IDENTIFICATION

PRODUCT CODE: MAINDEC-II-DVXAE-B-D

PRODUCT NAME: DLVII TEST

DATE: OCTOBER 1976

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: R.D. FIorentino

REVISION B: D.J. CARLETO

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 DLVII SERIAL LINE UNIT FOR THE LSI-II COMPUTER. THIS TEST WILL OPERATE ON THE DLVII 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 DLVII ADDRESSED AT 177560, AND UP TO 8 ADDITIONAL
DLVII'S WITH SPECIFIED ADDRESSES.

DEFAULT ADDRESSES FOR MULTIPLE DEVICE TESTING ARE:

17750   CONSOLE
179500  BASE ADDRESS FOR ADDITIONAL DLVII'S

FOR COMPLETE INSTRUCTIONS ON MULTIPLE DEVICE TESTING SEE
SECTION 5.5

2. REQUIREMENTS

2.1 EQUIPMENT

LSI-11 COMPUTER
DLVII SERIAL LINE UNIT
TERMINAL FOR DLVII
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 APPROPRIATE BIT IN THE SOFTWARE SWITCH REGISTER AT LOCATION 129. (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
RESET 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

<table>
<thead>
<tr>
<th>BIT15</th>
<th>CONTINUE ON ERROR</th>
<th>(100000)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIT14</td>
<td>LOOP ON CURRENT ERROR</td>
<td>(000000)</td>
</tr>
<tr>
<td>BIT13</td>
<td>NOT USED</td>
<td>(100000)</td>
</tr>
<tr>
<td>BIT12</td>
<td>NOT USED</td>
<td>(100000)</td>
</tr>
<tr>
<td>BIT11</td>
<td>NOT USED</td>
<td>(100000)</td>
</tr>
<tr>
<td>BIT10</td>
<td>LOOP ON CURRENT TEST</td>
<td>(100000)</td>
</tr>
<tr>
<td>BIT9</td>
<td>RUN MAPI TEST</td>
<td>(000000)</td>
</tr>
<tr>
<td>BIT8</td>
<td>SET DEVICE MAPI MANUALLY</td>
<td>(000000)</td>
</tr>
<tr>
<td>BIT7</td>
<td>NOT USED</td>
<td>(000000)</td>
</tr>
<tr>
<td>BIT6</td>
<td>NOT USED</td>
<td>(000000)</td>
</tr>
<tr>
<td>BIT5</td>
<td>NOT USED</td>
<td>(000000)</td>
</tr>
<tr>
<td>BIT4</td>
<td>NOT USED</td>
<td>(000000)</td>
</tr>
<tr>
<td>BIT3</td>
<td>NOT USED</td>
<td>(000000)</td>
</tr>
<tr>
<td>BIT2</td>
<td>NOT USED</td>
<td>(000000)</td>
</tr>
<tr>
<td>BIT1</td>
<td>NOT USED</td>
<td>(000000)</td>
</tr>
<tr>
<td>BIT0</td>
<td>NOT USED</td>
<td>(000000)</td>
</tr>
</tbody>
</table>

5.2 SLU CONFIGURATION REQUIREMENTS

<table>
<thead>
<tr>
<th>FOR BAUD RATE</th>
<th>#STOP BITS</th>
<th>#BITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>ALL OTHERS</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

No other configuration is tested by this program. EIA is tested with the MAPI module option test.

5.3 OPERATION UNDER AFT

This program will operate under AFT without modification to the SWR. However, in order to test the full capability of the DTVIL, a test with the test module is necessary. To do this, AFT 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 DLVII 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 LABELLED "ACSR''. THE ADDRESSES MUST BE IN THE FOLLOWING ORDER:

ACSR: ADDRESS OF RECEPTOR CSR
ABUF: ADDRESS OF RECEPTOR BUFFER
TCSR: ADDRESS OF TRANSMITTER CSR
TRBUF: ADDRESS OF TRANSMITTER BUFFER
RCVECT: ADDRESS OF RECEPTOR VECTOR
RCPSW: ADDRESS OF ASSOCIATED PSW
TCVECT: ADDRESS OF TRANSMITTER VECTOR
TCPSW: ADDRESS OF ASSOCIATED PSW

2. TO TEST A SINGLE DLVII AT A UNIQUE ADDRESS, INSERT THE ADDRESS OF THE RECEPTOR CSR IN LOCATION SBASE, AND THE ADDRESS OF THE VECTOR IN LOCATION SVECTI, IN THE ETABLE. THE PROGRAM WILL GENERATE THE ADDITIONAL CSR, BUFFER, AND VECTOR ADDRESSES NEEDED FOR TESTING.

