```
.REM ^
```

IDENTIFICATION

PRODUCT CODE: AC-E947B-MC

PRODUCT NAME: CXDVAB0 DV11 MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1976,1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

DVA IS AN IOMOD THAT EXERCISES UP TO AND INCLUDING FOUR
(CONSECUTIVELY ADDRESSED) DV11 SYNCRONOUS INTERFACES.
IT USES MAINTAINCE MODE TO XMIT AND RECEIVE A SET DATA
PATTERN RESEMBLING A BINARY COUNT FROM 00-77. THE
RECEIVER AND TRANSMITTER ISR ARE ALWAYS AT PRIORITY FIVE. (BR1,BR2)
DATA CHECKING IS PERFORMED AT LEVEL 0
AND DONE OUTSIDE THE ISRS.

2. REQUIREMENTS

HARDWARE: AT LEAST 1 DV11-AA AND ONE DV11-BA

STORAGE:: DVA REQUIRES:
1. DECIMAL WORDS: 3939
2. OCTAL WORDS: 07543
3. OCTAL BYTES: 17306

3. PASS DEFINITION

ONE PASS OF THE DVA MODULE CONSISTS OF HANDLING
100(8) CHARS TIMES THE NUMBER OF SELECTED LINES TIMES
THE NUMBER OF SELECTED DEVICES TIMES 600(8).

4. EXECUTION TIME

THIS VARIES DUE TO THE CONFIGURATION BUT SHOULD ALWAYS
BE UNDER 01 MIN.

5. CONFIGURATION PARAMETERS.

DEFAULT PARAMETERS:
ADDR: 175000, VECTOR: 310, BR1: 5, BR2: 5, DVID1: 1, SR1:0
FIRST EIGHT LINES FOR EACH DV11.
SYNC "A" SET TO 226
USER *MUST* SET SOFTWARE "SYNC" TO "377"
FOR TEST OF AN ASYNC LINE CARD. SEE SECTION 8.
USER MAY ALTER PARAMETERS. NO MORE THAN 4 DV11'S MAY BE RUN.

6. DEVICE OPTION SETUP

1. PARITY OFF
2. AT LEAST 6 BITS PER CHAR
3. SYNC "A" OVER(GREATER THAN) 20(8)
4. FULL DUPLEX

DEFAULT SHIP OF 8 BITS,PARITY OFF, FULL DUPLEX, ETC OK

TRANSMITER PATTERN

```
* \S S\ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD \ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD\  LINE: 0
** \S S\ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD \ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD\  LINE: 1
*** \S S\ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD \ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD\  LINE: 2
**** \S S\ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD \ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD\  LINE: 3
***** \S S\ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD \ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD\  LINE: 4
****** \S S\ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD \ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD\  LINE: 5
******* \S S\ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD \ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD\  LINE: 6
******** \S S\ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD \ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD\  LINE: 7
-----------------------------------------------------------------------------------
* \S S\ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD \ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD\  LINE: 8
** \S S\ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD \ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD\  LINE: 9
*** \S S\ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD \ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD\  LINE: 10
**** \S S\ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD \ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD\  LINE: 11
***** \S S\ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD \ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD\  LINE: 12
****** \S S\ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD \ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD\  LINE: 13
******* \S S\ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD \ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD\  LINE: 14
******** \S S\ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD \ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD\  LINE: 15
```

RECEIVER PATTERN

```
DDDDDDDDDDDDDDDDDDD#D#D#D#D#D#DD#DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD BB\ E\  LINE: 0
 DDDDDDDDDDDDDDDDDDD#D#D#D#D#D#DD#DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD BB\ E\  LINE: 1
  DDDDDDDDDDDDDDDDDDD#D#D#D#D#D#DD#DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD BB\ E\  LINE: 2
   DDDDDDDDDDDDDDDDDDD#D#D#D#D#D#DD#DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD BB\ E\  LINE: 3
    DDDDDDDDDDDDDDDDDDD#D#D#D#D#D#DD#DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD BB\ E\  LINE: 4
     DDDDDDDDDDDDDDDDDDD#D#D#D#D#D#DD#DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD BB\ E\  LINE: 5
      DDDDDDDDDDDDDDDDDDD#D#D#D#D#D#DD#DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD BB\ E\  LINE: 6
       DDDDDDDDDDDDDDDDDDD#D#D#D#D#D#DD#DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD BB\ E\  LINE: 7
-----------------------------------------------------------------------------------
DDDDDDDDDDDDDDDDDDD#D#D#D#D#D#DD#DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD BB\ E\  LINE: 8
 DDDDDDDDDDDDDDDDDDD#D#D#D#D#D#DD#DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD BB\ E\  LINE: 9
  DDDDDDDDDDDDDDDDDDD#D#D#D#D#D#DD#DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD BB\ E\  LINE: 10
   DDDDDDDDDDDDDDDDDDD#D#D#D#D#D#DD#DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD BB\ E\  LINE: 11
    DDDDDDDDDDDDDDDDDDD#D#D#D#D#D#DD#DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD BB\ E\  LINE: 12
     DDDDDDDDDDDDDDDDDDD#D#D#D#D#D#DD#DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD BB\ E\  LINE: 13
      DDDDDDDDDDDDDDDDDDD#D#D#D#D#D#DD#DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD BB\ E\  LINE: 14
       DDDDDDDDDDDDDDDDDDD#D#D#D#D#D#DD#DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD BB\ E\  LINE: 15
```

DEFINITIONS:

```
"*"  PAD(JUNK) CHARS XMITED TO STAGGER INTERUPTS. (NOT RECEIVED).
"S"  SYNC CHARS XMITTED TO GET RECEIVER ACTIVE. (NOT RECEIVED).
"D"  TX: DATA TRANSMITTED FROM MEMORY.  RX: DATA RECEIVED AND STORED IN MEMORY.
"#"  DLE CHARS IDLED FROM DV11 TRANSMITER  AND STORED INTO MEMORY BY RX.
"\"  INDICATES BUS INTERUPT.
"B"  BCC CHARS (RECEIVED BUT NOT STORED).
"E"  ETX  (END OF TEXT) LAST CHAR RECEIVER EXPECTS (NOT STORED).
" "  SPACES USED FOR CLARITY. NO FUNCTION
```

7. MODULE OPERATION

```
1.  LOAD SOFTWARE POINTERS IN LNKTABLE.
2.  LOAD VECTORS AND PRIORITIES IN TABLE
3.  DETERMINE WHICH SET OF LINES TO TEST
    (PING. PONG =0 LINES 00-07;  PING. PONG <>0 LINES 08-15)
4.  CLEAR ALL SECONDARY REGISTERS AND MASTER CLEAR DEVICE.
5.  LOAD ALL SECONDARY REGISTERS FOR SELECTED LINES
6.  UPDATE LINE COUNTER IF NOT DONE GOTO 5.
7.  LOAD INDIVIDUAL SECONDARY REGISTERS THAT VARY FROM
    LINE TO LINE OR DEVICE TO DEVICE.
8.  ENABLE SELECTED DEVICES.
9.  SCAN FOR ALL LINES OF ALL DEVICES TO FINISH
10. IF NOT DONE GOTO 9
    IF HUNG REPORT SO AND DROP MODULE.
11. CHECK DATA FOR ALL LINES SELECTED FOR ALL DEVICES SELECTED.
12. SWITCH TO NEXT GROUP OF LINES (COM   PING.PONG )
13. DECREMENT ITERATION COUNT
14. IF NOT =0 GOTO 1
15. SIGNAL ENDPAS.
16.     TXISR:     TRANSMITTER INTERUPT SERVICE ROUTINE.
17. GET INTERUPTING DVSCR.
18. WAS DVSCR 15=1? (IF NOT; REPORT ERROR)
19. READ DVNSR
20. CHECK FOR EITHER PRI BC OR ALT BC =0
21. RELOAD BA AND BC.
    RTI
22.     RXISR:     RECEIVER INTERUPT SERVICE ROUTINE.
23. GET INTERUPTING DVSCR
24. WAS DVSCR07=1?  (IF NOT; REPORT ERROR)
25. ARE ANY ERROR CODES SET?
26. IF NO ERROR CODES; WAS THIS SPECIAL CHAR 77?
27. IF YES (CHAR =77) THEN SHUT TX OFF AND RESYNC RX. (SIGNAL END OF LINE ACTIVITY)
28. IF ERROR CODE PRESENT THEN WAS IT BCC CODE?
29. IF NO THEN REPORT ERROR. IF YES WAS BCC =0? (IF NOT; REPORT ERROR)
30. RTI
```

8. OPERATOR OPTIONS

LOCATIONS IN DVA FROM 164(8) TO 232(8) ARE BASICALLY
SELF EXPLANITORY AND MAY ALTERED
VIA THE MOD CMD TO ALTER TO THE SPECIFIC DV11 CONFIGURATION.
NOTE: IF YOU HAVE MORE THAN 8 LINES INSTALLED; IT
IS YOUR RESPONSIBILITY TO ALTER THE "LINES(1,2,3,4)
LOCATIONS TO HAVE THESE LINES TESTED.
ASYNCRONOUS LINE CARD TESTING!!
        FOR TESTING OF ASYNC LINE CARDS; BOTH SYNC
        CHARS FOR THAT LINE CARD *MUST* BE SET TO
        "377" (WORD=177777) IN THIS PROGRAM. LOOK AT
        LOCATIONS 164(8) THRU 232(8) FOR CORRECT
        LOCATION OF SYNC CHARS FOR LINE CARD.

9. NON-STANDARD PRINTOUTS

IF THE MODULE "HANGS" IN WHICH NOT ALL LINES FOR
EACH DV11 SELECTED FAIL TO FINISH; A "HUNG" MESSAGE IS
PRINTED OUT. TO FIND OUT WHAT LINE(S) ON WHICH DEVICE(S)
HAVE FAILED; COMPARE THE LINES SELECTED BETWEEN
164(8)-222(8) TO THE IMAGE BEING CREATED AT THE "LNKTAB"
ARGUMENT FOLLOWING THE DVSCR IMAGE OF THE TRANSMITTER "JSR" ROUTINE.

THAT IS TO SAY:

```
LINES1(164) SHOULD EQUAL (COMPARE TO) LOC: XXX1  016742
LINES2(176) SHOULD EQUAL (COMPARE TO) LOC: XXX2  016762
LINES3(210) SHOULD EQUAL (COMPARE TO) LOC: XXX3  017002
LINES4(222) SHOULD EQUAL (COMPARE TO) LOC: XXX4  017022
```

