1 = 000000      369
ACDW2 = 000000      369
ACPUOP= 000000      369     384
ACT     00[illegible]     1179#
ADDWD = 000000      369
ADDW1 = 000000      369
ADDW10= 000000      369
ADDW11= 000000      369
ADDW12= 000000      369
ADDW13= 000000      369
ADDW14= 000000      369
ADDW15= 000000      369
ADDW2 = 000000      369
ADDW3 = 000000      369
ADDW4 = 000000      369
ADDW5 = 000000      369
ADDW6 = 000000      369
ADDW7 = 000000      369
ADDW8 = 000000      369
ADDW9 = 000000      369
ADEVCT= 000000      369     375
ADEVM = 000000      369     411
ADR     001712      608#    611
ADRTBL  000600      492#    531     606
AENV  = 000000      369     380
AENVM = 000000      369     381
AFATAL= 000000      369     372
AMADR1= 000000      369     397
AMADR2= 000000      369     401
AMADR3= 000000      369     404
AMADR4= 000000      369     407
AMAMS1= 000000      369     391
AMAMS2= 000000      369     399
AMAMS3= 000000      369     402
AMAMS4= 000000      369     405
AMSGAD= 000000      369     377
AMSGLG= 000000      369     378
AMSGTY= 000000      369     371
AMTYP1= 000000      369     392
AMTYP2= 000000      369     400
AMTYP3= 000000      369     403
AMTYP4= 000000      369     406
APASS = 000000      369     374
APRIOR= 000000      369
APTSIZ  001410      504     508     542#
ASWREG= 000000      369     382
ATESTN= 000000      369     373
AUNIT = 000000      369     376
AUSWR = 000000      369     383
AVECT1= 000200      369     408     441#
AVECT2= 000000      369     409
BIT0  = 000001      460#    501     584     682     710     720     730     731     741     742     755     757     775
                    847     946     991     1020    1051    1095    1123    1149
BIT1  = 000002      459#    583
BIT10 = 002000      450#    635     654     674     700     767     838     900     919     938     984     1012    1043
```

```
                    1086    1163
BIT11  = 004000      449#
BIT12  = 010000      448#
BIT13  = 020000      447#
BIT14  = 040000      446#    627     646     666     692     722     733     744     759     787     796     807     818
                     830     858     869     880     892     911     930     962     976    1004    1035    1064    1078
                    1127    1155
BIT15  = 100000      445#    928
BIT2   = 000004      458#
BIT3   = 000010      457#
BIT4   = 000020      456#    577
BIT5   = 000040      455#
BIT6   = 000100      454#    785     804     805     815     816     826     828     856     866     867     877     878
                     888     890     998    1002    1014    1027    1031    1058    1062
BIT7   = 000200      453#    503     956     960     974
BIT8   = 000400      452#    505
BIT9   = 001000      451#   1102
CHKR     005105     1141#   1142    1156
CHKSYN   004776     1119#
CHKT     005074     1138#   1139
CHK1     004754     1114#   1115
CHK2     004770     1117#   1118    1122
CSR    = 177560      444#    471     472     473     474
CTSTFL   000604      465#    589*    686     714     779     851     950     995    1024    1055    1099    1174*
DECR     004072      972#    973
DEVADR   001356      509     531#    550
DVBUF    000530      483#
DVCSR    000526      482#    522     588
DVPSW    000540      487#
DVVECT   000536      486#
DTBUF    000534      485#
DTCSR    000532      484#
DTPSW    000544      489#
DTVECT   000542      488#
DVXIT    001436      548     550#
ENDTST   005406     1208#
ERR12    000014      783     795#
ERR26    004610     1072    1077#
GOAGIN   005302     1180    1186#
INCR4    004766     1109#   1110
LP1      001754      623#    638
LP10     003466      927#    931
LP11     003776      954#    963     977
LP12     004204      998#   1005
LP13     004344     1027#   1036
LP14     004500     1058#   1065
LP15     004716     1104#   1128
LP2      002060      642#    647
LP3      002144      662#    667
LP4      002264      683     685     688#    693
LP5      002414      719#    723
LP5A     002406      711     713     716#
LP5B     002410      717#    718
LP6      002734      782#    788     797
LP8      003312      854#    859
LP9      003602      908#    912
```

D03

```
.MAIN.  MACY11 27(732)  04-OCT-76  14:23  PAGE 31
DVKAEB.P11      CROSS REFERENCE TABLE -- USER SYMBOLS

MANL      001232       502     505#
MSGPRT    005326       362     580     1177    1192#   1226
M1        005500       363     1230#
M2        005542       579*    581     1236#