5.5 TESTING MULTIPLE DLVII MODULES

ADDITIONAL DLVII'S MUST BE ADDRESSED WITHIN A RANGE OF 8 SEQUENTIAL ADDRESSES. A BASE ADDRESS MAY BE SPECIFIED BY THE OPERATOR OR A DEFAULT ADDRESS OF 177560 AND DEFAULT VECTOR OF 60 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 RECEPTOR 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:

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

DEVICE 60 = CONSOLE DEFAULT ADDRESS = 177560; VECTOR = 60
DEVICES 61-69 = ADDITIONAL DLVII'S
DEVICE 61 = BASE ADDRESS (SBASE); BASE VECTOR (SVECT1)
DEVICE 62 = BASE ADDRESS + 10; BASE VECTOR + 10
SETTING THE APPROPRIATE BITS IN THE DEVICE MAP
(BELOW LOC 456) WILL DESIGNATE THOSE DEVICES WHICH
ARE TO BE TESTED. THE PROGRAM IS STARTED AS USUAL.
BUT WILL ISSUE 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-II, 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 506, the unit number can also be used to
locate the defective device. 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

SIMAIL: START OF THE PROGRAM MAILBOX. MANY CLUES TO
        PROBLEMS CAN BE FOUND HERE

SFATAL: 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

X

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

;***********************************************************************
;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
;***********************************************************************

;SCL: : SAVE CURRENT LOCATION
;24: : SET POWER FAIL TO POINT TO START OF PROGRAM
;200: : FOR APT START UP
;44: : POINT TO APT INDIRECT ADDRESS PMTR.
;SAPTHDR: : POINT TO APT HEADER BLOCK
;= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

;***********************************************************************
;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDPII DIAGNOSTIC
;INTERFACE SPEC.
;***********************************************************************

SAPTHID: .WORD 0 ; TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
SAPBDR: .WORD 0 ; ADDRESS OF APT MAILBOX (BITS 0-15)
STSTH: .WORD 10 ; RUN TIM OF LONGEST TEST
SPASTH: .WORD 10 ; RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
SUNITH: .WORD 0 ; ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT

;EQUATES

X=1

;DEFAULT BASE VECTOR FOR ADDITIONAL UNITS
BASE=176500
;DEFAULT BASE DEVICE ADDRESS FOR ADDITIONAL UNITS
STXPTR=1000
;DEFAULT CONSOLE DEVICE ADDRESS
BIT0=1
TMP0:  .WORD 0  ;LOCATION TO STORE TABLE OFFSETS
TMP1:  .WORD 0  ;LOCATION TO STORE NO. OF TEST DEVICES
TMP2:  .WORD 0  ;LOCATION TO KEEP DEVICE MAP TEST MASK
TMP3:  .WORD 0  ;LOCATION TO KEEP DEVICE MAP TEST MASK
CTSTFL:  .WORD 0  ;FLAG TO INDICATE CURRENT TEST DEVICE IS CONSOLE
REGISTER AND VECTOR ADDRESSES FOR THE DLVII UNDER TEST

RCSR: 069340
RSBF: 086954
TCSR: 066968
TBUF: 0C6946
RVECT: 553540
RPSN: 553534
TVECT: 553533
TPSH: 553

INITIAL REGISTER AND VECTOR ADDRESSES FOR THE CONSOLE DLVII

RCSR: 177560  ADDRESS OF RECEIVER STATUS REGISTER
RSBF: 177562  ADDRESS OF RECEIVER DATA BUFFER REGISTER
TCSR: 177564  ADDRESS OF TRANSMITTER STATUS REGISTER
TBUF: 177566  ADDRESS OF TRANSMITTER DATA BUFFER REGISTER
RVECT: 653534
RPSN: 653534
TVECT: 653530
TPPSH: 656