FOR A COMPLETE RUN OF DVA. IF ANY OF THE FOLLOWING
LOCATIONS FAIL TO COMPARE A "HUNG" MESSAGE WILL BE
PRINTED AND THE MODULE WILL BE DROPPED.

```
                                        IOMOD <DVAB >,175000,310,5,5,0,600,74
                                        MODULE  140000,DVAB ,175000,310,5,5,0,600,74
                                        .TITLE  DVAB DEC/X11 SYSTEM EXERCISER MODULE
                                ;       DDXCOM  VERSION  6      23-MAY-78
                                        .LIST   BIN
                                ;*****************************************************************
000000'                         BEGIN:
000000'  053104  041101     040 MODNAM: .ASCII  /DVAB / ;MODULE NAME.
000005'      000                XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006'  175000                 ADDR:   175000+0                ;1ST DEVICE ADDR.
000010'  000310                 VECTOR: 310+0                   ;1ST DEVICE VECTOR.
000012'      240                BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013'      240                BR2:    .BYTE   PRTY5+0         ;2ND BR LEVEL.
000014'  000001                 DVID1:  0+1                     ;DEVICE INDICATOR 1.
000016'  000000                 SR1:    OPEN                    ;SWITCH REGISTER 1
000020'  000000                 SR2:    OPEN                    ;SWITCH REGISTER 2
000022'  000000                 SR3:    OPEN                    ;SWITCH REGISTER 3
000024'  000000                 SR4:    OPEN                    ;SWITCH REGISTER 4
                                ;*****************************************************************
000026'  140000                 STAT:   140000                  ;STATUS WORD.
000030'  000314'                INIT:   START                   ;MODULE START ADDR.
000032'  000224'                SPOINT: MODSP                   ;MODULE STACK POINTER.
000034'  000000                 PASCNT: 0                       ;PASS COUNTER.
000036'  000600                 ICONT:  600                     ;# OF ITERATIONS PER PASS=600
000040'  000000                 ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042'  000000                 SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044'  000000                 HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046'  000000                 SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050'  000000                 HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052'  000000                 SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054'  000000                 RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:                         ;RESERVED FOR MONITOR USE
000056'  000000                 RES1:   0                       ;RESERVED FOR MONITOR USE
000060'  000000                 RES2:   0                       ;RESERVED FOR MONITOR USE
000062'  000000                 SVR0:   OPEN                    ;LOC TO SAVE R0.
000064'  000000                 SVR1:   OPEN                    ;LOC TO SAVE R1.
000066'  000000                 SVR2:   OPEN                    ;LOC TO SAVE R2.
000070'  000000                 SVR3:   OPEN                    ;LOC TO SAVE R3.
000072'  000000                 SVR4:   OPEN                    ;LOC TO SAVE R4.
000074'  000000                 SVR5:   OPEN                    ;LOC TO SAVE R5.
000076'  000000                 SVR6:   OPEN                    ;LOC TO SAVE R6.
000100'  000000                 CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102'                         SBADR:                          ;ADDR OF GOOD DATA, OR
000102'  000000                 ACSR:   OPEN                    ;CONTENTS OF CSR.
000104'                         WASADR:                         ;ADDR OF BAD DATA, OR
000104'  000000                 ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106'                         ERRTYP:                         ;TYPE OF ERROR
000106'  000000                 ASB:    OPEN                    ;EXPECTED DATA.
000110'  000000                 AWAS:   OPEN                    ;ACTUAL DATA.
000112'  000342'                RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114'  000000                 WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116'  000000                 WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120'  000000                 INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122'  000074                 IDNUM:  74                      ;MODULE IDENTIFICATION NUMBER=74
000224'                         MODSP:
                                ;*****************************************************************
```



```
                                        ;*SETUP VARIABLES NEEDED FOR DV11 CONFIGURATION

000224'  000377                 LINES1: ^B<0000000011111111>    ;DEFAULT ALL 8 LINES
000226'      226      226       SYNC11: .BYTE   226,226         ;DEFAULT SYNC A  (1)
000230'      226      226       SYNC12: .BYTE   226,226         ;DEFAULT SYNC A  (2)
000232'      226      226       SYNC13: .BYTE   226,226         ;DEFAULT SYNC A  (3)
000234'      226      226       SYNC14: .BYTE   226,226         ;DEFAULT SYNC A  (4)

000236'  000377                 LINES2: ^B<0000000011111111>    ;DEFAULT ALL 8 LINES
000240'      226      226       SYNC21: .BYTE   226,226         ;DEFAULT SYNC A  (1)
000242'      226      226       SYNC22: .BYTE   226,226         ;DEFAULT SYNC A  (2)
000244'      226      226       SYNC23: .BYTE   226,226         ;DEFAULT SYNC A  (3)
000246'      226      226       SYNC24: .BYTE   226,226         ;DEFAULT SYNC A  (4)

000250'  000377                 LINES3: ^B<0000000011111111>    ;DEFAULT ALL 8 LINES
000252'      226      226       SYNC31: .BYTE   226,226         ;DEFAULT SYNC A  (1)
000254'      226      226       SYNC32: .BYTE   226,226         ;DEFAULT SYNC A  (2)
000256'      226      226       SYNC33: .BYTE   226,226         ;DEFAULT SYNC A  (3)
000260'      226      226       SYNC34: .BYTE   226,226         ;DEFAULT SYNC A  (4)

000262'  000377                 LINES4: ^B<0000000011111111>    ;DEFAULT ALL 8 LINES
000264'      226      226       SYNC41: .BYTE   226,226         ;DEFAULT SYNC A  (1)
000266'      226      226       SYNC42: .BYTE   226,226         ;DEFAULT SYNC A  (2)
000270'      226      226       SYNC43: .BYTE   226,226         ;DEFAULT SYNC A  (3)
000272'      226      226       SYNC44: .BYTE   226,226         ;DEFAULT SYNC A  (4)

                 000000         DVSCR=R0        ;SYSTEM CONTROL REGISETR
                 000002         DVRIC=2         ;RECEIVER INTERUPT CHAR REGISTER
                 000004         DVLCR=4         ;LINE CONTROL REGISTER
                 000006         DVSRS=6         ;SECONDARY REGISTER SELECT REGISTER
                 000007         DVSRSH=7        ;SECONDARY REGISTER SELECT REGISTER (HIGH BYTE)
                 000010         DVSRA=10        ;SECONDARY REGISTER ACCESS REGISTER
                 000012         DVSFR=12        ;SPECIAL FUNCTIONS REGISTER
                 000014         DVNSR=14        ;NPR STAUS REGSSTER
                 000016         DVRES=16        ;RESERVED REGISTER

                 000000         MODE0=0*40
                 000040         MODE1=1*40
                 000100         MODE2=2*40
                 000140         MODE3=3*40
                 000200         MODE4=4*40
                 000240         MODE5=5*40
                 000300         MODE6=6*40
                 000340         MODE7=7*40

000274'  000001                 PING.PONG:      .BLKW 1
000276'  000001                 VA:     .BLKW 1
000300'  000001                 PA:     .BLKW 1
000302'  000001                 EA:     .BLKW 1
000304'  000001                 SELECT: .BLKW 1
000306'  000001                 XCNT:   .BLKW 1
000310'  000001                 HOLD:   .BLKW 1
000312'  000001                 STORE:  .BLKW 1
```

```
                                        ;       BEGIN THE TESTS FOR THE DV11
                                        ;
000314' 012767  002000  177572          START:  MOV     #2000,WDTO      ;2000 WORDS TO MEM/ITERATION
000322' 012767  002000  177566                  MOV     #2000,WDFR      ;2000 WORDS FROM MEM/ITERATION
000330' 012767  000100  177562                  MOV     #64.,INTR       ;64. INTERRUPTS/ITERATION
000336' 005067  177732                          CLR     PING.PONG       ;SET WHICH LINES TO DO
000342' 005067  001534                  RESTRT: CLR     ABORT           ;CLEAR ABORT SERVICE FLAG
000346' 016767  177442  177730                  MOV     DVID1,SELECT    ;GET ACTIVE DEVICES.
000354' 001002                                  BNE     CONT            ;BR IF ANY *ARE* SELECTED
000356'                                 DROP:
000356' 104410  000000'                         END$,BEGIN              ;;NO DV11'S SELECTED!

000362' 032767  177760  177714          CONT:   BIT     #^C<17>,SELECT  ;MAKE SURE NO MORE THAN 4 DEVICES SELECTED
000370' 001372                                  BNE     DROP            ;ABSOLUTLY INVALID. NO MORE THAN 4 DEVICES!!
000372' 016701  177706                  SETUP1: MOV     SELECT,R1       ;GET IMAGE OF RUNNING DEVICES
000376' 001767                                  BEQ     DROP            ;BR IF ALL DV11'S DROPPED
000400' 016702  177404                          MOV     VECTOR,R2       ;SET INITIAL VECTOR
000404' 016700  177376                          MOV     ADDR,R0         ;SET INITAL CSR
000410' 012703  017006'                         MOV     #LNKTAB,R3      ;SET POINTER FOR ISR
000414' 016763  177604  000016                  MOV     LINES1,14.(R3)  ;LOAD DEFAULT ACTIVE LINES
000422' 016763  177610  000036                  MOV     LINES2,30.(R3)  ;LOAD DEFAULT ACTIVE LINES
000430' 016763  177614  000056                  MOV     LINES3,46.(R3)  ;LOAD DEFAULT ACTIVE LINES
000436' 016763  177620  000076                  MOV     LINES4,62.(R3)  ;LOAD DEFAULT ACTIVE LINES
000444' 006201                          1$:     ASR     R1              ;ACTIVE?
000446' 103410                                  BCS     3$              ;BR IF YES
000450' 001440                                  BEQ     SETUP2          ;BR IF DONE
000452' 062702  000020                  2$:     ADD     #20,R2          ;POP VECTOR POINTER (20)
000456' 062703  000020                          ADD     #20,R3          ;POP SOFTWARE ISR POINTER (20)
000462' 062700  000040                          ADD     #40,R0          ;POP CSR POINTER (40)
000466' 000766                                  BR      1$              ;CONTINUE
000470' 010312                          3$:     MOV     R3,(R2)         ;LOAD ISR POINTER (RECV)
000472' 116762  177314  000002                  MOVB    BR1,2(R2)       ;LOAD PRIORITY
000500' 010063  000004                          MOV     R0,4(R3)        ;LOAD DVSCR POINTER
000504' 010362  000004                          MOV     R3,4(R2)        ;LOAD ISR POINTER (TRAN)
000510' 062762  000010  000004                  ADD     #10.,4(R2)      ;POP IT
000516' 116762  177271  000006                  MOVB    BR2,6(R2)       ;LOAD PRIORITY
000524' 010063  000014                          MOV     R0,14(R3)       ;LOAD DVSCR POINTER
000530' 005767  177540                          TST     PING.PONG       ;WHICH SET OF LINES??
000534' 001003                                  BNE     4$              ;BR IF SECOND SET OF LINES
000536' 105063  000016                          CLRB    14.(R3)         ;*ZERO LOW LINES SELECTED
000542' 000743                                  BR      2$              ;CONT
000544' 105063  000017                  4$:     CLRB    15.(R3)         ;ZERO HIGH LINES SELECTED
000550' 000740                                  BR      2$              ;CONTINUE
000552' 016700  177230                  SETUP2: MOV     ADDR,R0         ;GET INITAL DVSCR
000556' 012701  000224'                         MOV     #LINES1,R1      ;SET LINES POINTER
000562' 005767  177506                          TST     PING.PONG       ;WHICH SET OF LINES?
000566' 001401                                  BEQ     1$              ;BR IF LOW SET OF LINES SELECTED
000570' 005201                                  INC     R1              ;MAKE EQUAL TO HIGH BYTE
000572' 012767  017106' 177510          1$:     MOV     #BUFFER.TABLE,HOLD
000600' 010105                                  MOV     R1,R5           ;PREPARE TO GET SYNC CHAR PONTER
000602' 042705  000001                          BIC     #BIT0,R5        ;INSURE EVEN ADDRESS
000606' 005725                                  TST     (R5)+           ;POP TO SYNC CHARS.
000610' 005767  177460                          TST     PING.PONG       ;WHAT SET OF LINES?
000614' 001401                                  BEQ     2$              ;BR IF LOW SET.
000616' 022525                                  CMP     (R5)+,(R5)+     ;SET OFFSET TO LAST TWO SYNC LINES.
```



```
000620' 006267  177460                  2$:     ASR     SELECT          ;ACTIVE?
000624' 103417                                  BCS     4$              ;YES
000626' 001002                                  BNE     3$              ;BR IF NOT DONE
000630' 000167  000556                          JMP     13$             ;JUMP IT DONE
000634' 062700  000040                  3$:     ADD     #40,R0          ;POP DVSCR POINTER
000640' 062701  000012                          ADD     #12,R1          ;POP LINES POINTER
000644' 010105                                  MOV     R1,R5           ;PREPARE TO GET SYNC CHAR POINTER
000646' 042705  000001                          BIC     #BIT0,R5        ;INSURE EVEN ADDRESS
000652' 005725                                  TST     (R5)+           ;POP TO SYNC CHARS
000654' 062767  000040  177426                  ADD     #32.,HOLD       ;POP BUFFER POINTER
000662' 000756                                  BR      2$              ;CONTINUE
                                        ;       ROUTINE USED TO CLEAR ALL DV11
                                        ;       SECONDARY REGISTERS FOR ALL LINES
                                        ;       THIS MAKES SURE NO JUNK IS LEFT FOR UCPU TO FIND.
000664' 012710  004000                  4$:     MOV     #MRESET,@DVSCR  ;INITIALIZE DV11
