```
.REM    !
```

IDENTIFICATION

PRODUCT CODE: AC-F887C-MC

PRODUCT NAME: CXDREC0 DR11M,DR11L MODULE

PRODUCT DATE: APRIL 1979

MAINTAINER: DEC/X11 SUPPORT GROUP

1.0 ABSTRACT

THIS MODULE IS DESIGNED TO EXERCISE THE DR11L (TWO WORD INPUT INTERFACE TO THE PDP-11 UNIBUS) AND/OR THE DR11M (TWO WORD OUTPUT INTERFACE TO THE PDP-11 UNIBUS). IT EXERCISES THE DR11L ALONE BY HAVING THE DR11L INTERRUPT THE CPU. IT EXERCISES THE DR11M ALONE BY HAVING THE DR11M INTERRUPT THE CPU AND BY WRITE/READING THE DR11M'S DBR. IF THE DR11L AND DR11M OPTIONS ARE CABLED TOGETHER FOR TEST, IT EXERCISES THEM BOTH BY HAVING THE DR11M INTERRUPT FOR DATA. THE DR11M TRANSFERS DATA TO THE DR11L AND THE DR11L INTERRUPTS THE CPU WITH THE DATA.

ALL DEVICE ADDRESSES AND VECTORS MUST BE CONSECUTIVE WITHIN THE DR11M OR DR11L GROUP.

DEFINITIONS

MODULE: A MODULE IS A SUB-PROGRAM DESIGNED TO BE CONFIGURED WITH THE DEC/X11 MONITOR IN ORDER TO TEST A PARTICULAR DEVICE.

WORD: A WORD IS A SUB-SECTION OF LOGIC OF EITHER THE DR11L OR DR11M. EACH WORD COULD BE CONSIDERED A SEPARATE DEVICE SINCE IT HAS ITS OWN CSR AND DBR.

2.0 REQUIREMENTS

HARDWARE:

1. DR11L OR DR11M- AN H8913 (MAINTENANCE LOOPBACK CONNECTOR) IS REQUIRED IN EACH WORD OF THE DEVICE TO BE TESTED.

AND/OR

2. DR11L AND DR11M-A BC08 CABLE (OR EQUIVALENT) IS REQUIRED FOR EACH WORD OF THE DEVICES TO BE TESTED.

STORAGE:: DRE REQUIRES:

1. DECIMAL WORDS: 932
2. OCTAL WORDS: 1644
3. OCTAL BYTES: 3511

3.0 PASS DEFINITION

ONE PASS OF THIS MODULE CONSISTS OF 65,536 INTERRUPTS FROM EACH WORD UNDER TEST.

4.0 EXECUTION TIME

ONE PASS RUNNING ALONE ON A PDP-11/05 TAKES APPROXIMATELY ONE MINUTE.

5.0 CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

```
DEVCNT:1  (NUMBER OF WORDS TO BE EXERCISED)
DEVADR:0  (ADDRESS OF THE 1ST CSR OF THE 1ST DR11L)
 ADDR2:1  (ADDRESS OF THE 1ST CSR OF THE 1ST DR11M)
   SR1:0  (HOW DEVICES ARE SETUP TO BE TESTED)
VECTOR:1  (VECTOR ADDR. OF 1ST WORD OF 1ST DR11L)
 VECT2:1  (VECTOR ADDR. OF 1ST WORD OF 1ST DR11M)
```

REQUIRED PARAMETERS:

TO RUN THIS MODULE, YOU MUST:

1. IF YOU ARE TESTING DR11L'S, YOU MUST SUPPLY THE MODULE WITH THE FIRST ADDRESS OF THE FIRST DR11L WORD TO BE TESTED. TO DO THIS AT CONFIGURATION TIME, YOU WOULD ENTER THAT ADDRESS IN "DEVADR". AT RUN TIME YOU WOULD MODIFY LOCATION "ADDR" (APC 6).

2. IF YOU ARE TESTING DR11L'S, YOU MUST ALSO SUPPLY THE VECTOR ADDRESS OF THE FIRST DR11L WORD. THIS MAY BE DONE AT CONFIGURATION TIME OR RUN TIME BY SUPPLYING THE VECTOR ADDRESS TO "VECTOR" (APC 10).

3. AT CONFIGURATION TIME YOU MUST SUPPLY "DEVCNT" WITH THE TOTAL NUMBER OF WORDS, DR11L OR DR11M, YOU WISH TO EXERCISE. THIS MAY BE DONE AT RUN TIME BY MODIFYING LOCATION "DVID1" (APC 14).

4. IF YOU ARE EXERCISING DR11M'S, AT RUN TIME YOU MUST MODIFY LOCATION "ADDR2" (APC 164) WITH THE FIRST ADDRESS OF THE FIRST DR11M TO BE TESTED.

5. IF YOU ARE EXERCISING DR11M'S, AT RUN TIME YOU MUST MODIFY LOCATION "VECT2" (APC 166) WITH THE VECTOR ADDR. OF THE FIRST DR11M WORD.