AORTBL: .BLO1 20
VOUTBL: .BLO1 10
DO2

LPS:  

BIT 0, ATCSR

CONTINUE IF CLEAR

BIT 10, ATCSR

CHECK BIT 0

BIT 11, SSREG

CHECK FOR LOOP ON ERROR

BIT 12, LPS

GO TO LOOP ERROR

BIT 13, BS

HALT IF SET

; BIT 0 IN TCSR SHOULD BE 0 AFTER RESET

BITCON:  

BIT 0, ATCSR

SET Bit 0 (BREAK)

BIT 1, TSREG

CONTINUE IF SET

BIT 11, SSREG

CHECK FOR LOOP ON ERROR

BIT 12, LPS

GO TO LOOP ERROR

BIT 13, BS

HALT IF SET

; CANNOT SET BIT 0 IN TCSR

BITCON1:  

BIT 0, ATCSR

CLEAR BIT 0

BIT 1, TSREG

CONTINUE IF CLEAR

BIT 11, SSREG

CHECK FOR LOOP ON ERROR

BIT 12, LPS

GO TO LOOP ERROR

BIT 13, BS

HALT IF SET

; CANNOT CLEAR BIT 0 IN TCSR

TSENC:  

MOV 10000, AN

; SET UP DELAY

; DELAY

TSEC:  

BIT 0, ATCSR

SET IT AGAIN

BIT 10, ATCSR

CHECK BIT 0 CLEAR AGAIN

BIT 11, SSREG

CONTINUE IF RESET

BIT 12, LPS

GO TO LOOP ERROR

BIT 13, BS

HALT IF SET

; RESET DID NOT CLEAR BIT 0 IN TCSR

TST:  

BIT 0, SSREG

CHECK FOR LOOP ON TEST

BRE TST
TEST THAT TCSR BIT 6 (TRANSMITTER INTERRUPT ENABLE) CAN BE
SET AND RESET AND THAT RESET Clears IT.

; TST6:
01234  00000  175474
01234  00001  175502
01234  175462
01234  175622
01234  00112

; TST6A:
01234  000014  175556
01234  000020  175534
01234  000000  175434
01234  000011  175412

; LP6:
01234  000014  175536
01234  000000  175404
01234  000012  175310
01234  000001  175308
01234  175300

; B12:
01234  001101  175400
01234  000000  175325
01234  000011  175282
01234  000000  175200
01234  175304

; B13:
01234  000010  175314
01234  000011  175282
01234  000000  175200
01234  175304

; TSYNC:
01234  000010  175314
01234  000011  175282
01234  000000  175200
01234  175304

; CLEAR:
01234  000010  175314
01234  000011  175282
01234  000000  175200
01234  175304

; (CANNOT Clear BIT 6 IN TCSR)
CANNOT CLEAR BIT6 IN TCSR

000000 000100 175310 BIS #BIT6, #TCSR
    SET BIT6 AGAIN

000000 000100 175300 BIT #BIT6, #TCSR
    TRY RESET AGAIN

001416 000000 175300 BEQ IS
    CHECK TO SEE IF CLEAR

032767 040000 175200 BIT #BIT14, #SSWREG
    CONTINUE IF CLEAR

001364 000000 #TCSR
    GO TO LOOP ERROR

012743 000000 175102 MOV #ISR, #PRST
    CHECK FOR LOOP ON ERROR

012743 000000 175101 MOV #ISR, #SSWREG
    GO TO LOOP ERROR

012743 175100 TST #SSWREG
    CHECK FOR HALT ON ERROR

012743 175100 HLT IS
    HALT IF SET

000000 000000 175100 ;<RESET DID NOT CLEAR BIT6 IN TCSR>

000000 002000 175144 BNE #BIT10, #SSWREG
    CHECK FOR LOOP ON TEST

001211 000100 175144 BNE #TST16
    GO TO LOOP ON TEST
TEST THAT RCSR BIT6 (RCVR INTERRUPT ENABLE) CAN BE SET AND CLEARED, AND THAT RESET CLEARS IT.