000670' 005060  000010                  5$:     CLR     DVSRA(R0)       ;ZERO RAMS
000674' 062760  170361  000006                  ADD     #^C<BIT11+BIT10+BIT9+BIT8+BIT3+BIT2+BIT1+BIT0>+BIT0,DVSRS(R0)
000702' 001372                                  BNE     5$              ;UPDATE TO NEXT LINE AND SEC REG  BR IF NOT DONE
000704' 111103                                  MOVB    (R1),R3         ;*GET ACTIVE LINES
000706' 005002                                  CLR     R2              ;ZERO LINE NO. IMAGE
000710' 005767  177360                          TST     PING.PONG       ;1ST GROUP OR SECOND?
000714' 001402                                  BEQ     6$              ;BR IF 1ST
000716' 052702  000010                          BIS     #BIT3,R2        ;ADJUST LINES
000722' 012767  000004  177356          6$:     MOV     #4,XCNT         ;SET TO CHANGE SYNC CHARS EVERY 4 LINES
000730' 012767  000001  000544                  MOV     #1,17$          ;SET FOR PAD(JUNK) CHAR COUNT.
000736' 000241                          7$:     CLC                     ;CLEAR CARRY
000740' 106003                                  RORB    R3              ;*LINE ACTIVE?
000742' 103413                                  BCS     10$             ;BR IF YES
000744' 005202                          8$:     INC     R2              ;UPDATE LINE POINTER
000746' 005367  177334                          DEC     XCNT            ;4 LINES DONE?
000752' 001004                                  BNE     9$              ;BR IF NO
000754' 005725                                  TST     (R5)+           ;POP POINTER TO NEXT SYNCS
000756' 012767  000004  177322                  MOV     #4,XCNT         ;RESET COUNTER
000764' 105703                          9$:     TSTB    R3              ;*ALL LINES DONE?
000766' 001363                                  BNE     7$              ;BR IF NO
000770' 000721                                  BR      3$              ;CONT
000772' 012704  012752'                 10$:    MOV     #FUNC.X,R4      ;GET SECONDARY REGISTER DATA
000776' 110260  000006                          MOVB    R2,DVSRS(R0)    ;LOAD LINE NUMBER
001002' 105060  000007                          CLRB    DVSRSH(R0)      ;CLEAR DVSRSH IMAGE
001006' 052760  134000  000004                  BIS     #BIT15+RENABL+M.MODE,DVLCR(R0)
001014' 005760  000004                  11$:    TST     DVLCR(R0)       ;WAIT FOR STROBE TO FINISH
001020' 100775                                  BMI     11$             ;BR IF NOT DONE
001022' 012460  000010                  12$:    MOV     (R4)+,DVSRA(R0) ;LOAD SECONDARY REGISTERS
001026' 105260  000007                          INCB    DVSRSH(R0)      ;UPDATE SECONDARY REG POINTER
001032' 105760  000007                          TSTB    DVSRSH(R0)      ;ALL DONE?
001036' 001371                                  BNE     12$             ;BR IF NO
001040' 010260  000006                          MOV     R2,DVSRS(R0)    ;LOAD LINE
001044' 112760  000002  000007                  MOVB    #2,DVSRSH(R0)   ;SELECT TX BAS
001052' 010567  177220                          MOV     R5,VA           ;SET ALTERNATE BA FOR EA BITS.
001056' 004767  000422                          JSR     PC,SET.EA       ;GO GET EA BITS.
001062' 005067  177224                          CLR     STORE           ;PREPARE TO
001066' 111567  177220                          MOVB    (R5),STORE      ;CLEAR TABLE+SYNC+MODE
001072' 062767  006156' 177212                  ADD     #TX.CONTROL.TABLE,STORE
001100' 105077  177206                          CLRB    @STORE          ;SO THAT CNTRL BYTE IS =0
001104' 122715  000377                          CMPB    #377,(R5)       ;IS THIS AN ASYNC LINE CARD?
001110' 001042                                  BNE     18$             ;BR IF NOT ASYNC.
```

```
001112' 005067  177174                  CLR     STORE           ;#ZERO
001116' 111567  177170                  MOVB    (R5),STORE      ;#GET "SYNC" CHAR (S/B 377 FOR ASYNC)
001122' 062767  002156' 177162          ADD     #RX.CONTROL.TABLE,STORE
001130' 112777  000020  177154          MOVB    #BIT4,@STORE    ;#SET DSCARD AND MODE ZERO.
001136' 012760  120000  000004          MOV     #BIT15+BIT13,DVLCR(R0)  ;#SET RX ENABLE.
001144' 005760  000004                  TST     DVLCR(R0)               ;#WAIT FOR STROBE TO FINISH
001150' 100775                          BMI     .-4                     ;#WAIT
001152' 012760  115000  000004          MOV     #BIT15+BIT12+BIT11+BIT9,DVLCR(R0)
001160' 005760  000004                  TST     DVLCR(R0)               ;#STROBE IN 8 BITS/PER/CHAR
001164' 100775                          BMI     .-4                     ;#WAIT FOR STROBE TO FINISH
001166' 012760  172000  000004          MOV     #BIT15+BIT14+BIT13+BIT12+BIT10,DVLCR(R0)
001174' 005760  000004                  TST     DVLCR(R0)               ;#STROBE IN 9600 BAUD RATE.
001200' 100775                          BMI     .-4                     ;#WAIT FOR STROBE DONE.
001202' 012760  107000  000004          MOV     #BIT15+BIT11+BIT10+BIT9,DVLCR(R0)
001210' 005760  000004                  TST     DVLCR(R0)               ;#STROBE IN MAIN INTERNAL MODE
001214' 100775                          BMI     .-4                     ;#WAIT FOR STROBE DONE
001216' 105260  000007          18$:    INCB    DVSRSH(R0)      ;SELECT BYTE CNT (ALTERNATE)
001222' 012760  177776  000010          MOV     #-2,DVSRA(R0)   ;SET FOR 2 SYNC CHARS
001230' 010267  177056                  MOV     R2,STORE        ;GET LINE NUMBER
001234' 006367  177052                  ASL     STORE           ;MULT BY 2 (MAKE EVEN)
001240' 066767  177044  177044          ADD     HOLD,STORE      ;GET POINTER FOR RX BUFFER
001246' 112760  000004  000007          MOVB    #4,DVSRSH(R0)   ;SEL RX BA
001254' 017767  177032  177014          MOV     @STORE,VA       ;PREPARE TO GET EA BITS FOR RX BA.
001262' 004767  000216                  JSR     PC,SET.EA       ;GO GET THEM
001266' 112760  000010  000007          MOVB    #10,DVSRSH(R0)  ;SET TX TABLE BASE ADD
001274' 016067  000010  176774          MOV     DVSRA(R0),VA    ;READ IT
001302' 004767  000176                  JSR     PC,SET.EA       ;SET EA BITS
001306' 105260  000007                  INCB    DVSRSH(R0)      ;SEL RX TABLE BASE ADDR
001312' 016067  000010  176756          MOV     DVSRA(R0),VA    ;READ IT
001320' 004767  000160                  JSR     PC,SET.EA       ;SET EA BITS
001324' 105060  000007                  CLRB    DVSRSH(R0)      ;SEL TX BA PRI.
001330' 016067  000010  176740          MOV     DVSRA(R0),VA    ;READ IT
001336' 004767  000142                  JSR     PC,SET.EA       ;GET EA BITS
001342' 116767  176734  011334          MOVB    EA,PRI.EA       ;SAVE PRINCIPLE EA BITS FOR ISR
001350' 016767  176724  011330          MOV     PA,PRI.PA       ;SAVE PRINCIPLE PA FOR ISR
001356' 062760  000020  000010          ADD     #20,DVSRA(R0)   ;POINT TXBA TO PAD(JUNK) CHARS.
001364' 105260  000007                  INCB    DVSRSH(R0)      ;SET TX BA BYTE CNT.
001370' 016760  000106  000010          MOV     17$,DVSRA(R0)   ;SET NUMBER OF PAD(JUNK) CHARS TO BE SENT.
001376' 005460  000010                  NEG     DVSRA(R0)       ;MAKE COUNT 2'S COMP.
001402' 005267  000074                  INC     17$             ;PAD CNT=PAD CNT+1
001406' 000167  177332                  JMP     8$              ;CONT
001412' 016700  176370          13$:    MOV     ADDR,R0         ;GET DVSCR POINTER
001416' 016767  176372  176660          MOV     DVID1,SELECT    ;GET ACTIVE DEVICES
001424' 016701  176654                  MOV     SELECT,R1       ;GET ACTIVE DV11'S
001430' 006201                  14$:    ASR     R1              ;ACTIVE?
001432' 103404                          BCS     16$             ;BR IF YES
001434' 001436                          BEQ     SCANNER         ;BR IF DONE
001436' 062700  000040          15$:    ADD     #40,R0          ;POP DVSCR POINTER
001442' 000772                          BR      14$             ;CONT
001444' 012767  013146' 176624  16$:    MOV     #TXBAS,VA       ;GET EA BITS FOR ALTERNATE BA
001452' 104415  000000' 000276'         GETPA$,BEGIN, VA                ;GET PHYSICAL ADDRESS FROM 16-BIT VA
001460' 116767  176616  011217          MOVB    EA,ALT.EA       ;SAVE ALTERNATE EA BITS
001466' 016767  176606  011214          MOV     PA,ALT.PA       ;SAVE ALTERNATE PA FOR ISR
001474' 052710  030101                  BIS     #NPR.IE+STORE.IE+RX.IE+BIT0,@DVSCR
                                                                ;SET MICO-PROCESSOR GO!!
001500' 000756                          BR      15$             ;CONT
```



```
001502' 000000                  17$:    0

001504'                         SET.EA:
001504' 104415  000000' 000276'         GETPA$,BEGIN, VA                ;GET PHYSICAL ADDRESS FROM 16-BIT VA
001512' 042710  000060                  BIC     #BIT5+BIT4,@DVSCR
                                                                ;CLEAR EA BITS
001516' 056710  176560                  BIS     EA,@DVSCR       ;SET EA BITS
001522' 016760  176552  000010          MOV     PA,DVSRA(R0)    ;LOAD PA
001530' 000207                          RTS     PC
```

```
                                        ;-----------------------------------------------------------------
                                        ;       BEGIN TO SCAN FOR ALL SELCETED LINES FOR
                                        ;       ALL SELECTED DEVICES TO FINISH.
                                        ;       IT IT IMPORTANT TO NOTE THAT *ALL* SELECTED
                                        ;       LINES FOR *ALL* SELECTED DEVICES MUST
                                        ;       FINISH FOR AN END PASS CONSIDERATION.
                                        ;
                                        ;-----------------------------------------------------------------
001532'                                 SCANNER:
001532'  012704  000100                         MOV     #100,R4         ;SET DELAY.
001536'  005005                         11$:    CLR     R5              ;SET FOR A LONG DELAY
001540'                                 1$:
001540'  104407  000000'                        BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
001544'  104407  000000'                        BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001550'  005767  000326                         TST     ABORT           ;SHOULD ROUTINE BE ABORTED?
001554'  001422                                 BEQ     65$             ;BR IF NO
001556'  005767  000320                         TST     ABORT           ;+1=TX ERR, -1=RX ERR
001562'  100407                                 BMI     63$             ;BR IF RX ERROR
                                        ;********** TRANSMITTER ERROR! ********
                                                        ;CHECK MADE IN THIS ORDER.
                                                        ;IF "CSRC" IS = TO DVSCR (SEL 0)
                                                        ;THEN THE ERROR WAS: BIT 15 NOT SET ON INTERUPT
                                                        ;IF "CSRC" IS = TO DVNSR (SEL 14)
                                                        ;THEN THE ERROR WAS: NEITHER PRINCIPAL OR
                                                        ;ALTERNATE BYTE COUNT WAS THE ENTRY IN DVNSR
001564'  012767  000020  176314                 MOV     #20,ERRTYP      ;UNKNOWN XMITTER ERROR
                                                        ;*****************************************************
001572'  104405  000000' 000000                 HRDER$,BEGIN,NULL       ;;SEE ABOVE
                                                        ;*****************************************************
001600'  000406                                 BR      64$             ;CONTINUE
                                        ;-----------------------------------------
                                        ;********* RECEIVER ERROR! ***************
                                                        ;CHECK MADE IN THIS ORDER
                                                        ;IF "CSRC " IS = TO DVSCR (SEL 0)
                                                        ;THEN THE ERROR WAS: BIT 07 NOT SET ON INTERUPT
                                                        ;IF "CSRC" IS=TO DVSRA (SEL 10)
                                                        ;THEN THE ERROR WAS: TRANS UNDERRUN OCCURED
                                                        ;IF "CSRC" IS=TO DVRIC (SEL 2)
                                                        ;THEN THE ERROR WAS: BCC ERROR OR
                                                        ;ILLEGAL ERROR CONDITION
001602'                                 63$:
001602'  012767  000017  176276                 MOV     #17,ERRTYP      ;UNKNOWN RECEIVER ERROR
                                                        ;*****************************************************
001610'  104405  000000' 000000                 HRDER$,BEGIN,NULL       ;;SEE ABOVE
                                                        ;*****************************************************
001616'  000167  176520                 64$:    JMP     RESTRT          ;RESTART MODULE
```



```
001622'  026767  176376  015174          65$:    CMP     LINES1,XXX1     ;DID ALL LINES FOR 1ST DV11 FINISH?
001630'  001113                                  BNE     10$             ;BR IF NO
001632'  026767  176400  015204                  CMP     LINES2,XXX2     ;DID ALL LINES FOR 2ND DV11 FINISH?
001640'  001107                                  BNE     10$             ;BR IF NO
001642'  026767  176402  015214                  CMP     LINES3,XXX3     ;DID ALL LINES FOR 3RD DV11 FINISH?
001650'  001103                                  BNE     10$             ;BR IF NO
001652'  026767  176404  015224                  CMP     LINES4,XXX4     ;DID ALL LINES FOR THE 4TH FINISH?
001660'  001077                                  BNE     10$             ;BR IF NO
001662'  016767  176120  176210                  MOV     ADDR,CSRA       ;LOAD BASE DV11 CSR
001670'  012767  017106' 000206                  MOV     #BUFFER.TABLE,BUFPNT