6. NOW THE LAST THING YOU MUST MODIFY AT RUN TIME, IS SR1 (APC 16). THE DEVICE REPRESENTATION BY SR1 IS SIMILAR TO DVID1. EACH BIT IN SR1 FROM BIT0 TO BIT15, REPRESENT THE FIRST DEVICE WORD THROUGH THE LAST DEVICE WORD. SR1 TELLS THE PROGRAM WHETHER THE DEVICES SET FOR TEST ARE CABLED TO EACH OTHER, OR HAVE H8913 IN THEM.

   EXAMPLE:

   1. 4 DR11L'S, THE FIRST HAS ADDRESS 176000, VECTOR 300; NO DR11M'S.

      DEVADR (ADDR) = 176000
      VECTOR = 300
      DEVCNT = 8 (2 WORDS PER DEVICE TIMES 4 DEVICES)
      DVID1 = 000377
      ADDR2 = 0
      VECT2 = 0
      SR1 = 0

      EACH DR11L WORD (J1, J2) WOULD HAVE AN H8913 INSTALLED. IF YOU HAD LESS THAN 8 H8913S, YOU WOULD INSTALL ALL THE H8913, AND SPECIFY WHICH WORDS HAD H8913 BY MODIFYING LOCATION "DVID1" TO INDICATE WHICH DR11L WORDS WERE TO BE TESTED.

   2. 4 DR11M'S, THE FIRST ADDRESS 177000, VECTOR 400; NO DR11L'S.

      VECTOR = 0
      DEVCNT = 8
      DVID1 = 000377
      ADDR2 = 177000
      VECT2 = 400
      SR1 = 0

   3. 4 DR11L'S, THE FIRST ADDRESS 176000, VECTOR 300
      4 DR11M'S, THE FIRST ADDRESS 177000, VECTOR 400
      6 BC08 CABLES, 4 H8913S.

      DEVADR (ADDR) = 176000
      VECTOR = 300
      DEVCNT = 8
      DVID1 = 000377
      ADDR2 = 177000
      VECT2 = 400
      SR1 = 000077

      SR1 BITS 00-05 TO INDICATE TO THE FIRST 6 WORDS OF THE DR11L'S AND DR11M'S ARE CABLED TO EACH OTHER FOR TEST. SR1 BITS 06 AND 07 = 0 TO INDICATE THAT THE LAST 2 WORDS (DR11L AND DR11M) HAVE H8913 INSTALLED

A) MAKE CERTAIN THAT EACH WORD (SELECTED BY DVID1 THAT HAS A CORRESPONDING BIT IN SR1 CLEARED) HAS AN H8913 INSTALLED IN IT.

OR

B) MAKE CERTAIN THAT EACH WORD (SELECTED BY DVID1 THAT HAS A CORRESPONDING BIT IN SR1 SET) HAS A CABLE BETWEEN THE DR11L INPUT WORD AND THE DR11M OUTPUT WORD.

7.0 MODULE OPERATION

A. ADDRESS EACH DR11L AND DR11M ADDRESS SELECTED FOR TEST. AT THIS TIME IF THE OPTIONS DON'T RETURN SLAVE-SYNC TO THE CPU, A "DEC/X11 SYS ERROR" WILL OCCUR.

B. THE VECTOR ADDRESS ENTERED BY THE OPERATOR FOR EACH DEVICE WILL BE CHECKED. IF THE VECTOR ADDRESS IS LESS THAN 100, AN ERROR MESSAGE WILL BE TYPED (SEE NON-STANDARD PRINTOUTS) AND THE WORD PAIR WILL BE DESELECTED FOR THE REMAINDER OF THE MODULE RUN. IF THE VECTOR ADDRESS ENTERED BY THE OPERATOR DOESN'T AGREE WITH THE VECTOR ADDRESS READ IN THE DEVICES' CSR, AN ERROR WILL BE TYPED.

C. IF THE INTERRUPT PRIORITY OF THE WORD IS 6 OR 7 A WARNING MESSAGE WILL BE TYPED (SEE 9 NON-STANDARD PRINTOUTS).

D. EACH WORD OF EACH DEVICE WILL BE TESTED TO SEE IF IT WILL INTERRUPT. IF A WORD FAILS TO INTERRUPT, AN ERROR MESSAGE WILL BE TYPED AND THAT WORD (PAIR) WILL BE DESELECTED FOR THE REMAINDER OF THE MODULE RUN.

E. (RESTRT) IF ALL WORD (PAIRS) HAVE BEEN DESELECTED FROM TEST DUE TO ERRORS, THE DEC/X11 MODULE WILL DROPPED.

F. POINT TO FIRST PATTERN.

G. POINT TO FIRST WORD PAIR.

H. IF NO DR11M OR DR11L TO BE TESTED, THEN STEP M. IF NO DR11M TO BE TESTED, THEN STEP N. IF SR1=0 THEN SET BIT 8 IN CSR TO CAUSE INTERRUPT, OTHERWISE READ DR11LS' DBR - THIS WILL CAUSE DR11M TO INTERRUPT WHEN ITS INTR. ENABLE IS SET.