TST7:

```
012767 000000 175116  NOV  #7,STESTM    ; MOVE TEST NUMBER TO MAILBOX
               BIT #B10,SENW    ; CHECK IF ON ANT
               BIT #TST70      ; IF NOT ON ANT
               TST #TST7A     ; CHECK IF IN FIRST PASS
               TST TST10      ; BR, IF NOT FIRST PASS
               TST CTSTFL     ; IF NOT FIRST PASS - SKIP TEST
               BNE TST10      ; IF CONSOLE IS UNDER TEST
```

TST7A:

```
002200  LPH: #2000    ; DISABLE INTERRUPTS
               LPS    ; CLEAR EVERYTHING
               BIT #B16,RCSR    ; CHECK BIT6 FOR 0
               SET TCON1      ; CONTINUE IF RESET
               BIT #B14,SSREG  ; CHECK FOR LOOP ON ERROR
               LPH                ; GO TO LOOP ERROR
               BNE FATAL      ; MOVE ERROR NUM TO MAILBOX
               TST SSREG    ; CHECK FOR HALT ON ERROR
               IS              ; HALT IF SET
               BNE FATAL      ; (BIT6 IN RCSR NOT CLEAR BY RESET)
```

IS:

```
022777  TCON1: BIT #B15,RCSR    ; SET BIT6
               BIT #B16,RCSR    ; CHECK IF BIT SET
               SET TCON2             ; CONTINUE IF SET
               BIT #B14,SSREG  ; CHECK FOR LOOP ON ERROR
               LPH                ; GO TO LOOP ERROR
               BNE FATAL      ; MOVE ERROR NUM TO MAILBOX
               TST SSREG    ; CHECK FOR HALT ON ERROR
               IS              ; HALT IF SET
               BNE FATAL      ; (CANNOT SET BIT6 IN RCSR)
```

IS:

```
022777  TCON2: BIT #B15,RCSR    ; CLEAR BIT6
               BIT #B16,RCSR    ; CHECK IF CLEAR
               SET TCON3             ; CONTINUE IF RESET
               BIT #B14,SSREG  ; CHECK FOR LOOP ON ERROR
               LPH                ; GO TO LOOP ERROR
               BNE FATAL      ; MOVE ERROR NUM TO MAILBOX
               TST SSREG    ; CHECK FOR HALT ON ERROR
               IS              ; HALT IF SET
               BNE FATAL      ; (CANNOT CLEAR BIT6 IN RCSR)
```

IS:

```
022777  TCON3: BIT #B16,RCSR    ; SET BIT6 AGAIN
               SET TCON3             ; ISSUE ANOTHER RESET
               BIT #B16,RCSR    ; CHECK IF RESET CLEARED BIT6.
               IS              ; CONTINUE IF CLEAR
               SET TCON3             ; CONTINUE IF CLEAR
               BIT #B14,SSREG  ; CHECK FOR LOOP ON ERROR
               LPH                ; GO TO LOOP ERROR
               BNE FATAL      ; MOVE ERROR NUM TO MAILBOX
               TST SSREG    ; CHECK FOR HALT ON ERROR
```
MAIN MACY11 27(732) 04-OCT-76 14:23 PAGE 21

H02

ERROR 21 SECOND RESET DID NOT CLEAR BIT 6

BMT IS ; HALT IF SET

IS: ;<SECOND RESET DID NOT CLEAR BIT6>

IS:

BIT #BIT10, $SREG ; CHECK FOR LOOP ON TEST
BNE TST7 ; GO TO LOOP ON TEST

; TEST THAT RCSR BIT 7 (RCVR DONE) IS CLEAR (BY RESET) AND CAN BE READ RELIABLY.