                                                                         ;SET POINTER OF ALL RECV BUFFERS.
001676'  012701  000224'                         MOV     #LINES1,R1      ;SET POINTER OF 1ST LINES ACTIVE PARAMETER.
001702'  005767  176366                          TST     PING.PONG       ;WHAT SET?
001706'  001404                                  BEQ     2$              ;LO
001710'  062767  000020  000166                  ADD     #16.,BUFPNT     ;HI
001716'  005201                                  INC     R1              ;HIGH SET
001720'  111102                          2$:     MOVB    (R1),R2         ;*GET ACTIVE LINES INTO R2
001722'  016700  000156                          MOV     BUFPNT,R0       ;SET POINTER.
001726'  006267  176352                          ASR     SELECT          ;SPOOL FOR ACTIVE DEVICES
001732'  103412                                  BCS     4$              ;BR IF THE DEVICE WAS ACTIVE
001734'  001417                                  BEQ     5$              ;BR IF ALL DEVICES DONE.
001736'  062767  000040  176134          99$:    ADD     #40,CSRA        ;UPDATE CSR IMAGE
001744'  062767  000040  000132                  ADD     #32.,BUFPNT     ;POP AN ENTIRE DV11S LINE BUFFER
001752'  062701  000012                  3$:     ADD     #12,R1          ;POP TO NEXT GROUP OF LINES
001756'  000760                                  BR      2$              ;CONTINUE ALONG
001760'  000241                          4$:     CLC                     ;CLEAR CPU CARRY
001762'  106002                                  RORB    R2              ;*SPOOL FOR ACTIVE LINES
001764'  103411                                  BCS     7$              ;BR IF LINE WAS ACTIVE
001766'  001763                                  BEQ     99$             ;BR IF ALL LINES FOR DV11 DONE
001770'  005720                                  TST     (R0)+           ;POP BUFFER POINTER TO NEXT BUFFER
001772'  000772                                  BR      4$              ;CONTINUE ALONG
001774'  005167  176274                  5$:     COM     PING.PONG       ;*PREPARE SELECTOR
002000'  104413  000000'                         ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                         ;MONITOR SHALL TEST END OF PASS
                                        ;-----------------------------------------------------------------
                                        ;-----------------------------------------------------------------
002004'  000167  176332                  6$:     JMP     RESTRT          ;RESTART PROGRAM
002010'  012004                          7$:     MOV     (R0)+,R4        ;GET RECEIVER BUFFER POINTER
002012'  012705  013012'                         MOV     #DATA.TABLE,R5  ;GET GOOD DATA POINTER
002016'  121514                          8$:     CMPB    (R5),(R4)       ;DO THE DATA CHECKING
002020'  001412                                  BEQ     9$              ;BR IF DATA IS GOOD
002022'  010567  176054                          MOV     R5,SBADR        ;PREPARE TO REPORT DATA ERROR!
002026'  010467  176052                          MOV     R4,WASADR       ;LOAD GOOD ADDR AND BAD ADDR
002032'  111567  176050                          MOVB    (R5),ASB        ;LOAD GOOD DATA
002036'  111467  176046                          MOVB    (R4),AWAS       ;LOAD BAD DATA
                                                        ;*****************************************************
002042'  104404  000000'                         DATER$,BEGIN            ;DATA ERROR!!!
                                                        ;*****************************************************
002046'  105024                          9$:     CLRB    (R4)+           ;CLEAR BUFFER. MAY BE PATCHED TO "TSTB (R4)+"(105724)
002050'  122725  000076                          CMPB    #76,(R5)+       ;HAS ALL DATA BEEN CHECKED?
002054'  001360                                  BNE     8$              ;BR IF NO
002056'  000740                                  BR      4$              ;CONT. ALL DATA CHECKED.
002060'  005305                          10$:    DEC     R5              ;STALL FOR ALL DV11'S TO FINISH
002062'  001226                                  BNE     1$              ;BR IF DELAY NOT =0
002064'  005304                                  DEC     R4              ;DELAY COUNT
002066'  001223                                  BNE     11$             ;KEEP WAITING.
```

```
002070' 104403  000000' 002106'          MSGN$,BEGIN,HUNG          ;ASCII MESSAGE CALL WITH COMMON HEADER
002076' 104410  000000'                  END$,BEGIN                ;
002102' 000000                   ABORT:  .WORD 0                   ;ABORT FLAG
002104' 000000                   BUFPNT: 0
002106' 002112'                  HUNG:   XHUNG                     ;POINTER TO ASCIZ MSG
002110' 177777                           -1                        ;TERMINATOR
002112' 042045  030526  020061   XHUNG:  .ASCIZ  '%DV11 MODULE IS HUNG. SEE LISTING%'
002120' 047515  052504  042514
002126' 044440  020123  052510
002134' 043516  020056  042523
002142' 020105  044514  052123
002150' 047111  022507     000
        002156'                  .EVEN
```



```
002156'                          RX.CONTROL.TABLE:
                                 ;UNUSED AREA OF CONTROL TABLE IS FILLED WITH 'GEN INT'(MODE0)
                                 ;TO CATCH RECEIVER GOING TO WRONG CNTRL BYTE.

        002156'                  .=RX.CONTROL.TABLE+0              ;MODE 0
        002156'                  .=.+0
002156'     010                          .BYTE BIT3+MODE0                  ;00  INC/BCC+MODE0
002157'     010                          .BYTE BIT3+MODE0                  ;01  INC/BCC+MODE0
002160'     010                          .BYTE BIT3+MODE0                  ;02  INC/BCC+MODE0
002161'     010                          .BYTE BIT3+MODE0                  ;03  INC/BCC+MODE0
002162'     010                          .BYTE BIT3+MODE0                  ;04  INC/BCC+MODE0
002163'     010                          .BYTE BIT3+MODE0                  ;05  INC/BCC+MODE0
002164'     010                          .BYTE BIT3+MODE0                  ;06  INC/BCC+MODE0
002165'     070                          .BYTE BIT4+BIT3+MODE1             ;07  DSCARD+INC/BCC+MODE1
002166'     020                          .BYTE BIT4+MODE0                  ;10  DSCARD+MODE0
002167'     020                          .BYTE BIT4+MODE0                  ;11  DSCARD+MODE0
002170'     020                          .BYTE BIT4+MODE0                  ;12  DSCARD+MODE0
002171'     020                          .BYTE BIT4+MODE0                  ;13  DSCARD+MODE0
002172'     020                          .BYTE BIT4+MODE0                  ;14  DSCARD+MODE0
002173'     020                          .BYTE BIT4+MODE0                  ;15  DSCARD+MODE0
002174'     020                          .BYTE BIT4+MODE0                  ;16  DSCARD+MODE0
002175'     020                          .BYTE BIT4+MODE0                  ;17  DSCARD+MODE0
002176'     020                          .BYTE BIT4+MODE0                  ;20  DSCARD+MODE0
002177'     020                          .BYTE BIT4+MODE0                  ;21  DSCARD+MODE0
002200'     020                          .BYTE BIT4+MODE0                  ;22  DSCARD+MODE0
002201'     020                          .BYTE BIT4+MODE0                  ;23  DSCARD+MODE0
002202'     020                          .BYTE BIT4+MODE0                  ;24  DSCARD+MODE0
002203'     020                          .BYTE BIT4+MODE0                  ;25  DSCARD+MODE0
002204'     020                          .BYTE BIT4+MODE0                  ;26  DSCARD+MODE0
002205'     020                          .BYTE BIT4+MODE0                  ;27  DSCARD+MODE0
```

```
                002556'                 .=RX.CONTROL.TABLE+400          ;MODE1
                002566'                 .=.+10
002566'             050                         .BYTE BIT3+MODE1                ;10  INC/BCC+MODE1
002567'             050                         .BYTE BIT3+MODE1                ;11  INC/BCC+MODE1
002570'             050                         .BYTE BIT3+MODE1                ;12  INC/BCC+MODE1
002571'             050                         .BYTE BIT3+MODE1                ;13  INC/BCC+MODE1
002572'             050                         .BYTE BIT3+MODE1                ;14  INC/BCC+MODE1
002573'             050                         .BYTE BIT3+MODE1                ;15  INC/BCC+MODE1
002574'             050                         .BYTE BIT3+MODE1                ;16  INC/BCC+MODE1
002575'             130                         .BYTE BIT4+BIT3+MODE2           ;17 DSCARD+INC/BCC+MODE2

                003156'                 .=RX.CONTROL.TABLE+1000         ;MODE2
                003176'                 .=.+20
003176'             112                         .BYTE BIT3+BIT1+MODE2           ;20  INC/BCC+STORE+MODE2
003177'             130                         .BYTE BIT4+BIT3+MODE2           ;21  DSCARD+INC/BCC+MODE2
003200'             112                         .BYTE BIT3+BIT1+MODE2           ;22  INC/BCC+STORE+MODE2
003201'             130                         .BYTE BIT4+BIT3+MODE2           ;23  DSCARD+INC/BCC+MODE2
003202'             112                         .BYTE BIT3+BIT1+MODE2           ;24  INC/BCC+STORE+MODE2
003203'             130                         .BYTE BIT4+BIT3+MODE2           ;25  DSCARD+INC/BCC+MODE2
003204'             112                         .BYTE BIT3+BIT1+MODE2           ;26  INC/BCC+STORE+MODE2
003205'             170                         .BYTE BIT4+BIT3+MODE3           ;27  DSCARD+INC/BCC+MODE3
                003556'                 .=RX.CONTROL.TABLE+1400         ;MODE3
                003606'                 .=.+30
003606'             142                         .BYTE BIT1+MODE3                ;30  RESERVED+MODE3
003607'             142                         .BYTE BIT1+MODE3                ;31  RESERVED+MODE3
003610'             142                         .BYTE BIT1+MODE3                ;32  RESERVED+MODE3
003611'             142                         .BYTE BIT1+MODE3                ;33  RESERVED+MODE3
003612'             142                         .BYTE BIT1+MODE3                ;34  RESERVED+MODE3
003613'             142                         .BYTE BIT1+MODE3                ;35  RESERVED+MODE3
003614'             142                         .BYTE BIT1+MODE3                ;36  RESERVED+MODE3
003615'             202                         .BYTE BIT1+MODE4                ;37  RESERVED+MODE4
```



```
                004156'                 .=RX.CONTROL.TABLE+2000         ;MODE4
                004216'                 .=.+40
004216'             212                         .BYTE BIT3+BIT1+MODE4           ;40  INC/BCC+RESERVED+MODE4
004217'             212                         .BYTE BIT3+BIT1+MODE4           ;41  INC/BCC+RESERVED+MODE4
004220'             212                         .BYTE BIT3+BIT1+MODE4           ;42  INC/BCC+RESERVED+MODE4
004221'             212                         .BYTE BIT3+BIT1+MODE4           ;43  INC/BCC+RESERVED+MODE4
004222'             212                         .BYTE BIT3+BIT1+MODE4           ;44  INC/BCC+RESERVED+MODE4
004223'             212                         .BYTE BIT3+BIT1+MODE4           ;45  INC/BCC+RESERVED+MODE4
004224'             212                         .BYTE BIT3+BIT1+MODE4           ;46  INC/BCC+RESERVED+MODE4
004225'             242                         .BYTE BIT1+MODE5                ;47  RESERVED+MODE5

                004556'                 .=RX.CONTROL.TABLE+2400         ;MODE5
                004626'                 .=.+50
004626'             252                         .BYTE BIT3+BIT1+MODE5           ;50  INC/BCC+RESERVED+MODE5
004627'             252                         .BYTE BIT3+BIT1+MODE5           ;51  INC/BCC+RESERVED+MODE5
004630'             252                         .BYTE BIT3+BIT1+MODE5           ;52  INC/BCC+RESERVED+MODE5
004631'             252                         .BYTE BIT3+BIT1+MODE5           ;53  INC/BCC+RESERVED+MODE5
004632'             252                         .BYTE BIT3+BIT1+MODE5           ;54  INC/BCC+RESERVED+MODE5
004633'             252                         .BYTE BIT3+BIT1+MODE5           ;55  INC/BCC+RESERVED+MODE5
004634'             252                         .BYTE BIT3+BIT1+MODE5           ;56  INC/BCC+RESERVED+MODE5
004635'             302                         .BYTE BIT1+MODE6                ;57  RESERVED+MODE6
                005156'                 .=RX.CONTROL.TABLE+3000         ;MODE6
                005236'                 .=.+60
005236'             312                         .BYTE BIT3+BIT1+MODE6           ;60  INC/BCC+STORE+MODE6
005237'             330                         .BYTE BIT4+BIT3+MODE6           ;61  DSCARD+INC/BCC+MODE6
005240'             312                         .BYTE BIT3+BIT1+MODE6           ;62  INC/BCC+STORE+MODE6
005241'             330                         .BYTE BIT4+BIT3+MODE6           ;63  DSCARD+INC/BCC+MODE6
005242'             312                         .BYTE BIT3+BIT1+MODE6           ;64  INC/BCC+STORE+MODE6
005243'             330                         .BYTE BIT4+BIT3+MODE6           ;65  DSCARD+INC/BCC+MODE6
005244'             312                         .BYTE BIT3+BIT1+MODE6           ;66  INC/BCC+STORE+MODE6
005245'             370                         .BYTE BIT4+BIT3+MODE7           ;67  DSCARD+INC/BCC+MODE7

                005556'                 .=RX.CONTROL.TABLE+3400         ;MODE7
                005646'                 .=.+70
005646'             352                         .BYTE BIT3+BIT1+MODE7           ;70  INC/BCC+RESERVED+MODE7
005647'             352                         .BYTE BIT3+BIT1+MODE7           ;71  INC/BCC+RESERVED+MODE7
005650'             352                         .BYTE BIT3+BIT1+MODE7           ;72  INC/BCC+RESERVED+MODE7
005651'             352                         .BYTE BIT3+BIT1+MODE7           ;73  INC/BCC+RESERVED+MODE7
005652'             352                         .BYTE BIT3+BIT1+MODE7           ;74  INC/BCC+RESERVED+MODE7