I. SET DR11M'S VECTOR AND STATUS ADDRESSES.

J. SET DR11M'S INTR. ENABLE.

K. DEC/X11 EXIT. DR11M WILL INTR. TO L.

L. CLEAR DR11M'S INTR. ENABLE.

M. DEC/X11 PIRQ, WILL RETURN TO N.

N. LOAD CURRENT PATTERN INTO DR11M'S DBR, IF SR1=1, THIS WILL CAUSE DR11L TO INTERRUPT WHEN ITS INTR. ENABLE SET, OTHERWISE IF SR1=0 SET DR11L'S CSR BIT 8=10.

O. IF NOT TESTING A DR11L, GOTO STEP T.

P. SET DR11L'S INTR. ENABLE.

Q. DEC/X11 EXIT, DR11L WILL INTR. TO R.

R. CLEAR DR11L'S INTR. ENABLE.

S. DECX11 PIRQ. THE PROGRAM WILL RETURN TO T.

T. IF TESTED A WORD PAIR, READ DR11L'S DBR, OTHERWISE IF TESTING A LONE DP11M, READ DR11M'S DBR.

U. COMPARE CONTENTS TO THAT OF DATA SENT TO DR11M. IF NO ERROR STEP W.

V. DEC/X11 DATERR.

W. POINT TO NEXT WORD PAIR TO BE TESTED IF NO MORE WORD PAIRS THEN STEP X OTHERWISE STEP H.

X. UPDATE PATTERN POINTER IF NOT DONE ALL PATTERNS THEN GO TO STEP G.

Y. IF NOT DONE ENOUGH ITERATIONS GO TO STEP F, ELSE DEC/X11 END PAS.

8.0 OPERATION OPTIONS

DEVADR: THE ADDRESS OF THE CSR OF THE FIRST WORD OF THE DR11L UNDER TEST. IF RUNNING MULTIPLE DR11L'S, THEIR ADDRESSES MUST BE CONSECUTIVE. IF NO DR11L'S ARE TO BE TESTED, THIS LOCATION MUST BE ZEROED.
NOTE: SEE 5.0 CONFIGURATION REQUIREMENTS.

DEVAD2: THE ADDRESS OF THE CSR OF THE FIRST WORD OF THE FIRST DR11M UNDER TEST. IF RUNNING MULTIPLE DR11M'S THEIR ADDRESSES MUST BE CONSECUTIVE. IF NO DR11M'S ARE TO BE TESTED, THIS LOCATION MUST BE ZEROED.

NOTE: SEE SECTION 5.0 CONFIGURATION REQUIREMENTS.

DEVCNT: AT CONFIGURATION TIME, YOU SUPPLY THIS LOCATION WITH THE NUMBER OF WORDS (MAX=16, TWO WORDS PER DEVICE) THAT YOU WISH TO EXERCISE. THIS COUNT IN OCTAL REPRESENTS BOTH DR11L'S AND DR11M'S. IMPORTANT: IF YOU ARE EXERCISING BOTH DR11L'S AND DR11M'S, YOU MUST PREPARE AN EQUAL NUMBER FOR TEST. IF YOU CANNOT MEET THIS REQUIREMENT, YOU MAY CONFIGURE THIS MODULE FOR THE REMAINING UNEQUAL NUMBER. FOR EXAMPLE; IF YOU HAD 2 DR11L'S AND 3 DR11M'S, YOU WOULD CONFIGURE ONE MODULE TO EXERCISE 2 DR11M'S AND 2 DR11L'S, AND A SECOND MODULE TO EXERCISE ONE DR11M.

THE CONFIGURATOR WILL TAKE THE NUMBER YOU SUPPLIED AND FILL IN A LOCATION WITHIN THE MODULE CALLED "DVID1". EACH BIT OF THE WORD DVID1 (FROM BIT00 TO BIT15) REPRESENT A WORD TO BE TESTED (FROM WORD 1 OF 1ST DEVICE TO 1 OF 2 OF THE NTH (1 TO 8) DEVICE). TO DELETE ANY OR (BOTH DR11L AND DR11M) FROM TEST, ZERO THE CORRESPONDING BIT IN DVID1.

SR1: SR1 IS USED BY THE MODULE TO DETERMINE WHETHER YOU ARE TESTING THE DEVICES ALONE (USING AN H8913) OR WHETHER YOU HAVE THEM CABLED. EACH BIT OF SR1 (FROM BIT00 TO BIT15) REPRESENT A WORD TO BE TESTED (FROM WORD 1 OF FIRST DEVICE (PAIR) TO WORD 1 OR 2 OF THE NTH (1 TO 8) DEVICE (PAIR)).

IF AN SR1 BIT (BIT00 TO BIT15) IS ZERO, AND THE CORRESPONDING BIT IN DVID1 IS SET, THE PROGRAM WILL ASSUME YOU WISH TO EXERCISE THE CORRESPONDING WORD OF THE DR11L AND/OR DR11M WITH AN H8913 INSTALLED.