M3        005572       1178    1240#
M4        005610       1227    1243#
N       = 000035       440#    629     630#    648     649#    668     669#    694     695#    724     725#    735     736#
                       746     747#    761     762#    789     790#    798     799#    809     810#    820     821#    832
                       833#    860     861#    871     872#    882     883#    894     895#    913     914#    932     933#
                       964     965#    978     979#    1006    1007#   1037    1038#   1066    1067#   1080    1081#   1129
                       1130#   1157    1158#   1210    1211#
NOAPT     003776       947     949     953#
NOAPT1    004706       1096    1098    1101#
PC      =%000007       362*    509*    539*    550*    580*    1177*   1182*   1195*   1203*   1226*
PRCONT    000220       358#    1228
PWRFL     005426       355     1215#   1221
PWRUP     005446       1215    1221#
RBUF      000510       472#    643     1107    1119    1137    1143
RCSR      000506       471#    587     607     624     856     866*    867     877*    878     888*    890     909     928
                       1111*   1117    1123*   1141    1149*
RPSW      000520       476#
RSTRT     005372       1188    1204#
RVECT     000516       475#
R0      =%000000       513*    514*    515*    516     520     522*    524     532*    533     535     536*    537     554*
                       555*    556     558*    563*    565*    566*    567*    569*    571*    573*    587*    590*    591
                       606*    608     609     610     613*    614     615     616     1179*   1182    1196*   1197*   1200
                       1223*   1224*
R1      =%000001       533*    534*    537     553*    557*    559     564*    570*    572*    574*    575*    576*    577
                       579     588*    590     607*    608*    614*    970*    971*    972*    1135*   1140    1144    1146*
                       1147
R2      =%000002       531*    535*    543*    544     547*    552*    556*    603*    606     613     1136*   1143*   1144
R3      =%000003       1106*   1121*
R4      =%000004       716*    717*    752*    753*    1108*   1109*   1167*   1168*
SDATA     005102       1140#   1148
SETADR    001660       596     601#
SETTBL    001246       506     509#
SETUP     001630       586     595#    600     1175
SHIFT     001512       569#    578
SIZED     001554       500     582#
SP      =%000006       495*    1073*   1074*   1190*   1194*   1196    1202*
START     001200       364     495#    1205
STKPTR=   001000       443#    495     1190
TBUF      000514       474#    689     955*    959*    1116*   1140*
TCONSL    001600       585     587#
TCSR      000512       473#    663     720     730*    731     741*    742     755*    757     785     804*    805     815*
                       816     826*    828     956     960     974     998*    1002*   1014*   1027*   1031*   1058*   1062*
                       1114    1138
TMP0      000474       461#
TMP1      000476       462#    582*    598*    603     604#
TMP2      000500       463#    510*    518*    526*    542*    546*    563     1172    1187
TMP3      000502       464#    583*    595     597*    602*
TPB       005402       1200*   1206#
TPS       005404       1198    1207#
TPSW      000524       478#    591     1000*   1030*   1060*
TSTDVM    001420       544#    549
TSTTY     001566       584#
```

DVKAEB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
TST1    001746          593     618     621#    636     1191
TST10   003574          852     906#    920
TST11   003660          925#    939
TST12   003744          944#    965
TST13   004152          951     989#    1013
TST13A  004204          992     994     997#
TST14   004312          996     1018#   1044
TST14A  004344         1021    1023     1026#
TST15   004446         1025    1049#    1087
TST15A  004500         1052    1054     1057#
TST16   004654         1056    1074     1093#   1164
TST17   005214         1103    1153     1165#
TST2    002042          640#    655
TST3    002136          660#    675
TST4    002262          680#    701
TST5    002354          687     708#     768
TST6    002702          715     773#     839
TST6A   002734          776     778      781#
TST7    003260          780     845#     901
TST7A   003312          848     850      853#
TST999  005524         1100    1170#