005653'             352                         .BYTE BIT3+BIT1+MODE7           ;75  INC/BCC+RESERVED+MODE7
005654'             354                         .BYTE BIT3+BIT2+MODE7           ;76  INC/BCC+EXP/BCC+MODE7
005655'             001                         .BYTE BIT0+MODE0                ;77  GEN/INT+MODE0
```

```
                006156'                 .=RX.CONTROL.TABLE+4000
006156'                                 TX.CONTROL.TABLE:

                006156'                 .=TX.CONTROL.TABLE+0            ;MODE 0
                006156'                 .=.+0
006156'             010                         .BYTE BIT3+MODE0                ;00  INC/BCC+MODE0
006157'             010                         .BYTE BIT3+MODE0                ;01  INC/BCC+MODE0
006160'             010                         .BYTE BIT3+MODE0                ;02  INC/BCC+MODE0
006161'             010                         .BYTE BIT3+MODE0                ;03  INC/BCC+MODE0
006162'             010                         .BYTE BIT3+MODE0                ;04  INC/BCC+MODE0
006163'             010                         .BYTE BIT3+MODE0                ;05  INC/BCC+MODE0
006164'             010                         .BYTE BIT3+MODE0                ;06  INC/BCC+MODE0
006165'             050                         .BYTE BIT3+MODE1                ;07  INC/BCC+MODE1
006166'             001                         .BYTE   BIT0+MODE0              ;10  RESERVED+MODE0
006167'             001                         .BYTE   BIT0+MODE0              ;11  RESERVED+MODE0
006170'             001                         .BYTE   BIT0+MODE0              ;12  RESERVED+MODE0
006171'             001                         .BYTE   BIT0+MODE0              ;13  RESERVED+MODE0
006172'             001                         .BYTE   BIT0+MODE0              ;14  RESERVED+MODE0
006173'             001                         .BYTE   BIT0+MODE0              ;15  RESERVED+MODE0
006174'             001                         .BYTE   BIT0+MODE0              ;16  RESERVED+MODE0
006175'             001                         .BYTE   BIT0+MODE0              ;17  RESERVED+MODE0
006176'             001                         .BYTE   BIT0+MODE0              ;20  RESERVED+MODE0
006177'             001                         .BYTE   BIT0+MODE0              ;21  RESERVED+MODE0
006200'             001                         .BYTE   BIT0+MODE0              ;22  RESERVED+MODE0
006201'             001                         .BYTE   BIT0+MODE0              ;23  RESERVED+MODE0
006202'             001                         .BYTE   BIT0+MODE0              ;24  RESERVED+MODE0
006203'             001                         .BYTE   BIT0+MODE0              ;25  RESERVED+MODE0
006204'             001                         .BYTE   BIT0+MODE0              ;26  RESERVED+MODE0
006205'             001                         .BYTE   BIT0+MODE0              ;27  RESERVED+MODE0
006206'         000074                  RXB10:  .BLKB    74       ;THIS IS RECEIVER BUFFER FOR DEVICE #1  LINE #0
006302'         000074                  RXB11:  .BLKB    74       ;THIS IS RECEIVER BUFFER FOR DEVICE #1  LINE #1
006376'         000074                  RXB12:  .BLKB    74       ;THIS IS RECEIVER BUFFER FOR DEVICE #1  LINE #2
006472'         000074                  RXB13:  .BLKB    74       ;THIS IS RECEIVER BUFFER FOR DEVICE #1  LINE #3
                                        .MEXIT
```



```
                006556'                 .=TX.CONTROL.TABLE+400          ;MODE1
                006566'                 .=.+10
006566'             050                         .BYTE BIT3+MODE1                ;10  INC/BCC+MODE1
006567'             050                         .BYTE BIT3+MODE1                ;11  INC/BCC+MODE1
006570'             050                         .BYTE BIT3+MODE1                ;12  INC/BCC+MODE1
006571'             050                         .BYTE BIT3+MODE1                ;13  INC/BCC+MODE1
006572'             050                         .BYTE BIT3+MODE1                ;14  INC/BCC+MODE1
006573'             050                         .BYTE BIT3+MODE1                ;15  INC/BCC+MODE1
006574'             050                         .BYTE BIT3+MODE1                ;16  INC/BCC+MODE1
006575'             110                         .BYTE BIT3+MODE2                ;17  INC/BCC+MODE2
006576'         000074                  RXB14:  .BLKB    74       ;THIS IS RECEIVER BUFFER FOR DEVICE #1  LINE #4
006672'         000074                  RXB15:  .BLKB    74       ;THIS IS RECEIVER BUFFER FOR DEVICE #1  LINE #5
006766'         000074                  RXB16:  .BLKB    74       ;THIS IS RECEIVER BUFFER FOR DEVICE #1  LINE #6
007062'         000074                  RXB17:  .BLKB    74       ;THIS IS RECEIVER BUFFER FOR DEVICE #1  LINE #7
                                        .MEXIT

                007156'                 .=TX.CONTROL.TABLE+1000         ;MODE2
                007176'                 .=.+20
007176'             111                         .BYTE BIT3+BIT0+MODE2           ;20  INC/BCC+RESERVED+MODE2
007177'             111                         .BYTE BIT3+BIT0+MODE2           ;21  INC/BCC+RESERVED+MODE2
007200'             111                         .BYTE BIT3+BIT0+MODE2           ;22  INC/BCC+RESERVED+MODE2
007201'             111                         .BYTE BIT3+BIT0+MODE2           ;23  INC/BCC+RESERVED+MODE2
007202'             111                         .BYTE BIT3+BIT0+MODE2           ;24  INC/BCC+RESERVED+MODE2
007203'             111                         .BYTE BIT3+BIT0+MODE2           ;25  INC/BCC+RESERVED+MODE2
007204'             111                         .BYTE BIT3+BIT0+MODE2           ;26  INC/BCC+RESERVED+MODE2
007205'             151                         .BYTE BIT3+BIT0+MODE3           ;27  INC/BCC+RESERVED+MODE3
007206'         000074                  RXB110: .BLKB    74       ;THIS IS RECEIVER BUFFER FOR DEVICE #1  LINE #10
007302'         000074                  RXB111: .BLKB    74       ;THIS IS RECEIVER BUFFER FOR DEVICE #1  LINE #11
007376'         000074                  RXB112: .BLKB    74       ;THIS IS RECEIVER BUFFER FOR DEVICE #1  LINE #12
007472'         000074                  RXB113: .BLKB    74       ;THIS IS RECEIVER BUFFER FOR DEVICE #1  LINE #13
                                        .MEXIT
```

```
          007556'                 .=TX.CONTROL.TABLE+1400          ;MODE3
          007606'                 .=.+30
                                          .BYTE BIT1+MODE3                 ;30  SND/DLE+MODE3
                                          .BYTE BIT1+MODE3                 ;31  SND/DLE+MODE3
                                          .BYTE BIT1+MODE3                 ;32  SND/DLE+MODE3
                                          .BYTE BIT1+MODE3                 ;33  SND/DLE+MODE3
                                          .BYTE BIT1+MODE3                 ;34  SND/DLE+MODE3
                                          .BYTE BIT1+MODE3                 ;35  SND/DLE+MODE3
                                          .BYTE   BIT0+MODE3               ;36  RESERVED+MODE3
                                          .BYTE BIT1+MODE4                 ;37  SND/DLE+MODE4
 007606'     142                  RXB114: .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #1  LINE #14
 007607'     142                  RXB115: .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #1  LINE #15
 007610'     142                  RXB116: .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #1  LINE #16
 007611'     142                  RXB117: .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #1  LINE #17
 007612'     142                  .MEXIT
 007613'     142
 007614'     141
 007615'     202
 007616' 000074
 007712' 000074
 010006' 000074
 010102' 000074


          010156'                 .=TX.CONTROL.TABLE+2000          ;MODE4
          010216'                 .=.+40
                                          .BYTE BIT3+BIT0+MODE4            ;40  INC/BCC+RESERVED+MODE4
                                          .BYTE BIT3+BIT0+MODE4            ;41  INC/BCC+RESERVED+MODE4
                                          .BYTE BIT3+BIT0+MODE4            ;42  INC/BCC+RESERVED+MODE4
                                          .BYTE BIT3+BIT0+MODE4            ;43  INC/BCC+RESERVED+MODE4
                                          .BYTE BIT3+BIT0+MODE4            ;44  INC/BCC+RESERVED+MODE4
                                          .BYTE BIT3+BIT0+MODE4            ;45  INC/BCC+RESERVED+MODE4
                                          .BYTE BIT3+BIT0+MODE4            ;46  INC/BCC+RESERVED+MODE4
                                          .BYTE BIT0+MODE5                 ;47  RESERVED+MODE5
 010216'     211                  RXB20:  .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #2  LINE #0
 010217'     211                  RXB21:  .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #2  LINE #1
 010220'     211                  RXB22:  .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #2  LINE #2
 010221'     211                  RXB23:  .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #2  LINE #3
 010222'     211                  .MEXIT
 010223'     211
 010224'     211
 010225'     241
 010226' 000074
 010322' 000074
 010416' 000074
 010512' 000074


          010556'                 .=TX.CONTROL.TABLE+2400          ;MODE5
          010626'                 .=.+50
                                          .BYTE BIT3+BIT0+MODE5            ;50  INC/BCC+RESERVED+MODE5
                                          .BYTE BIT3+BIT0+MODE5            ;51  INC/BCC+RESERVED+MODE5
                                          .BYTE BIT3+BIT0+MODE5            ;52  INC/BCC+RESERVED+MODE5
                                          .BYTE BIT3+BIT0+MODE5            ;53  INC/BCC+RESERVED+MODE5
                                          .BYTE BIT3+BIT0+MODE5            ;54  INC/BCC+RESERVED+MODE5
                                          .BYTE BIT3+BIT0+MODE5            ;55  INC/BCC+RESERVED+MODE5
                                          .BYTE BIT3+BIT0+MODE5            ;56  INC/BCC+RESERVED+MODE5
                                          .BYTE BIT0+MODE6                 ;57  RESERVED+MODE6
 010626'     251                  RXB24:  .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #2  LINE #4
 010627'     251                  RXB25:  .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #2  LINE #5
 010630'     251                  RXB26:  .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #2  LINE #6
 010631'     251                  RXB27:  .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #2  LINE #7
 010632'     251                  .MEXIT
 010633'     251
 010634'     251
 010635'     301
 010636' 000074
 010732' 000074
 011026' 000074
 011122' 000074
```



```
          011156'                 .=TX.CONTROL.TABLE+3000          ;MODE6
          011236'                 .=.+60
                                          .BYTE BIT3+BIT0+MODE6            ;60  INC/BCC+RESERVED+MODE6
                                          .BYTE BIT3+BIT0+MODE6            ;61  INC/BCC+RESERVED+MODE6
                                          .BYTE BIT3+BIT0+MODE6            ;62  INC/BCC+RESERVED+MODE6
                                          .BYTE BIT3+BIT0+MODE6            ;63  INC/BCC+RESERVED+MODE6
                                          .BYTE BIT3+BIT0+MODE6            ;64  INC/BCC+RESERVED+MODE6
                                          .BYTE BIT3+BIT0+MODE6            ;65  INC/BCC+RESERVED+MODE6
                                          .BYTE BIT3+BIT0+MODE6            ;66  INC/BCC+RESERVED+MODE6
                                          .BYTE BIT3+BIT0+MODE7            ;67  INC/BCC+RESERVED+MODE7
 011236'     311                  RXB210: .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #2  LINE #10
 011237'     311                  RXB211: .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #2  LINE #11
 011240'     311                  RXB212: .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #2  LINE #12
 011241'     311                  RXB213: .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #2  LINE #13
 011242'     311                  .MEXIT
 011243'     311
 011244'     311
 011245'     351
 011246' 000074
 011342' 000074
 011436' 000074
 011532' 000074


          011556'                 .=TX.CONTROL.TABLE+3400          ;MODE7
          011646'                 .=.+70
                                          .BYTE BIT3+BIT0+MODE7            ;70  INC/BCC+RESERVED+MODE7
                                          .BYTE BIT3+BIT0+MODE7            ;71  INC/BCC+RESERVED+MODE7
                                          .BYTE BIT3+BIT0+MODE7            ;72  INC/BCC+RESERVED+MODE7
                                          .BYTE BIT3+BIT0+MODE7            ;73  INC/BCC+RESERVED+MODE7
                                          .BYTE BIT3+BIT0+MODE7            ;74  INC/BCC+RESERVED+MODE7
                                          .BYTE BIT3+BIT0+MODE7            ;75  INC/BCC+RESERVED+MODE7
                                          .BYTE BIT3+BIT2+MODE7            ;76  INC/BCC+SND/BCC+MODE7
                                          .BYTE BIT0+MODE0                 ;77  RESERVED+MODE0
 011646'     351                  RXB214: .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #2  LINE #14
 011647'     351                  RXB215: .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #2  LINE #15
 011650'     351                  RXB216: .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #2  LINE #16
 011651'     351                  RXB217: .BLKB   74       ;THIS IS RECEIVER BUFFER FOR DEVICE #2  LINE #17
 011652'     351                  .MEXIT
 011653'     351
 011654'     354
 011655'     001
 011656' 000074
 011752' 000074
 012046' 000074
 012142' 000074
```

```
012236'                                 TXISR:
012236' 010046                                  MOV     R0,-(SP)        ;SAVE R0 ON THE STACK
012240' 010446                                  MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