IF AN SR1 BIT (BIT00 TO BIT15) IS SET (=1) AND THE CORRESPONDING BIT IN DVID1 IS SET, THE PROGRAM WILL ASSUME YOU WISH TO EXERCISE THE CORRESPONDING WORDS OF THE DR11L AND DR11M TOGETHER, AS THEY ARE CABLED.

9.0 NON-STANDARD PRINTOUTS

A. MOST PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN THE DEC/X11 DOCUMENT.

B. IF A VECTOR ADDRESS IS LESS THAN 100, THE MODULE WILL NOT EXERCISE THE MODULE. IT WILL DROP THE OPTIONS WORD FROM TEST AND TYPE THIS MESSAGE:

"DR11(L OR M) ADDR: XXXXXX VECTOR ADDRESS LESS THAN 100 RESET VECTOR SWITCHES ON DR11 (L OR M)."

C. IF THE PRIORITY OF A DR11 (L OR M) WORD IS EITHER 6 OR 7 A WARNING MESSAGE WILL BE TYPED:

"DR11 (L OR M) PRIORITY OF 6 OR 7 MAY INTERFERE WITH ERROR FREE EXECUTION OF OTHER MODULES."

D. IF A WORD OR WORD PAIR (DR11L AND/OR DR11M) IS DROPPED FROM TEST:

"REPEATING DR11(L OR M) DROPPED FROM TEST".

E. IF ALL WORDS HAVE BEEN DROPPED FROM TEST, THE FOLLOWING MESSAGE WILL BE TYPED:

"DREA? NO MORE WORDS TO EXERCISE".

FOLLOW BY A DEC/X11 "END" (OR DROPPED) MESSAGE.