TTYMT   005642         1224#   1225
TVECT   000532          477#    783*     999*    1029*   1059*   1072*
TYPE    005744         1193    1196#
T30CN1  004064          957     961      970#
T31CN1  004304          999    1014#
T33CN1  004570         1059    1072#    1079
T3HCON  005062         1120    1135#
T3SDEC  005220         1168#   1169
T5CON   002462          721     730#     734
T5CON1  002634          732     741#     745
T5CON2  003006          743     752#     760
T5DEC   003212          753#    754
T6CON1  003050          786     804#     808
T6CON2  003122          806     815#     819
T6CON3  003174          817     826#     831
T8CON1  003644          857     866#     870
T8CON2  003736          868     877#     881
T8CON3  004010          879     888#     893
T9998   005230         1172#
T9998   005250         1173    1176#
UINT    005410         1209#
VCTADR  001442          527     552#
VCTTBL  000640          493#    555      613
VECT    001730          614#    617
WAIT    005350         1198#   1199     1201
X     = 000017          439#    622      623#    641     642#    661     662#    681     682#    709     710#    774     775#
                        846     847#     907     908#    926     927#    945     946#    990     991#    1019    1020#   1050
                       1051#   1094     1095#
SAPTHD  000460          423     429#
SBASE   000454          410#    513      520     532
SCPUOP  000426          384#
SDEVCT  000410          360*    375#    1171*   1172    1186*
SDEVM   000456          411#    511*     517*    519*    525*    543     584     595
SENDAD  005272         1182#
SENV    000420          380#    501      682     710     775     847     946     991     1020    1051    1095
```

```
SENVM   000421          381#    503     1192
SETABL  000420          379#
SETEND  000460          412#    435
SFATAL  000402          358#    372#    629#    648#    668#    694#    724#    735#    746#    761#    789#    798#    809#
                        820#    832#    860#    871#    882#    894#    913#    932#    964#    978#   1006#   1037#   1066#
                       1080#   1129#   1157#   1210#
SHIBTS  000460          430#
SHNDR1  000432          397#
SHNDR2  000436          401#
SHNDR3  000442          404#
SHNDR4  000446          407#
SHNIL   000400          370#    431     435
SHNRS1  000430          391#
SHNRS2  000434          399#
SHNRS3  000440          402#
SHNRS4  000444          405#
SHNSOR  000462          431#
SHSGNO  000414          377#
SHSGLG  000416          378#
SHSGTY  000400          371#    630#    649#    669#    695#    725#    736#    747#    762#    790#    799#    810#    821#
                        833#    861#    872#    883#    895#    914#    933#    965#    979#   1007#   1038#   1067#   1081#
                       1130#   1158#   1211#
SHTYP1  000431          392#
SHTYP2  000435          400#
SHTYP3  000441          403#
SHTYP4  000445          406#
SPASS   000406          357#    374#    499     684     712     777     849     948     993    1022    1053    1097    1176#
SPASTM  000466          433#
SSWREG  000422          382#    505     627     631     635     646     650     654     666     670     674     692     696
                        700     722     726     733     737     744     748     759     763     767     787     791     796
                        800     807     811     818     822     830     834     838     858     862     869     873     880
                        884     892     896     900     911     915     919     930     934     938     962     966     976
                        980     984    1004    1008    1012    1035    1039    1043    1064    1068    1078    1082    1086
                       1102    1127    1131    1155    1159    1163
STESTN  000404          359#    373#    622#    641#    661#    681#    709#    774#    846#    907#    926#    945#    990#
                       1019#   1050#   1094#
STSTN   000464          432#
SUNIT   000412          361#    376#    599#    601#   1204#
SUNITM  000470          434#
SUSWR   000424          383#
SVECT1  000450          408#    554
SVECT2  000452          409#
.       = 005620        319#    320     322     324     326     328     330#    331     333     335     337#    338     340
                        342     344     346     348     351#    354#    365#    419     420#    422#    424#    491#    492#