012242' 012500                                  MOV     (R5)+,R0        ;GET DEVICE CSR ON INTERUPT
012244' 032710  002000                          BIT     #OVRFLO,@DVSCR  ;OVERFLOW?
012250' 001002                                  BNE     2$              ;REPORT OVERFLOW
012252' 005710                          1$:     TST     @DVSCR          ;IS IS=1?
012254' 100407                                  BMI     3$              ;BR IF YES
012256' 010067  165616                  2$:     MOV     R0,CSRA         ;LOAD FOR PRINT OUT
012262' 011067  165616                          MOV     @DVSCR,ASTAT    ;GET BAD LOC SET UP
012266' 010067  165610                          MOV     R0,ACSR         ;SET FOR TYPE OUT
012272' 000447                                  BR 65$                  ;LEAVE ROUTINE
012274' 016004  000014                  3$:     MOV     DVNSR(R0),R4    ;GET DVNSR REGISTER
012300' 110405                                  MOVB    R4,R5           ;GET LINE NUMBER
012302' 042705  177400                          BIC     #^C<377>,R5     ;CLEAR ANY SIGN EXTEND
012306' 052705  100400                          BIS     #BIT15+BIT8,R5  ;SET 1 PROBABILITY OF GOOD RESULTS
012312' 020504                                  CMP     R5,R4           ;ARE THEY EQUAL?
012314' 001416                                  BEQ     4$              ;BR IF YES
012316' 052705  001000                          BIS     #BIT9,R5        ;GET 2ND CHOICE OF GOOD RESULTS
012322' 020504                                  CMP     R5,R4           ;NOW ARE THESE OK?
012324' 001412                                  BEQ     4$              ;BR IF YES
012326' 010067  165546                          MOV     R0,CSRA         ;SAVE FOR PRINTOUT
012332' 010067  165544                          MOV     R0,ACSR         ;SET FOR ERROR
012336' 062767  000014  165536                  ADD     #14,ACSR        ;POINTER TO DVNSR
012344' 010467  165534                          MOV     R4,ASTAT        ;GET BAD RESULTS
012350' 000420                                  BR 65$                  ;LEAVE ROUTINE
012352' 010460  000006                  4$:     MOV     R4,DVSRS(R0)    ;LOAD LINE NUMBER
012356' 012760  177740  000010                  MOV     #-40,DVSRA(R0)  ;SET FOR 40(8) CHARS
012364' 042710  000060                          BIC     #BIT5+BIT4,@DVSCR
                                                                        ;CLEAR EA BITS
012370' 105360  000007                          DECB    DVSRSH(R0)      ;IS THIS PRI OR ALT?
012374' 001014                                  BNE     5$              ;BR IF IT WAS ALT
012376' 156710  000302                          BISB    PRI.EA,@DVSCR   ;SET EA BITS
012402' 016760  000300  000010                  MOV     PRI.PA,DVSRA(R0);LOAD PRI
012410' 000413                                  BR      6$              ;GET OUT
012412' 052710  004000                  65$:    BIS     #MRESET,@DVSCR  ;RESET DEVICE ON ERROR
012416' 012767  000001  167456                  MOV     #+1,ABORT       ;SET ERROR FLAG
012424' 000405                                  BR 6$
012426' 156710  000253                  5$:     BISB    ALT.EA,@DVSCR   ;SET EA BITS
012432' 016760  000252  000010                  MOV     ALT.PA,DVSRA(R0) ;LOAD ALTERNATE
012440'                                 6$:
012440' 012604                                  MOV     (SP)+,R4        ;POP STACK TO R4
012442' 012600                                  MOV     (SP)+,R0        ;POP STACK TO R0
012444' 012605                                  MOV     (SP)+,R5        ;POP STACK TO R5
012446' 000002                                  RTI
```



```
012450'                                 RXISR:
012450' 010046                                  MOV     R0,-(SP)        ;SAVE R0 ON THE STACK
012452' 010146                                  MOV     R1,-(SP)        ;SAVE R1 ON THE STACK
012454' 012500                                  MOV     (R5)+,R0        ;GET DVSCR FOR ISR USE
012456' 052710  000000                          BIS     #0,@DVSCR       ;-NOP- (DATAIP BANG DV11 REGISTER)
012462' 016001  000002                          MOV     DVRIC(R0),R1    ;GET DVRIC REGISTER
012466' 105710                                  TSTB    @DVSCR          ;DVSCR BIT7=1?
012470' 100407                                  BMI     1$              ;BR IF YES
012472' 010067  165402                          MOV     R0,CSRA         ;SAVE FOR TYPE OUT
012476' 011067  165402                          MOV     @DVSCR,ASTAT    ;GET BAD DATA
012502' 010067  165374                          MOV     R0,ACSR         ;
012506' 000462                                  BR      7$              ;BR IF ERROR
012510' 032701  170000                  1$:     BIT     #BIT15+BIT14+BIT13+BIT12,R1
                                                                        ;ARE ANY "ERROR CODES" SET?
012514' 001037                                  BNE     5$              ;BR IF YES (RCC S/B THE ONLY)
012516' 122701  000077                          CMPB    #77,R1
012522' 001043                                  BNE     6$
012524' 000301                          2$:     SWAB    R1              ;GET LINE IN LOW BYTE
012526' 042701  177400                          BIC     #<377*400>,R1   ;CLEAR HIGH BYTE
012532' 010160  000006                          MOV     R1,DVSRS(R0)    ;LOAD LINE NUMBER
012536' 112760  000013  000007                  MOVB    #13,DVSRSH(R0)  ;SEL LINE STATE REGISTER
012544' 042760  000004  000010                  BIC     #TX.GO,DVSRA(R0) ;CLEAR TX GO.
012552' 052760  000002  000010                  BIS     #RESYNC,DVSRA(R0) ; SET RX RESYNC
012560' 112760  000017  000007                  MOVB    #17,DVSRSH(R0)  ;SEL CNTRL BYTE STORE
012566' 052760  000020  000010                  BIS     #DSCARD,DVSRA(R0) ;THROW AWAY THE 77 CHAR
012574' 052710  000400                          BIS     #BIT8,@DVSCR    ;RESTART DV11 UCPU
012600' 062705  000010                          ADD     #10,R5          ;POP TO END PASS FLAG POINTER
012604' 006301                                  ASL     R1              ;MAKE EVEN (MULT BY 2).
012606' 056115  012712'                         BIS     EOP.TABLE(R1),(R5) ;SET END-OF-PASS FLAG.
012612' 000430                                  BR      9$              ;CONTINUE ALONG
012614' 010105                          5$:     MOV     R1,R5           ;LOAD LINE NUMBER INTO R5
012616' 042705  170377                          BIC     #^C<BIT11+BIT10+BIT9+BIT8>,R5
012622' 052705  050000                          BIS     #BIT14+BIT12,R5 ;MAKE IT EQUAL TO RCC INDICATOR
012626' 020501                                  CMP     R5,R1           ;IS RIC OK?
012630' 001417                                  BEQ     8$              ;BR IF YES
012632' 010067  165244                  6$:     MOV     R0,ACSR         ;SET FOR ERROR
012636' 062767  000002  165236                  ADD     #2,ACSR         ;POINT TO DVRIC
012644' 010167  165234                          MOV     R1,ASTAT        ;LOAD BAD RESULTS
012650' 010067  165224                          MOV     R0,CSRA         ;SET FOR PRINT OUT
012654' 012710  004000                  7$:     MOV     #MRESET,@DVSCR
012660' 012767  177777  167214                  MOV     #-1,ABORT       ;SET ERROR INDICATOR.
012666' 000402                                  BR      9$              ;CONT
012670' 052710  000400                  8$:     BIS     #BIT8,@DVSCR    ;RESTART DV11
012674'                                 9$:
012674' 012601                                  MOV     (SP)+,R1        ;POP STACK TO R1
012676' 012600                                  MOV     (SP)+,R0        ;POP STACK TO R0
012700' 012605                                  MOV     (SP)+,R5        ;POP STACK TO R5
012702' 000002                                  RTI                     ;LEAVE ISR
```

```
012704'     001                 PRI.EA: .BYTE   1
012705'     001                 ALT.EA: .BYTE   1
012706' 000001                  PRI.PA: .WORD   1
012710' 000001                  ALT.PA: .WORD   1

012712'                         EOP.TABLE:
012712' 000001                          ^B<0000000000000001>    ;LINE 0
012714' 000002                          ^B<0000000000000010>    ;LINE 1
012716' 000004                          ^B<0000000000000100>    ;LINE 2
012720' 000010                          ^B<0000000000001000>    ;LINE 3
012722' 000020                          ^B<0000000000010000>    ;LINE 4
012724' 000040                          ^B<0000000000100000>    ;LINE 5
012726' 000100                          ^B<0000000001000000>    ;LINE 6
012730' 000200                          ^B<0000000010000000>    ;LINE 7
012732' 000400                          ^B<0000000100000000>    ;LINE 8
012734' 001000                          ^B<0000001000000000>    ;LINE 9
012736' 002000                          ^B<0000010000000000>    ;LINE 10
012740' 004000                          ^B<0000100000000000>    ;LINE 11
012742' 010000                          ^B<0001000000000000>    ;LINE 12
012744' 020000                          ^B<0010000000000000>    ;LINE 13
012746' 040000                          ^B<0100000000000000>    ;LINE 14
012750' 100000                          ^B<1000000000000000>    ;LINE 15
```



```
                                ;       THIS IS A SAMPLE OF WHAT IS LOADED INTO
                                ;       THE DV11 SECONDARY REGISTER FOR THE LINES
                                ;       THAT WERE SELECTED TO RUN.
                                ;       HOWEVER THE BUS ADDRESS AND CHAR CNTS MAY CHANGE.
                                ;
012752' 013106'                 FUNC.X: TXBAP                   ;00     TX PRI BUS ADDR.
012754' 177740                          -40                     ;01     TX PRI BYTE CNT
012756' 013146'                         TXBAS                   ;02     TX ALT BUS ADDR.
012760' 177740                          -40                     ;03     TX ALT BYTE CNT
012762' 000000                          RXB                     ;04     RX BUS ADDR.
012764' 177703                          -75                     ;05     RX BYTE CNT
012766' 000000                          0                       ;06     TX BCC REG
012770' 000000                          0                       ;07     RX BCC REG
012772' 006156'                         TX.CONTROL.TABLE        ;10     TX CNTRL TABLE
012774' 002156'                         RX.CONTROL.TABLE        ;11     RX CNTRL TABLE
012776' 014013                          DLE*HIBYTE*CRC16+STRIP.SYNC+IDLE.MARK
                                                                ;12     TX DLE REGISTER  STRIP LEADING SYNC; IDLE MARK
013000' 000004                          TX.GO                   ;13     LINE STATE REGISTER
013002' 000000                          MODE0                   ;14     TX MODE BIT REG
013004' 000000                          MODE0                   ;15     RX MODE BIT REG
013006' 000000                          0                       ;16     LINE PROTOCOL REG
013010' 000000                          0                       ;17     RX CONTROL BYTE STORAGE

                                ;       THIS IS AN EXACT IMAGE
                                ;       OF WHAT DATA SHOULD BE FOUND IN ANY COMPLETED
                                ;       DV11 RECEIVER BUFFER.   (NOTE: DLE=30(8))
                                ;
        013012'                 DATA.TABLE:
        013012'     000     001     002         .BYTE   0,1,2,3,4,5,6
        013021'     010     011     012         .BYTE   10,11,12,13,14,15,16
        013030'     020     022     024         .BYTE   20,22,24,26
        013034'     030     030     030         .BYTE   DLE,30,DLE,31,DLE,32,DLE,33,DLE,34,DLE,35,36,DLE,37
        013053'     040     041     042         .BYTE   40,41,42,43,44,45,46,47
        013063'     050     051     052         .BYTE   50,51,52,53,54,55,56,57
        013073'     060     062     064         .BYTE   60,62,64,66
        013077'     070     071     072         .BYTE   70,71,72,73,74,75,76;-

                                ;       THIS THE DATA THAT THE TRANSMITTERS SEND.
                                ;
(1)     013106'                 TXBAP:
(2)     013106'     000     001     002         .BYTE   0,1,2,3,4,5,6,7
(2)     013116'     010     011     012         .BYTE   10,11,12,13,14,15,16,17
(2)     013126'     020     021     022         .BYTE   20,21,22,23,24,25,26,27
(2)     013136'     030     031     032         .BYTE   30,31,32,33,34,35,36,37
(1)     013146'                 TXBAS:
(2)     013146'     040     041     042         .BYTE   40,41,42,43,44,45,46,47
(2)     013156'     050     051     052         .BYTE   50,51,52,53,54,55,56,57
(2)     013166'     060     061     062         .BYTE   60,61,62,63,64,65,66,67
(2)     013176'     070     071     072         .BYTE   70,71,72,73,74,75,76,77
                                .EVEN
```

```
                                        ;RECEIVER BUFFERS FOR DEVICES #3 AND #4.
                                        ;
013206' 000074                          RXB30:  .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #3  LINE #0
013302' 000074                          RXB31:  .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #3  LINE #1
013376' 000074                          RXB32:  .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #3  LINE #2
013472' 000074                          RXB33:  .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #3  LINE #3
                                        .MEXIT
013566' 000074                          RXB34:  .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #3  LINE #4
013662' 000074                          RXB35:  .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #3  LINE #5
013756' 000074                          RXB36:  .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #3  LINE #6
014052' 000074                          RXB37:  .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #3  LINE #7
                                        .MEXIT
014146' 000074                          RXB310: .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #3  LINE #10