TST10:

MOV #10, $TESTN ; MOVE TEST NUMBER TO MAILBOX

LP9:

BIT #BIT, $ACSR ; CLEAR EVERYTHING

BIT #BIT, $ACSR ; CHECK IF BIT 7 CLEAR.
BNE LP9 ; CONTINUE IF RESET

IS: ; CONTINUE IF RESET

IS:

BIT #BIT14, $SREG ; CHECK FOR LOOP ON ERROR
BNE LP9 ; GO TO LOOP ERROR

MOV #20, $FATAL ; MOVE ERROR NUM TO MAILBOX

BNE LP9 ; GO TO LOOP ERROR

TST $SREG ; CHECK FOR HLT ON ERROR
BMT IS ; HALT IF SET

IS: ;<BIT7 IN RCSR NOT CLEAR BY RESET>

IS:

BIT #BIT10, $SREG ; CHECK FOR LOOP ON TEST
BNE TST10 ; GO TO LOOP ON TEST

; TEST THAT RCSR BIT 15 (DATA SET STATUS) IS CLEAR (BY RESET) AND CAN BE READ RELIABLY.

TST11:

MOV #11, $TESTN ; MOVE TEST NUMBER TO MAILBOX

LP10:

BIT #BIT15, $ACSR ; CLEAR EVERYTHING

BIT #BIT15, $ACSR ; CHECK IF BIT 15 IS CLEAR
BNE LP10 ; CONTINUE IF CLEAR

IS: ; CONTINUE IF CLEAR

IS:

BIT #BIT14, $SREG ; CHECK FOR LOOP ON ERROR
BNE LP10 ; GO TO LOOP ERROR

MOV #20, $FATAL ; MOVE ERROR NUM TO MAILBOX

BNE LP10 ; GO TO LOOP ERROR

TST $SREG ; CHECK FOR HLT ON ERROR
BMT IS ; HALT IF SET

IS: ;<BIT15 IN RCSR NOT CLEAR BY RESET>

IS:

BIT #BIT10, $SREG ; CHECK FOR LOOP ON TEST
BNE TST11 ; GO TO LOOP ON TEST
TEST THAT LOADING TRUF CLEARS TCSR BIT7 (READY) AND THAT BIT IS SET AFTER THE TRANSMITTER IS DONE.

TST12:

 sett: MOV #12 STSTTH ; MOVE TEST NUMBER TO MAILBOX
 BIT #0110, BCAV ; CHECK IF ON APT
 BIT #0111, BCAV ; IF NOT
 BEQ #0000, NOAPT ; BR IF NOT FIRST PASS
 BIT #00000011, NPA ; CHECK IF IN FIRST PASS
 BEQ #00000001, RO ; BR IF NOT FIRST PASS
 BEQ #00000000, NOAPT ; IF NOT FIRST PASS - SKIP TEST
 BNE TS113 ; IF CONSOLE IS UNDER TEST

NOAPT:

LP11: RESET ; INITIALIZE (SET TCSR BIT7)
 CLR #00000000, BCS1 ; MOVE ZEROS TO BUFFER
 CLR #01111111, BCS1 ; CHECK IF BIT7 CLEAR.
 BIT #00000000, BCS1 ; CONTINUE IF CLEAR
 ; DO IT ONE MORE TIME IN CASE REFRESH BODIED THE FIRST TEST
 CLR #00000000, BCS1 ; DROP READY WITH THIS CHAR.
 CLR #01111111, BCS1 ; CHECK FOR READY LOW
 BEQ #00000000, RO ; THIS IS A REAL ERROR IF NOT 0
 BEQ #01111111, RO ; CONTINUE IF SET.
 BEQ #00000000, RO ; CHECK FOR LOOP ON ERROR
 BNE #00000000, RO ; GO TO LOOP ERROR
 MOV #00000000, BCS1 ; MOVE ERROR NUM TO MAILBOX
 TEST #00000000, BCS1 ; CHECK FOR HALT ON ERROR
 BNE TS113 ; HALT IF SET

HALT ; (TCSR BIT7 DID NOT CLEAR AFTER DATA XRMT)

ISR:

TCS1: CLR #00000000, R1 ; CLEAR COUNT REG.
 BIT #00000000, R1 ; SET COUNT
 DEC# DEC R1 ; DECREMENT COUNT
 DEC R1 ; CONTINUE IF NOT 0
 BEQ #01100000, RO ; CONTINUE IF READY IS BACK ON.
 BEQ #11111111, RO ; CHECK IF BIT7 ON RETURN.
 BIT #00000000, BCS1 ; CONTINUE IF SET.
 BEQ #01111111, RO ; CHECK FOR LOOP ON ERROR
 BNE #00000000, RO ; GO TO LOOP ERROR
 MOV #00000000, BCS1 ; MOVE ERROR NUM TO MAILBOX
 TEST #00000000, BCS1 ; CHECK FOR HALT ON ERROR
 BNE TS113 ; HALT IF SET

HALT ; (TCSR BIT7 DID NOT SET AFTER XRMT)

ISR:

BIT #0110, SSMREG ; CHECK FOR LOOP ON TEST
 BNE TS112 ; GO TO LOOP ON TEST

TEST THAT THE TRANSMITTER CAN INTERRUPT AT THE CORRECT VECTOR

TST13:

 sett: MOV #13 STSTTH ; MOVE TEST NUMBER TO MAILBOX
 BIT #0110, BCAV ; CHECK IF ON APT
 BIT #0111, BCAV ; IF NOT ON APT
 BEQ #00000011, NPA ; BR IF NOT FIRST PASS
 BIT #00000001, NPA ; CHECK IF FIRST PASS
 BEQ #00000000, TS113 ; BR, IF NOT FIRST PASS
 BEQ #00000000, TS113 ; IF NOT FIRST PASS - SKIP TEST
J02

; IF CONSOLE IS UNDER TEST
TST14:
BNE TST14

; DISABLE XMITTER INTERRUPT
BIC @BIT5, @TCSR

; SET VECTOR ADDRESS
MOV @T31CN, @TVECT

; SET PSH FOR INTERRUPT
MOV @TPSH

; ALLLOW INTERRUPTS
MOV 1, @BIT5, @TCSR

; ENABLE INTERRUPTS
BR, LP12

; GIVE IT SOME TIME

; CHECK FOR LOOP ON ERROR
BIT @BIT14, @SSNREG

; GO TO LOOP ERROR
BNE TST12

; MOVE ERROR NUM TO MAILBOX
MOV 01, @BIT5, @TCSR

; CHECK FOR HALT ON ERROR
BR, TST11

; HALT IF SET
HALT

; NO INTERRUPT FROM DEVICE

; CHECK FOR LOOP ON TEST
BIT @BIT10, @SSNREG

; GO TO LOOP ON TEST
BNE TST10

; DISABLE INTERRUPT
BIC @BIT5, @TCSR

; TEST THAT TRANSMITTER DOES NOT INTERRUPT WHEN PSH DISABLES
TST14:

; MOVE TEST NUMBER TO MAILBOX
MOV @BIT14, @SMON

; CHECK IF ON APT
BIT TST14A

; IF NOT ON APT
BR, TST14A

; CHECK IF IN FIRST PASS
BR, TST14A

; BR, IF NOT FIRST PASS
BR, TST14A

; IF NOT FIRST PASS - SKIP TEST
BR, TST14A

; IF CONSOLE IS UNDER TEST
TST14A:

; CLEAR INT ENABLE INTERRUPT
BIC @BIT5, @TCSR

; DISABLE INTERRUPT
MOV @TPSH

; Vector point to error
MOV @TVECT

; CLEAR PSH FOR INTERRUPT
CLR @TPSH

; ENABLE INTERRUPTS
BR, LP13

; GIVE IT SOME TIME

; CONTINUE IF NO INTERRUPT.

; CHECK FOR LOOP ON ERROR
BIT @BIT14, @SSNREG

; GO TO LOOP ERROR
BNE TST13

; MOVE ERROR NUM TO MAILBOX
MOV 01, @BIT5, @TCSR

; CHECK FOR HALT ON ERROR
BR, TST11

; HALT IF SET
HALT

; DEVICE GAVE INTERRUPT WITH PSH DISABLE

; CHECK FOR LOOP ON TEST
BIT @BIT10, @SSNREG

; GO TO LOOP ON TEST
BNE TST10

; TEST THAT XMITTER DOES NOT RE-INTERRUPT AFTER THE FIRST INTERRUPT IS SERVICED.
TST15:

; MOVE TEST NUMBER TO MAILBOX
MOV @BIT15, @SMON

; CHECK IF ON APT
BIT @BIT10, @SMON
NO2.