                        493#    494#   1218
.SX     = 000460        419#    424
```

```
COMMEN      1#
ENDCOM      1#
ERNUM     467#    629     648     668     694     724     735     746     761     789     798     809     820     832     860
          871     882     894     913     932     964     978    1006    1037    1066    1080    1129    1157    1210
ERR       467#    627     646     666     692     722     733     744     759     787     795     807     818     830     858
          869     880     892     911     930     962     976    1004    1034    1064    1077    1127    1154    1209
ESCAPE      1#
GETPRI      1#
GETSWR      1#
MULT        1#
NEWTST      1#
POP         1#
PUSH        1#
REPORT      1#
SEQ       468#    622     641     661     681     708     774     846     907     926     945     990    1019    1050    1094
SETPRI      1#
SETUP       1#
SKIP        1#
SLASH       1#
STARS       1#    350#    368     416     418     425
SWRSU       1#
TYPBIN      1#
TYPDEC      1#
TYPHWH      1#
TYPNUM      1#
TYPOCS      1#
TYPOCT      1#
TYPTXT      1#
STNUM     468#    622     641     661     681     709     774     846     907     926     945     990    1019    1050    1094
$$ESCA      1#
$$RENT      1#
$$SKIP      1#
.EQUAT      1#
.HEADE      1#
.KT11       1#
.SETUP      1#
.SWHI       1#
.$ACTI      1#
.$APTB      1#    350#    366
.$APTH      1#    350#    414
.$APTY      1#    350#
.$ASTA      1#
.$CATC      1#
.$CHTA      1#
.$D$20      1#
.$D820      1#
.$DIV       1#
.$EOP       1#
.$ERRO      1#
.$ERRT      1#
.$MLLT      1#
.$POWE      1#
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#
```

```
.$R2AZ          1#
.$SAVE          1#
.$SR2D          1#
.$SR2O          1#
.$SCOP          1#
.$SIZE          1#
.$SUPR          1#
.$TRAP          1#      350#
.$TYPB          1#
.$TYPD          1#
.$TYPE          1#      350#
.$TYPO          1#
.$HOCV          1#
.1170           1#
```

DVKAEB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
ADD     514   534   536   558   575   598   604  1194  1202
ASL     519   547   566   567   569   571   573   597   602
BEQ     502   506   548   578   683   685   711   713   721   743   758   776   778   786   817
        824   848   850   857   879   891   910   929   947   949   957   961   992   994  1021
       1023  1052  1054  1096  1098  1103  1120  1173  1180  1193
BGE     560   592
BIC     555   741   815   877   998  1014  1027  1058
BIS     730   755   804   826   866   888   971  1002  1031  1062  1111
BIT     505   577   584   595   610   616   627   635   646   654   666   674   682   692   700
        710   720   722   731   733   742   744   757   759   767   775   785   787   796   805
        807   816   818   828   830   838   847   856   858   867   869   878   880   890   892
        900   909   911   919   928   930   938   946   956   960   962   974   976   984   991
       1004  1012  1020  1035  1043  1051  1064  1078  1086  1095  1102  1127  1155  1163
BITB    501   503  1192
BLT     521   538
BMI     632   651   671   697   727   738   749   764   792   801   812   823   835   863   874
        885   897   916   935   967   981  1009  1040  1069  1083  1132  1160
BNE     500   504   585   596   611   617   628   636   647   655   667   675   687   693   701
        715   718   723   732   734   745   754   760   768   780   788   797   806   808   819
        831   839   852   859   868   870   881   893   901   912   920   931   939   951   963
        973   975   977   985   996  1005  1013  1025  1036  1044  1056  1065  1079  1087  1100
       1110  1122  1128  1145  1148  1156  1164  1169  1188  1201  1225
BPL     545  1115  1118  1139  1142  1199
BR      508   527   549   586   600   625   644   664   690  1033  1153  1218
CLR     357   358   359   360   361   510   511   542   553   564   582   716   955   959   970
       1000  1030  1060  1073  1108  1135  1167  1174  1186  1204  1222  1223
CMP     520   537   559   591  1144  1172  1187
CMPB   1119
DEC     753   972  1121  1168
HALT    507   633   652   672   698   728   739   750   765   793   802   813   824   836   864
        875   886   908   917   936   968   982  1010  1041  1070  1084  1133  1161  1212  1217
INC     517   518   525   526   546   557   589   599   601   717  1109  1146  1171  1176  1224
JMP     364   593   618  1175  1191  1205  1220