!

```
                                        IOMOD   <DREC >,1,1,4,4,,0,72
                                        MODULE  140000,DREC ,1,1,4,4,,0,72
                                        .TITLE  DREC DEC/X11 SYSTEM EXERCISER MODULE
                                ;       DDXCOM  VERSION  6       23-MAY-78
                                        .LIST   BIN
                                ;*********************************************************************
000000'                         BEGIN:
000000' 051104  041505      040 MODNAM: .ASCII  /DREC / ;MODULE NAME.
000005'     000                 XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 000001                  ADDR:   1+0             ;1ST DEVICE ADDR.
000010' 000001                  VECTOR: 1+0                     ;1ST DEVICE VECTOR.
000012'     200                 BR1:    .BYTE   PRTY4+0         ;1ST BR LEVEL.
000013'     200                 BR2:    .BYTE   PRTY4+0         ;2ND BR LEVEL.
000014' 000001                  DVID1:  +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' 000274'                 INIT:   START                   ;MODULE START ADDR.
000032' 000224'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                       ;PASS COUNTER.
000036' 000060                  ICONT:  0                       ;# OF ITERATIONS PER PASS=0
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' 002044'                 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' 000072                  IDNUM:  72                      ;MODULE IDENTIFICATION NUMBER=72
          000040                        .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
```



```
                                        .NLIST
                                        .WORD   0
                                        .LIST
                                        .ENDR
000224'                         MODSP:
                                ;*********************************************************************
```

```
                                        ;USER REQUIRED INFORMATION
000224' 000001          ADDR2:  1               ;USER ENTERED ADDRESS OF 1ST CSR OF FIRST WORD
                                                ;OF FIRST DR11M (IF ANY) ENTERED FOR TEST.
000226' 000001          VECT2:  1               ;USER ENTERED VECTOR ADDRESS OF FIRST WORD OF
                                                ;FIRST DR11M (IF ANY) ENTERED FOR TEST.


                                        ;MODULE REQUIRED REGISTERS - SET UP BY THIS MODULE.

000230' 000000          DRLADR: OPEN            ;ADDRESS OF CURRENT DR11L UNDER TEST
000232' 000000          DRLDBR: OPEN
000234' 000000          DRMADR: OPEN            ;ADDRESS OF CURRENT DR11M UNDER TEST
000236' 000000          DRMDBR: OPEN
000240' 000000          DVIDL:  OPEN            ;POINTS TO DR11LS TO BE EXERCISED
000242' 000000          DVIDM:  OPEN            ;POINTS TO DR11MS TO BE EXERCISED.
000244' 000000          SRL:    OPEN            ;POINTS TO DR11LS WITH H8913 INSTALLED.
000246' 000000          SRM:    OPEN            ;POINTS TO DR11MS WITH H8913 INSTALLED.
000250' 000000          SRB:    OPEN            ;POINTS TO DR11L AND DR11M CABLED PAIR.
000252' 000000          DRVCT:  OPEN            ;GETS LOADED FROM VECTOR OR VECT2 OF DEVICE UNDER TEST.
000254' 000000          VCTCAL: OPEN            ;SAME AS DRVCT ONLY CALCULATED FROM DEVICE CSR.
000256' 000000          PRIOR:  OPEN            ;PRIORITY OF DEVICE AS CALCULATED FROM DEVICE CSR.
000260' 000000          POINT:  OPEN            ;POINTER TO CURRENT DEVICE UNDER TEST.
000262' 000000          ITCNT:  OPEN            ;CONTAINS LOOP COUNT
000264' 140700          ITSET:  140700          ;CONTIANS NUMBER TO SET ITCNT TO
                                                ;IN ORDER TO GET 1 MIN. OF RUN TIME.
                                                ;RESTRT MAY VARY THIS NUMBER IF RUNNING
                                                ;MULTIPLE DEVICES IN ORDER TO SHORTEN RUN TIME.
000266' 000000          IFLAG:  OPEN            ;FLAG USED TO INDICATE IF AN
                                                ;INTERRUPTED OCCURED 0=NO;1=-YES.
000270' 000000          WARN:   OPEN            ;FLAG USED TO TYPE A WARNING MESSAGE IF = 0.
000272' 000000          PATPNT: OPEN            ;POINTS TO CURRENT PATTERN.


                                        ;
                                        ;THIS SECTION WILL TAKE ALL THE INFORMATION AS
                                        ;ENTERED BY THE USER AND MAKE CORRESPONDING
                                        ;SOFTWARE FLAGS.
                                        ;

000274' 016701  177514          START:  MOV     DVID1,R1        ;DEV COUNT TO R1
000300' 006201                  1$:     ASR     R1              ;SHIFT IN A BIT
000302' 103402                          BCS     2$              ;IF A BIT IN THIS POS=SHIFT
000304' 001413                          BEQ     3$              ;IF NO DEVS LEFT = BRANCH
000306' 000774                          BR      1$
000310' 062767  000002  177602  2$:     ADD     #2,INTR         ;2 MORE INTERRUPTS
000316' 062767  000002  177570          ADD     #2,WDTO         ;2 MORE WORDS TO MEM
000324' 062767  000002  177564          ADD     #2,WDFR         ;2 MORE WORDS FROM MEM
000332' 000762                          BR      1$              ;GO CHECK SOME MORE
000334' 005767  177446          3$:     TST     ADDR            ;ANY DR11L ADDR. ENTERED?