014242' 000074                          RXB311: .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #3  LINE #11
014336' 000074                          RXB312: .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #3  LINE #12
014432' 000074                          RXB313: .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #3  LINE #13
                                        .MEXIT
014526' 000074                          RXB314: .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #3  LINE #14
014622' 000074                          RXB315: .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #3  LINE #15
014716' 000074                          RXB316: .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #3  LINE #16
015012' 000074                          RXB317: .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #3  LINE #17
                                        .MEXIT
015106' 000074                          RXB40:  .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #4  LINE #0
015202' 000074                          RXB41:  .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #4  LINE #1
015276' 000074                          RXB42:  .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #4  LINE #2
015372' 000074                          RXB43:  .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #4  LINE #3
                                        .MEXIT
015466' 000074                          RXB44:  .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #4  LINE #4
015562' 000074                          RXB45:  .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #4  LINE #5
015656' 000074                          RXB46:  .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #4  LINE #6
015752' 000074                          RXB47:  .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #4  LINE #7
                                        .MEXIT
016046' 000074                          RXB410: .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #4  LINE #10
016142' 000074                          RXB411: .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #4  LINE #11
016236' 000074                          RXB412: .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #4  LINE #12
016332' 000074                          RXB413: .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #4  LINE #13
                                        .MEXIT
016426' 000074                          RXB414: .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #4  LINE #14
016522' 000074                          RXB415: .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #4  LINE #15
016616' 000074                          RXB416: .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #4  LINE #16
016712' 000074                          RXB417: .BLKB   74      ;THIS IS RECEIVER BUFFER FOR DEVICE #4  LINE #17
                                        .MEXIT
```



```
017006'                                 LNKTAB:
                                        ;       ALL INTERUPTS WILL BE CHANNELED THROUGH
                                        ;       THE APPROIATE CALL.
                                        ;
017006' 004567  173436                  JSR     R5,RXISR        ;GOTO RECEIVER ISR FOR DEVICE #1.
017012' 000000  006206'                 .WORD   $DVSCR,RXB10
017016' 004567  173214                  JSR     R5,TXISR        ;GOTO TRANSMITTER ISR FOR DEVICE #1
017022' 000000                          .WORD   DVSCR
017024' 000224'                         XXX1:   .WORD   LINES1  ;END PASS FLAG.

017026' 004567  173416                  JSR     R5,RXISR        ;GOTO RECEIVER ISR FOR DEVICE #2.
017032' 000000  010226'                 .WORD   $DVSCR,RXB20
017036' 004567  173174                  JSR     R5,TXISR        ;GOTO TRANSMITTER ISR FOR DEVICE #2
017042' 000000                          .WORD   DVSCR
017044' 000236'                         XXX2:   .WORD   LINES2  ;END PASS FLAG.

017046' 004567  173376                  JSR     R5,RXISR        ;GOTO RECEIVER ISR FOR DEVICE #3.
017052' 000000  013206'                 .WORD   $DVSCR,RXB30
017056' 004567  173154                  JSR     R5,TXISR        ;GOTO TRANSMITTER ISR FOR DEVICE #3
017062' 000000                          .WORD   DVSCR
017064' 000250'                         XXX3:   .WORD   LINES3  ;END PASS FLAG.

017066' 004567  173356                  JSR     R5,RXISR        ;GOTO RECEIVER ISR FOR DEVICE #4.
017072' 000000  015106'                 .WORD   $DVSCR,RXB40
017076' 004567  173134                  JSR     R5,TXISR        ;GOTO TRANSMITTER ISR FOR DEVICE #4
017102' 000000                          .WORD   DVSCR
017104' 000262'                         XXX4:   .WORD   LINES4  ;END PASS FLAG.

                                        .MEXIT
017106'                                 BUFFER.TABLE:
017106' 006206'                         RXB10           ;BUFFER FOR DEVICE #1 LINE #0
017110' 006302'                         RXB11           ;BUFFER FOR DEVICE #1 LINE #1
017112' 006376'                         RXB12           ;BUFFER FOR DEVICE #1 LINE #2
017114' 006472'                         RXB13           ;BUFFER FOR DEVICE #1 LINE #3
017116' 006576'                         RXB14           ;BUFFER FOR DEVICE #1 LINE #4
017120' 006672'                         RXB15           ;BUFFER FOR DEVICE #1 LINE #5
017122' 006766'                         RXB16           ;BUFFER FOR DEVICE #1 LINE #6
017124' 007062'                         RXB17           ;BUFFER FOR DEVICE #1 LINE #7
017126' 007206'                         RXB110          ;BUFFER FOR DEVICE #1 LINE #10
017130' 007302'                         RXB111          ;BUFFER FOR DEVICE #1 LINE #11
017132' 007376'                         RXB112          ;BUFFER FOR DEVICE #1 LINE #12
017134' 007472'                         RXB113          ;BUFFER FOR DEVICE #1 LINE #13
017136' 007616'                         RXB114          ;BUFFER FOR DEVICE #1 LINE #14
017140' 007712'                         RXB115          ;BUFFER FOR DEVICE #1 LINE #15
017142' 010006'                         RXB116          ;BUFFER FOR DEVICE #1 LINE #16
017144' 010102'                         RXB117          ;BUFFER FOR DEVICE #1 LINE #17

017146' 010226'                         RXB20           ;BUFFER FOR DEVICE #2 LINE #0
017150' 010322'                         RXB21           ;BUFFER FOR DEVICE #2 LINE #1
017152' 010416'                         RXB22           ;BUFFER FOR DEVICE #2 LINE #2
017154' 010512'                         RXB23           ;BUFFER FOR DEVICE #2 LINE #3
017156' 010636'                         RXB24           ;BUFFER FOR DEVICE #2 LINE #4
017160' 010732'                         RXB25           ;BUFFER FOR DEVICE #2 LINE #5
017162' 011026'                         RXB26           ;BUFFER FOR DEVICE #2 LINE #6
017164' 011122'                         RXB27           ;BUFFER FOR DEVICE #2 LINE #7
017166' 011246'                         RXB210          ;BUFFER FOR DEVICE #2 LINE #10
```

```
017170' 011342'                         RXB211          ;BUFFER FOR DEVICE #2 LINE #11
017172' 011436'                         RXB212          ;BUFFER FOR DEVICE #2 LINE #12
017174' 011532'                         RXB213          ;BUFFER FOR DEVICE #2 LINE #13
017176' 011656'                         RXB214          ;BUFFER FOR DEVICE #2 LINE #14
017200' 011752'                         RXB215          ;BUFFER FOR DEVICE #2 LINE #15
017202' 012046'                         RXB216          ;BUFFER FOR DEVICE #2 LINE #16
017204' 012142'                         RXB217          ;BUFFER FOR DEVICE #2 LINE #17

017206' 013206'                         RXB30           ;BUFFER FOR DEVICE #3 LINE #0
017210' 013302'                         RXB31           ;BUFFER FOR DEVICE #3 LINE #1
017212' 013376'                         RXB32           ;BUFFER FOR DEVICE #3 LINE #2
017214' 013472'                         RXB33           ;BUFFER FOR DEVICE #3 LINE #3
017216' 013566'                         RXB34           ;BUFFER FOR DEVICE #3 LINE #4
017220' 013662'                         RXB35           ;BUFFER FOR DEVICE #3 LINE #5
017222' 013756'                         RXB36           ;BUFFER FOR DEVICE #3 LINE #6
017224' 014052'                         RXB37           ;BUFFER FOR DEVICE #3 LINE #7
017226' 014146'                         RXB310          ;BUFFER FOR DEVICE #3 LINE #10
017230' 014242'                         RXB311          ;BUFFER FOR DEVICE #3 LINE #11
017232' 014336'                         RXB312          ;BUFFER FOR DEVICE #3 LINE #12
017234' 014432'                         RXB313          ;BUFFER FOR DEVICE #3 LINE #13
017236' 014526'                         RXB314          ;BUFFER FOR DEVICE #3 LINE #14
017240' 014622'                         RXB315          ;BUFFER FOR DEVICE #3 LINE #15
017242' 014716'                         RXB316          ;BUFFER FOR DEVICE #3 LINE #16
017244' 015012'                         RXB317          ;BUFFER FOR DEVICE #3 LINE #17

017246' 015106'                         RXB40           ;BUFFER FOR DEVICE #4 LINE #0
017250' 015202'                         RXB41           ;BUFFER FOR DEVICE #4 LINE #1
017252' 015276'                         RXB42           ;BUFFER FOR DEVICE #4 LINE #2
017254' 015372'                         RXB43           ;BUFFER FOR DEVICE #4 LINE #3
017256' 015466'                         RXB44           ;BUFFER FOR DEVICE #4 LINE #4
017260' 015562'                         RXB45           ;BUFFER FOR DEVICE #4 LINE #5
017262' 015656'                         RXB46           ;BUFFER FOR DEVICE #4 LINE #6
017264' 015752'                         RXB47           ;BUFFER FOR DEVICE #4 LINE #7
017266' 016046'                         RXB410          ;BUFFER FOR DEVICE #4 LINE #10
017270' 016142'                         RXB411          ;BUFFER FOR DEVICE #4 LINE #11
017272' 016236'                         RXB412          ;BUFFER FOR DEVICE #4 LINE #12
017274' 016332'                         RXB413          ;BUFFER FOR DEVICE #4 LINE #13
017276' 016426'                         RXB414          ;BUFFER FOR DEVICE #4 LINE #14
017300' 016522'                         RXB415          ;BUFFER FOR DEVICE #4 LINE #15
017302' 016616'                         RXB416          ;BUFFER FOR DEVICE #4 LINE #16
017304' 016712'                         RXB417          ;BUFFER FOR DEVICE #4 LINE #17

                                .MEXIT
        000001                          .END
```



```
.ABORT   002102R   333*   518    520    609#   910*   959*
ACSR     000102R   260#   884*   896*   897*   929*   954*   955*
ADDR     000006R   226#   344    370    479    559
ADDR22=  001000    273#
ALT.EA   012705R   489*   912    968#
ALT.PA   012710R   490*   913    970#
ASB      000106R   264#   594*
ASTAT    000104R   262#   883*   898*   928*   956*
AWAS     000110R   265#   595*
BEGIN    000000R   223#   337    488    497    516    517    531    548    583    597    607    608
BIT0   = 000001    217    273#   377    389    398    491    727    742    743    744    745    746    747
                   748    749    750    751    752    753    754    755    756    757    781    782    783
                   784    785    786    787    788    802    813    814    815    816    817    818    819
                   820    829    830    831    832    833    834    835    836    844    845    846    847
                   848    849    850    851    860    861    862    863    864    865    867
BIT1   = 000002    217    273#   398    666    668    670    672    676    677    678    679    680    681
                   682    683    687    688    689    690    691    692    693    694    698    699    700
                   701    702    703    704    705    708    710    712    714    720    721    722    723
                   724    725    796    797    798    799    800    801    803
BIT10  = 002000    217    273#   398    448    451    950
BIT11  = 004000    217    273#   398    445    451    950
BIT12  = 010000    217    273#   445    448    931    951
BIT13  = 020000    217    273#   442    448    931
BIT14  = 040000    273#   448    931    951
BIT15  = 100000    273#   421    442    445    448    451    889    931
BIT2   = 000004    273#   398    726    866
BIT3   = 000010    217    273#   398    404    627    628    629    630    631    632    633    634    655
                   656    657    658    659    660    661    662    666    667    668    669    670    671
                   672    673    687    688    689    690    691    692    693    698    699    700    701
                   702    703    704    708    709    710    711    712    713    714    715    720    721
                   722    723    724    725    726    734    735    736    737    738    739    740    741
                   765    766    767    768    769    770    771    772    781    782    783    784    785
                   786    787    788    813    814    815    816    817    818    819    829    830    831
                   832    833    834    835    844    845    846    847    848    849    850    851    860
                   861    862    863    864    865    866
BIT4   = 000020    217    273#   441    498    634    635    636    637    638    639    640    641    642
                   643    644    645    646    647    648    649    650    662    667    669    671    673
                   709    711    713    715    902
BIT5   = 000040    273#   498    902
BIT6   = 000100    217    273#
BIT7   = 000200    273#
BIT8   = 000400    273#   398    889    944    950    961
BIT9   = 001000    273#   398    445    451    892    950
BREAK$= 104407     273#   516    517
BR1      000012R   228#   358
BR2      000013R   229#   362
BTOD$ = 104421     273#
BUFFER   017106R   375    560    1084#
BUFPNT   002104R   560*   565*   568    573*   610#
CDATA$= 104412     273#
CONFIG   000056R   248#
CONT     000362R   335    339#
CRC16  = 000010    217#   1005
CSRA     000100R   258#   559*   572*   882*   895*   927*   957*
DATA.T   013012R   589    1012#
DATCK$= 104411     273#
```

```
DATER$= 104404          273#    597
DEBUG = ****** U       1059
DLE   = 000030         1005    1012#