JSR     362   509   550   580  1177  1182  1226
MOV     355   356   496   512   513   522   523   531   532   533   535   543   552   556   563
        578   583   587   588   590   603   606   607   608   613   614   622   623   629   630
        637   641   642   648   649   656   661   662   668   669   676   681   688   694   695
        702   709   724   725   736   746   747   752   761   762   774   783   789   790
        798   799   809   810   821   832   833   846   860   861   871   872   882   883
        894   895   907   913   914   926   932   933   945   964   965   978   979   990   999
       1006  1007  1019  1029  1037  1038  1050  1059  1066  1067  1072  1074  1080  1081  1094
       1106  1116  1123  1129  1130  1136  1140  1143  1149  1157  1158  1179  1190  1196  1197
       1210  1211  1215  1216  1221
MOVB    554  1200
MTPS    784   854  1001  1028  1061  1105
NOP     703  1003  1032  1063  1076  1112  1113  1124  1125  1150  1151  1183  1184  1185
RESET   719   756   782   827   855   889   908   927   954  1104  1126  1152  1166  1181
ROLB    570   572   574
RTI     353  1075
RTS     539  1195  1203
SUB     515
SWAB    565   576
TST     499   516   524   544   609   615   624   631   643   650   663   670   684   686   689
        696   712   714   726   737   748   763   777   791   800   811   822   834   849
        851   862   873   884   896   915   934   948   950   966   980   993   995  1008  1022
       1024  1039  1053  1055  1068  1082  1097  1099  1107  1131  1137  1159
```

DVKAEB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
TSTB    1114  1117  1138  1141  1147  1198
.ASCIZ  1230  1236  1240  1243
.BLKW    492  493
.BYTE    360  381  391  392  399  400  402  403  405  406
.ENABL     1  316
.END    1245
.ENDC    369  391  399  402  405  408  409  410  411  414  417  419  426  629  633
         637  648  652  656  668  672  676  694  698  702  724  728  730  735  739
         741  746  750  752  761  765  769  789  793  795  798  802  804  809  813
         815  820  824  826  832  836  840  860  864  866  871  875  877  882  886
         888  894  898  902  913  917  921  932  936  940  964  968  970  978  982
         986 1006 1010 1014 1037 1041 1045 1066 1070 1072 1080 1084 1088 1129 1133
        1135 1157 1161 1165 1210 1212 1214
.EVEN    369
.IF      368  391  399  402  405  408  409  410  411  412  414  416  418  425  627
         631  635  646  650  654  666  670  674  692  696  700  722  726  730  733
         737  741  744  748  752  759  763  767  787  791  795  796  800  804  807
         811  815  818  822  826  830  834  838  858  862  866  869  873  877  880
         884  888  892  896  900  911  915  919  930  934  938  962  966  970  976
         980  984 1004 1008 1012 1035 1039 1043 1064 1068 1072 1078 1082 1086 1127
        1131 1135 1155 1159 1163 1210 1212 1214
.IFF     369  417  419  426
.IIF     369
.LIST      1  318  369  623  629  630  642  648  649  662  668  669  682  694  695
         710  724  725  735  736  746  747  761  762  775  789  790  798  799  809
         810  820  821  832  833  847  860  861  871  872  882  883  894  895  908
         913  914  927  932  933  946  964  965  978  979  991 1006 1007 1020 1037
        1038 1051 1066 1067 1080 1081 1095 1129 1130 1157 1158 1210 1211
.MACRO     1  467  468
.MCALL   350
.MEXIT   413
.NLIST     1  317  369  623  629  630  642  648  649  662  668  669  682  694  695
         710  724  725  735  736  746  747  761  762  775  789  790  798  799  809
         810  820  821  832  833  847  860  861  871  872  882  883  894  895  908
         913  914  927  932  933  946  964  965  978  979  991 1006 1007 1020 1037
        1038 1051 1066 1067 1080 1081 1095 1129 1130 1157 1158 1210 1211
.PAGE    414  461  469  494  769  841  940
.REM       1
.REPT    340
.SBTTL   366  414  629  648  668  694  724  735  746  761  789  798  809  820  832
         860  871  882  894  913  932  964  978 1006 1037 1066 1080 1129 1157 1210
.WORD    363  371  372  373  374  375  376  377  378  382  383  384  397  401  404
         407  408  409  410  411  430  431  432  433  434  435  461  462  463  464
         465  581 1178 1206 1207 1227
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

```
*,DVKAEB.SEQ/SOL/CRF/PAGNUM/NL:TOC=SYSMAC.CO,DVKAEB.P11
RUN-TIME:  24 28 2 SECONDS
RUN-TIME RATIO: 217/56=3.8
CORE USED:  33K  (65 PAGES)
```