000340' 100405                          BMI     ST2             ;YES -WE CAN PROCEED.

000342' 005767  177656                  TST     ADDR2           ;NO L'S BUT IS ANY DR11M ADDR. ENTERED?
000346' 100402                          BMI     ST2             ;YES=PROCEED.

000350' 000167  001712                  JMP     DROP            ;NO=NO DR11L OR DR11M=LETS DROP THIS MODULE.

000354'                         ST2:
```



```
000354' 005767  177426                  TST     ADDR            ;ANY DR11L ADDRESS ENTERED?
000360' 100011                          BPL     1$              ;NO - SKIP NEXT CODE UNTIL 1$.

000362' 016767  177426  177654          MOV     DVID1,SRL       ;YES SET UP DR11L FLAGS.
000370' 046767  177422  177646          BIC     SP1,SRL         ;MAKE DR11L W/H8913 FLAG.
000376' 016767  177412  177634          MOV     DVID1,DVIDL     ;KEEP TRACK OF DR11LS SELECTED.

000404' 005767  177614          1$:     TST     ADDR2           ;ANY DR11M ADDRESSES ENTERED?
000410' 100011                          BPL     2$              ;NO - SKIP NEXT CODE UNTIL 2$.

000412' 016767  177376  177626          MOV     DVID1,SRM       ;YES SET UP DR11M FLAGS.
000420' 046767  177372  177620          BIC     SP1,SPM         ;MAKE DR11M W/8913 FLAG.
000426' 016767  177362  177606          MOV     DVID1,DVIDM     ;KEEP TRACK OF DR11MS SELECTED.

000434' 016767  177354  177606  2$:     MOV     DVID1,SRB       ;NOW MAKE A FLAG TO
000442' 046767  177576  177600          BIC     SRL,SRB         ;SHOW HOW MANY DR11LS ARE
000450' 046767  177572  177572          BIC     SPM,SRB         ;CONNECTED TO DR11MS


                                        ;
                                        ;THIS SECTION OF CODE WILL ADDRESS ALL DR11LS
                                        ;SELECTED FOR TEST, IF ANY DR11LS FAIL TO RETURN
                                        ;SLAVE-SYNC TO THE CPU, A "SYS ERR" WILL OCCUR
                                        ;
000456' 012767  000001  177574  CKADR:  MOV     #1,POINT        ;SET UP POINTER.
000464' 016767  177316  177536          MOV     ADDR,DRLADR     ;POINT TO FIRST DR11L ADDRESS.
000472' 036767  177562  177540  1$:     BIT     POINT,DVIDL     ;ANY DR11L SELECTED?
000500' 001402                          BEQ     2$              ;NO - GOTO 2$.

000502' 005777  177522                  TST     @DRLADR         ;YES - ADDRESS THE DR11L.

000506' 062767  000004  177514  2$:     ADD     #4,DRLADR       ;UPDATE TO LOOK AT NEXT DR11L ADDR.
000514' 006367  177540                  ASL     POINT           ;POINT TO NEXT DEVICE.
000520' 103364                          BCC     1$              ;LOOP IF NOT LOOKED AT ALL POSSIBLE DR11LS.

                                        ;
                                        ;THIS SECTION WILL ADDRESS ALL DR11MS SELECTED
                                        ;FOR TEST, IF ANY DR11MS FAIL TO RETURN
                                        ;SLAVE-SYNC TO THE CPU, A "SYS ERR" WILL OCCUR.
                                        ;

000522' 012767  000001  177530          MOV     #1,POINT        ;SET UP POINTER.
000530' 016767  177470  177476          MOV     ADDR2,DRMADR    ;POINT TO FIRST DR11M ADDR.

000536' 036767  177516  177476  3$:     BIT     POINT,DVIDM     ;ANY DR11M SELECTED?
000544' 001402                          BEQ     4$              ;NO - GOTO 4$.

000546' 005777  177462                  TST     @DRMADR         ;YES - ADDRESS THE DR11M.

000552' 062767  000004  177454  4$:     ADD     #4,DRMADR       ;UPDATE TO LOOK AT NEXT DR11M ADDR.
000560' 006367  177474                  ASL     POINT           ;POINT TO NEXT DEVICE.
000564' 103364                          BCC     3$              ;LOOP IF NOT LOOKED AT ALL POSSIBLE DR11MS.

                                        ;
                                        ;THIS SECTION OF CODE WILL COMPARE THE VECTOR ADDRESS
                                        ;ENTERED BY THE OPERATOR AGAINST THE VECTOR ADDRESS
                                        ;CALCULATED BY THE PROGRAM FOR THE DR11L BY READING
```

```
                                        ;ITS CSR
                                        ;
000566' 012767  000001  177464  CKV1A:  MOV     #1,POINT        ;SET UP POINTER.
000574' 016767  177206  177426          MOV     ADDR,DRLADR     ;SET UP FIRST DR11L ADDRESS.
000602' 016767  177202  177442          MOV     VECTOR,DRVCT    ;SET UP FIRST DR11L VECTOR ADDR, AS
                                        ;ENTERED BY USER.
000610' 005067  177454                  CLR     WARN            ;INDICATE THAT NO WARNING MESSAGE,
                                        ;HAS BEEN TYPED.
000614' 036767  177440  177416  1$:     BIT     POINT,DVIDL     ;ANY DR11L SELECTED?
000622' 001474                          BEQ     4$              ;NO GOTO 4$.
000624' 016767  177400  177246          MOV     DRLADR,CSRA     ;SET DR11L ADDR, FOR ERROR TYPEOUT (IF ANY).
000632' 026727  177414  000100          CMP     DRVCT,#100      ;IS VECTOR ADDR, > 100?
000640' 002020                          BGE     2$              ;IS YES GOTO 2$.
000642' 012767  000015  177236          MOV     #15,ERRTYP      ;WRONG VEC ADDR.
                                        ;***************************************************************
000650' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;;DR11L VECTOR LESS THAN 100
                                        ;***************************************************************