012737 005435 000024 PURUP: MOV #PHRFL,3424 ; SET POWER FAIL VECTOR

055000 000028 CLRR 4026 ; SET POWER UP PSW

002000 000026 CLRR 40 ; CLEAR TTY WAIT TIMER

001376 000220 INE 40 ; INCREMENT TIMER

014767 177634 TTYWT: BNE TTYWT ; WAIT FOR TTY POWER UP

000610 172520 JSR PC, MSGPR ; TYPE "POWER" MESSAGE

000167 172520 .WORD CP ; START AT BEGINNING OF PRESENT PASS

040515 047111 042504 M1: .ASCIZ MAINDEC-11-DVKAEB-B

OLVII TEST. <15>(12)

025700 053114 053105 M2: .ASCIZ DEVICES UNDER TEST. <15>(12)

020700 053105 020700 M3: .ASCIZ END OF PASS. <15>(12)

042527 006522 M4: .ASCIZ POWER. <15>(12)

.END
<table>
<thead>
<tr>
<th>REFERENCE</th>
<th>1177</th>
<th>1192</th>
<th>1226</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1236</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>559</td>
<td>550</td>
<td>580</td>
</tr>
<tr>
<td>1162</td>
<td></td>
<td>1182</td>
<td>1195</td>
</tr>
<tr>
<td>1221</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1182</td>
<td>1119</td>
<td>1137</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1143</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1123</td>
<td>1124</td>
<td>1135</td>
</tr>
<tr>
<td></td>
<td>1284</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>565</td>
<td>575</td>
<td>585</td>
</tr>
<tr>
<td>580</td>
<td></td>
<td>595</td>
<td>605</td>
</tr>
<tr>
<td>600</td>
<td></td>
<td>620</td>
<td>630</td>
</tr>
<tr>
<td>640</td>
<td></td>
<td>655</td>
<td>665</td>
</tr>
<tr>
<td>670</td>
<td></td>
<td>685</td>
<td>695</td>
</tr>
<tr>
<td>710</td>
<td></td>
<td>720</td>
<td>730</td>
</tr>
<tr>
<td>750</td>
<td></td>
<td>775</td>
<td>785</td>
</tr>
<tr>
<td>830</td>
<td></td>
<td>860</td>
<td>870</td>
</tr>
<tr>
<td>1000</td>
<td></td>
<td>1030</td>
<td>1060</td>
</tr>
<tr>
<td>TEST</td>
<td>1117</td>
<td>1118</td>
<td>1119</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>BYTE</td>
<td>3116</td>
<td>3116</td>
<td>3116</td>
</tr>
<tr>
<td>DIO</td>
<td>391</td>
<td>392</td>
<td>399</td>
</tr>
<tr>
<td>DIOc</td>
<td>391</td>
<td>392</td>
<td>399</td>
</tr>
<tr>
<td>EVEN</td>
<td>1119</td>
<td>1119</td>
<td>1119</td>
</tr>
<tr>
<td>IF</td>
<td>1211</td>
<td>1211</td>
<td>1211</td>
</tr>
<tr>
<td>LIST</td>
<td>1211</td>
<td>1211</td>
<td>1211</td>
</tr>
<tr>
<td>MATCH</td>
<td>1211</td>
<td>1211</td>
<td>1211</td>
</tr>
<tr>
<td>MIN</td>
<td>1211</td>
<td>1211</td>
<td>1211</td>
</tr>
<tr>
<td>STOR</td>
<td>1211</td>
<td>1211</td>
<td>1211</td>
</tr>
</tbody>
</table>

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0
# DVMRED.SEO/SOL/COR/PRGM/ML:TOC=SYSMAC.CD,DVMRED.P11
RUN-TIME: 24 28 2 SECONDS
RUN-TIME RATIO: 217/54=3.8
CORE USED: 33K (65 PAGES)