DROP    000356R         336#    340     342
DSCARD= 000020          217#    943
DVID1   000014R         230#    334     480
DVLCR = 000004          301#    421*    422     442*    443     445*    446     448*    449     451*    452
DVNSR = 000014          306#    886
DVRES = 000016          307#
DVRIC = 000002          300#    924
DVSFR = 000012          305#
DVSRA = 000010          304#    397*    424*    455*    463     466     469     473*    475*    476*    501*    901*    907*
                        913*    940*    941*    943*
DVSRS = 000006          302#    398*    419*    428*    900*    938*
DVSRSH= 000007          303#    420*    425*    426     429*    454*    459*    462*    465*    468*    474*    904*    939*
                        942*
EA      000302R         321#    471     489     500
ENDIT$= 104413          273#    583
END$  = 104410          273#    337     608
EOP.TA  012712R         947     972#
ERRTYP  000106R         263#    529*    546*
EXIT$ = 104400          273#
FUNC.X  012752R         418     995#
GETPA$= 104415          273#    488     497
GWBUF$= 104414          273#
HIBYTE= 000400          217#    1005
HOLD    000310R         324#    375*    391*    458
HRDCNT  000044R         243#
HRDER$= 104405          273#    531     548
HRDPAS  000050R         245#
HUNG    002106R         607     611#
ICONT   000036R         240#
ICOUNT  000040R         241#
IDLE.M= 000001          217#    1005
IDNUM   000122R         270#
INIT    000030R         237#
INTR    000120R         269#    331*
LINES1  000224R         275#    346     371     551     562     1063
LINES2  000236R         281#    347     553     1069
LINES3  000250R         287#    348     555     1075
LINES4  000262R         293#    349     557     1081
LNKTAB  017006R         345     1055#
MAP22$= 104416          273#
MODE0 = 000000          309#    627     628     629     630     631     632     633     635     636     637     638     639
                        640     641     642     643     644     645     646     647     648     649     650     727     734
                        735     736     737     738     739     740     742     743     744     745     746     747     748
                        749     750     751     752     753     754     755     756     757     867     1008    1009
MODE1 = 000040          310#    634     655     656     657     658     659     660     661     741     765     766     767
                        768     769     770     771
MODE2 = 000100          311#    662     666     667     668     669     670     671     672     772     781     782     783
                        784     785     786     787
MODE3 = 000140          312#    673     676     677     678     679     680     681     682     788     796     797     798
                        799     800     801     802
MODE4 = 000200          313#    683     687     688     689     690     691     692     693     803     813     814     815
                        816     817     818     819
MODE5 = 000240          314#    694     698     699     700     701     702     703     704     820     829     830     831
```



```
                        832     833     834     835
MODE6 = 000300          315#    705     708     709     710     711     712     713     714     836     844     845     846
                        847     848     849     850
MODE7 = 000340          316#    715     720     721     722     723     724     725     726     851     860     861     862
                        863     864     865     866
MODNAM  000000R         224#
MODSP   000224R         238     271#
MRESET= 004000          217#    396     909     958
MSGN$ = 104403          273#    607
MSGS$ = 104402          273#
MSGS  = 104401          273#
M.MODE= 014000          217#    421
NPR.IE= 020000          217#    491
NULL  = 000000          273#    531     548
OPEN  = 000000          225     231     232     233     234     251     252     253     254     255     256     257     258
                        260     262     264     265     267     268     269     273#
OTOA$ = 104420          273#
OVRFLO= 002000          217#    878
PA      000300R         320#    472     490     501
PASCNT  000034R         239#
PING.P  000274R         318#    332*    364     372     379     402     563     582*
PIRQ$ = 000004          273#
POPSP = 005726          273#
POPSP2= 022626          273#
PRI.EA  012704R         471*    906     967#
PRI.PA  012706R         472*    907     969#
PRTY  = 000000          273#
PRTY0 = 000000          273#
PRTY1 = 000040          273#
PRTY2 = 000100          273#
PRTY3 = 000140          273#
PRTY4 = 000200          273#
PRTY5 = 000240          228     229     273#
PRTY6 = 000300          273#
PRTY7 = 000340          273#
PS    = 177776          273#
PSW   = 177776          273#
PUSH  = 005746          273#
PUSH2 = 024646          273#
RAND$ = 104417          273#
RANNUM  000054R         247#
RENABL= 020000          217#    421
RESTRT  000342R         266     333#    550     587
RESYNC= 000002          217#    941
RES1    000056R         249#
RES2    000060R         250#
RSTRT   000112R         266#
RXB   = 000000          999     1012#
RXB10   006206R         758#    1060    1085
RXB11   006302R         759#    1086
RXB110  007206R         789#    1093
RXB111  007302R         790#    1094
RXB112  007376R         791#    1095
RXB113  007472R         792#    1096
RXB114  007616R         804#    1097
RXB115  007712R         805#    1098
```

```
RXB116  010006R          806#   1099
RXB117  010102R          807#   1100
RXB12   006376R          760#   1087
RXB13   006472R          761#   1088
RXB14   006576R          773#   1089
RXB15   006672R          774#   1090
RXB16   006766R          775#   1091
RXB17   007052R          776#   1092
RXB20   010226R          821#   1066    1102
RXB21   010322R          822#   1103
RXB210  011246R          852#   1110
RXB211  011342R          853#   1111
RXB212  011436R          854#   1112
RXB213  011532R          855#   1113
RXB214  011656R          868#   1114
RXB215  011752R          869#   1115
RXB216  012046R          870#   1116
RXB217  012142R          871#   1117
RXB22   010416R          823#   1104
RXB23   010512R          824#   1105
RXB24   010636R          837#   1106
RXB25   010732R          838#   1107
RXB26   011026R          839#   1108
RXB27   011122R          840#   1109
RXB30   013206R         1015#   1072    1119
RXB31   013302R         1016#   1120
RXB310  014146R         1025#   1127
RXB311  014242R         1026#   1128
RXB312  014336R         1027#   1129
RXB313  014432R         1028#   1130
RXB314  014526R         1030#   1131
RXB315  014622R         1031#   1132
RXB316  014716R         1032#   1133
RXB317  015012R         1033#   1134
RXB32   013376R         1017#   1121
RXB33   013472R         1018#   1122
RXB34   013566R         1020#   1123
RXB35   013662R         1021#   1124
RXB36   013756R         1022#   1125
RXB37   014052R         1023#   1126
RXB40   015106R         1035#   1078    1136
RXB41   015202R         1036#   1137
RXB410  016046R         1045#   1144
RXB411  016142R         1046#   1145
RXB412  016236R         1047#   1146
RXB413  016332R         1048#   1147
RXB414  016426R         1050#   1148
RXB415  016522R         1051#   1149
RXB416  016616R         1052#   1150
RXB417  016712R         1053#   1151
RXB42   015276R         1037#   1138
RXB43   015372R         1038#   1139
RXB44   015466R         1040#   1140
RXB45   015562R         1041#   1141
RXB46   015656R         1042#   1142
RXB47   015752R         1043#   1143
```



```
RXISR   012450R          919#   1059    1065    1071    1077
RX.CON  002156R          440     620#    624     625     653     664     674     685     696     706     718     729     1004
RX.IE = 000100           217#    491
SBADR   000102R          259#    592*
SCANNE  001532R          484     512#
SECTAB= 000200           217#
SELECT  000304R          322#    334*    339     341     382*    480*    481     569*
SETUP1  000372R          341#
SETUP2  000552R          352     370#
SET.EA  001504R          431     461     464     467     470     496#
SOFCNT  000042R          242#
SOFERS= 104406           273#
SOFPAS  000046R          244#
SPOINT  000032R          238#
SPSIZ = 000040             1#    271
SR1     000016R          231#
SR2     000020R          232#
SR3     000022R          233#
SR4     000024R          234#
START   000314R          237     329#
STAT    000026R          236#
STORE   000312R          325#    432*    433*    434*    435*    438*    439*    440*    441*    456*    457*    458*    460
STORE.= 010000           217#    491
STRIP.= 000002           217#   1005
SVR0    000062R          251#
SVR1    000064R          252#
SVR2    000066R          253#
SVR3    000070R          254#
SVR4    000072R          255#
SVR5    000074R          256#
SVR6    000076R          257#
SYNC11  000226R          276#
SYNC12  000230R          277#
SYNC13  000232R          278#
SYNC14  000234R          279#
SYNC21  000240R          282#
SYNC22  000242R          283#
SYNC23  000244R          284#
SYNC24  000246R          285#
SYNC31  000252R          288#
SYNC32  000254R          289#
SYNC33  000256R          290#
SYNC34  000260R          291#
SYNC41  000264R          294#
SYNC42  000266R          295#
SYNC43  000270R          296#
SYNC44  000272R          297#
SYSCNT  000052R          246#
TRPDFD= 000022           273#
TXBAP   013106R          995    1012#
TXBAS   013146R          487     997    1012#
TXISR   012236R          874#   1061    1067    1073    1079
TX.CON  006156R          434     730#    732     763     779     794     811     827     842     858     1003
TX.GO = 000004           217#    940    1007
VA      000276R          319#    430*    460*    463*    466*    469*    487*    488     497
VECTOR  000010R          227#    343
```

```
WASADR  000104R     261#    593*
WDFR    000116R     268#    330*
WDTO    000114R     267#    329*
XCNT    000306R     323#    405*    411*    414*
XFLAG   000005R     225#
XHUNG   002112R     611     613#
XXX1    017024R     551    1063#
XXX2    017044R     553    1069#
XXX3    017064R     555    1075#
XXX4    017104R     557    1081#
X0      = 000100   1012#
X1      = 000071   1012#
X2      = 000072   1012#
X3      = 000073   1012#
X4      = 000074   1012#
X5      = 000075   1012#
X6      = 000076   1012#
X7      = 000077   1012#
$DVSCR= 000000      217#    1060    1066    1072    1078
$F      = 000000    758#    759#    760#    761#    762#    773#    774#    775#    776#    777#    789#    790#    791#
                    792#    793#    804#    805#    806#    807#    808#    821#    822#    823#    824#    825#    837#
                    838#    839#    840#    841#    852#    853#    854#    855#    856#    868#    869#    870#    871#
                    872#   1015#   1016#   1017#   1018#   1019#   1020#   1021#   1022#   1023#   1024#   1025#   1026#
                   1027#   1028#   1029#   1030#   1031#   1032#   1033#   1034#   1035#   1036#   1037#   1038#   1039#
                   1040#   1041#   1042#   1043#   1044#   1045#   1046#   1047#   1048#   1049#   1050#   1051#   1052#
                   1053#   1054#
$FIB    = 000005   1059#   1065#   1071#   1077#   1083#   1085#   1102#   1119#   1136#   1153#
$QFIG   = 000005    217#    758     759     760     761     773     774     775     776     789     790     791     792
                    804     805     806     807     808#    821     822     823     824     837     838     839     840
                    852     853     854     855     868     869     870     871     872#    1015    1016    1017    1018
                   1020    1021    1022    1023    1025    1026    1027    1028    1030    1031    1032    1033    1034#
                   1035    1036    1037    1038    1040    1041    1042    1043    1045    1046    1047    1048    1050
                   1051    1052    1053    1054#
$ZFIG = 000000      217#    758     759#    760#    761#    762#    763     773     774#    775#    776#    777#    778
                    789     790#    791#    792#    793#    794     804     805#    806#    807#    808#    809     821
                    822#    823#    824#    825#    826     837     838#    839#    840#    841#    842     852     853#
                    854#    855#    856#    857     868     869#    870#    871#    872#    873     1015    1016#   1017#
                   1018#   1019#   1020    1021#   1022#   1023#   1024#   1025    1026#   1027#   1028#   1029#   1030
                   1031#   1032#   1033#   1034#   1035    1036#   1037#   1038#   1039#   1040    1041#   1042#   1043#
                   1044#   1045    1046#   1047#   1048#   1049#   1050    1051#   1052#   1053#   1054#   1055
$ZX     = 000004   1059#   1065    1071    1077    1083    1085#   1102    1119    1136    1153
.       = 017306R   318#    319#    320#    321#    322#    323#    324#    325#    444     447     450     453     619#
                    624#    625#    626#    653#    654#    664#    665#    674#    675#    685#    686#    696#    697#
                    706#    707#    718#    719#    729#    732#    733#    758#    759#    760#    761#    763#    764#
                    773#    774#    775#    776#    779#    780#    789#    790#    791#    792#    794#    795#    804#
                    805#    806#    807#    811#    812#    821#    822#    823#    824#    827#    828#    837#    838#
                    839#    840#    842#    843#    852#    853#    854#    855#    858#    859#    868#    869#    870#
                    871#   1015#   1016#   1017#   1018#   1020#   1021#   1022#   1023#   1025#   1026#   1027#   1028#
                   1030#   1031#   1032#   1033#   1035#   1036#   1037#   1038#   1040#   1041#   1042#   1043#   1045#
                   1046#   1047#   1048#   1050#   1051#   1052#   1053#

. ABS.  000000     000
        017306     001
```



```
ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

XDVAB0,XDVAB0/SOL/CRF:SYM=DDXCOM,XDVAB0
RUN-TIME: 5 7 .8 SECONDS
RUN-TIME RATIO: 25/13=1.8
CORE USED:  8K  (15 PAGES)
```