                                        ;NOTE IF ANY VECTOR WERE ALLOWED
                                        ;TO BE LESS THAN 100, IT COULD
                                        ;INTERFER WITH SOFTWARE TRAP CALLS, ETC.
000656' 005067  177356                  CLR     DVIDL           ;DROP ALL SOFTWARE
000662' 005067  177356                  CLR     SRL             ;FLAGS THAT POINT TO ANY DR11LS.
000666' 005067  177356                  CLR     SPB
000672' 104403  000000' 003154'         MSGN$,BEGIN,MS4P        ;ASCII MESSAGE CALL WITH COMMON HEADER
000700' 000456                          BR      5$              ;EXIT THIS TEST
000702' 016767  177322  177170  2$:     MOV     DRLADR,CSRA     ;SET DR11L ADDR, FOR TYPEOUT (IF ANY).
000710' 004767  002010                  JSR     PC,GETVP        ;GO GET VECTOR + PRIORITY FROM DR11L CSR
000714' 026727  177332  177332          CMP     DRVCT,VCTCAL    ;DOES THE USER ENTERED VECTOR ADDRESS AGREE
                                        ;WITH THE ADDRESS READ FROM THE CSR?
000722' 001420                          BEQ     3$              ;BR IF YES TO 3$.
                                        ;***************************************************************
000724' 104405  000000' 000000          HRDER$,BEGIN,NULL       ; ENTERED VEC ADDR NOT SAME AS ONE IN CSR,
                                        ;***************************************************************
000732' 046767  177322  177300          BIC     POINT,DVIDL     ;DROP ALL SOFTWARE
000740' 046767  177314  177276          BIC     POINT,SRL       ;FLAGS THAT POINT TO
000746' 046767  177306  177274          BIC     POINT,SPB       ;ANY DR11LS
000754' 104403  000000' 003154'         MSGN$,BEGIN,MS4P        ;ASCII MESSAGE CALL WITH COMMON HEADER
000762' 000414                          BR      4$              ;GOTO 4$
000764' 005767  177300          3$:     TST     WARN            ;HAS ANY WARNING MESSAGE BEEN TYPED OUT?
000770' 001011                          BNE     4$              ;YES - GOTO 4$
000772' 026727  177260  000005          CMP     PRIOR,#5        ;NO, IS THE PRIORITY OF THIS DEVICE > 5?
001000' 003405                          BLE     4$              ;NO GOTO 4$
                                                        ;YES,
001002' 104403  000000' 003144'         MSGN$,BEGIN,MS2P        ;ASCII MESSAGE CALL WITH COMMON HEADER
                                        ;TEXT: WARNING! DR11L/M PRIORITY IS
                                        ;GREATER THEN 5, THIS MAY INTERFER WITH
```



```
                                        ;ERROR EXECUTION OF OTHER DEC/X11 MODULES.
001010' 005267  177254                  INC     WARN            ;RECORD HAVING TYPED OUT THE WARNING MESSAGE.
001014' 062767  000004  177206  4$:     ADD     #4,DRLADR       ;UPDATE TO POINT TO NEXT DR11L ADDR.
001022' 062767  000004  177222          ADD     #4,DRVCT        ;LOOK AT NEXT VECTOR.
001030' 006367  177224                  ASL     POINT           ;POINT TO NEXT DR11L.
001034' 103267                          BCC     1$              ;IF WE HAVENT LOOK AT ALL, LOOP.
001036'                         5$:

                                        ;THIS SECTION OF CODE WILL COMPARE THE VECTOR ADDRESS
                                        ;ENTERED BY THE OPERATOR AGAINST THE VECTOR ADDRESS
                                        ;CALCULATED BY THE PROGRAM FOR THE DR11M BY READING ITS CSR
                                        ;
001036' 012767  000001  177214          MOV     #1,POINT        ;SET UP POINTER.
001044' 016767  177154  177162          MOV     ADDR2,DRMADR    ;SET UP FOR FIRST DR11M ADDRESS.
001052' 016767  177150  177172          MOV     VECT2,DRVCT     ;SET UP FIRST DR11M VECTOR ADDR, AS
                                        ;ENTERED BY USER.
001060' 036767  177174  177154  6$:     BIT     POINT,DVIDM     ;ANY DR11M SELECTED?
001066' 001477                          BEQ     8$              ;NO - GOTO 8$.
001070' 016767  177140  177002          MOV     DRMADR,CSRA     ;SET DR11M ADDR, FOR TYPEOUT (IF ANY).
001076' 026727  177150  000100          CMP     DRVCT,#100      ;IS VECTOR ADDR, > 100?
001104' 002020                          BGE     7$              ;IF YES GOTO 7$.
                                        ;NO.
001106' 012767  000015  176772          MOV     #15,ERRTYP              ;WRONG VEC ADDR
                                        ;***************************************************************
001114' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;;DR11M VECTOR LESS THAN 100
                                        ;***************************************************************
                                        ;NOTE: IF ANY VECTOR WERE ALLOWED TO BE LESS
                                        ;  THAN 100, IT COULD INTERFER WITH
                                        ;  SOFTWARE TRAP CALL, ETC.

001122' 005067  177114                  CLR     DVIDM           ;CLEAR ALL SOFTWARE FLAGS THAT
001126' 005067  177114                  CLR     SRM             ;POINT TO ANY DR11MS.
001132' 005067  177112                  CLR     SPB
001136' 104403  000000' 003154'         MSGN$,BEGIN,MS4P        ;ASCII MESSAGE CALL WITH COMMON HEADER
001144' 000461                          BR      IDRINT          ;GOTO IDRINT.

001146' 016767  177062  176724  7$:     MOV     DRMADR,CSRA     ;SET DR11M ADDR, FOR TYPEOUT (IF ANY).
001154' 004767  001544                  JSR     PC,GETVP        ;GO GET THE VECTOR + PRIORITY FROM DR11M CSR.

001160' 026767  177066  177066          CMP     DRVCT,VCTCAL    ;DOES THE USER ENTERED VECTOR ADDRESS AGREE
                                        ;WITH THE ADDRESS READ FROM THE CSR?
001166' 001423                          BEQ     7$              ;BR IF YES TO 7$.

001170' 012767  000015  176710          MOV     #15,ERRTYP              ;WRONG VEC ADDR.
                                        ;***************************************************************
001176' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;;USER ENTERED VECTOR ADDRESS NOT SAME AS ON READ IN CSR
                                        ;***************************************************************

001204' 046767  177050  177030          BIC     POINT,DVIDM     ;DROP ALL SOFTWARE
001212' 046767  177042  177026          BIC     POINT,SRM       ;FLAGS THAT POINT TO
```

```
001220' 046767  177034  177022          BIC     POINT,SRB       ;ANY DR11MS,
001226' 104403  000000' 003154'         MSGN$,BEGIN,MS4P        ;ASCII MESSAGE CALL WITH COMMON HEADER
001234' 000414                          BR      8$              ;GOTO 8$,

001236' 005767  177026          71$:    TST     WARN            ;HAS ANY WARNING MESSAGE BEEN TYPED?
001242' 001011                          BNE     8$              ;IF YES - GOTO 8$

001244' 026727  177006  000005          CMP     PRIOR,#5        ;NO, IS THE PRIORITY OF THIS DEVICE > 5?
001252' 003405                          BLE     8$              ;NO - GOTO 8$,
                                                                ;YES,

001254' 104403  000000' 003144'         MSGN$,BEGIN,MS2P        ;ASCII MESSAGE CALL WITH COMMON HEADER
                                                                ;TEXT: WARNING! DR11L/M PRIORITY IS GREATER
                                                                ;THEN 5, THIS MAY INTERFER WITH ERROR FREE
                                                                ;EXECUTION OF OTHER DECX11 MODULES

001262' 005267  177002                  INC     WARN

001266' 062767  000004  176740  8$:     ADD     #4,DRMADR       ;UPDATE TO POINT TO NEXT DR11M ADDR,
001274' 062767  000004  176750          ADD     #4,DRVCT        ;LOOK AT NEXT VECTOR
001302' 006367  176752                  ASL     POINT           ;POINT TO NEXT DR11M
001306' 103264                          BCC     6$              ;IF WE HAVEN'T LOOKED AT ALL, LOOP,

                                        ;
                                        ;THIS SECTION WILL DETERMINE IF ALL DR11LS SELECTED FOR
                                        ;TEST WILL INTERRUPT,

001310' 016767  176472  176712  IDPINT: MOV     ADDR,DRLADR     ;SET ADDR, OF 1ST DR11L (IF ANY),
001316' 016767  176702  176712          MOV     ADDR2,DRMDBR    ;SET ADDR OF 1ST DR11M (IF ANY),
001324' 062767  000002  176704          ADD     #2,DRMDBR       ;FIX ADDR, OF DR11M'S DBR,
001332' 012767  000001  176720          MOV     #1,POINT        ;POINT TO 1ST ONE,

001340' 036767  176714  176672  1$:     BIT     POINT,DVIDL     ;ANY DR11L AT THIS POINT?
001346' 001476                          BEQ     5$              ;IF NO - GOTO 5$,
001350' 016767  176654  176522          MOV     DRLADR,CSRA     ;RECORD ADDR, OF DR11L,
001356' 004767  001342                  JSR     PC,GETVP        ;GET VECTOR ADDR,
001362' 012777  001524' 176664          MOV     #4$,@VCTCAL     ;SET VECTOR ADDR, INTR TO "4$",
001370' 005067  176672                  CLR     IFLAG           ;CLEAR HAS INTERRUPTED FLAG,
001374' 036767  176660  176646          BIT     POINT,SRB       ;IS IT CABLED TO A DR11M?
001402' 001004                          BNE     2$              ;IF YES GOTO 2$,

001404' 052777  000400  176616          BIS     #BIT08,@DPLADR  ;NO - SET BIT08 IN CSR TO GENERATE INTR,
001412' 000402                          BR      3$              ;GOTO 3$,

001414' 005077  176616          2$:     CLR     @DRMDBR         ;LOAD DR11MS DBR WITH DATA - THIS,
                                                                ;SHOULD CAUSE DR11L TO INTR,

001420' 052777  000500  176602  3$:     BIS     #BIT06!BIT08,@DRLADR    ;SET INTR, ENABLE,

001426' 000240                          NOP                     ;GIVE TIME TO INTERRUPT (NO BREAK NEED, INTR, IMMED,),
001430' 005767  176632                  TST     IFLAG           ;HAS DEV, INTERRUPTED(IFLAG=1)?
001434' 001402                          BEQ     31$             ;NO THEN REPORT AN ERROR,
001436' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT,
                                                                ;CONTINUE AT THE POINT AFTER
                                                                ;THE INTR, SERVICE ROUTINE,
001442' 017767  176562  176432  31$:    MOV     @DRLADR,ACSR    ;SAVE CONTENTS OF CSR FOR TYPEOUT,
```



```

001450' 042777  000500  176552          BIC     #BIT06!BIT08,@DRLADR    ;CLEAR INTR ENABLE

001456' 012767  000023  176422          MOV     #23,ERRTYP              ;NO INTERRUPT
                                        ;*************************************************************
001464' 104405  000000' 000000          HDER$,BEGIN,NULL        ;;NO INTERRUPT FROM DR11L
                                        ;*************************************************************

001472' 046767  176562  176544          BIC     POINT,SRL       ;DROP THIS DEVICE FROM
001500' 046767  176554  176542          BIC     POINT,SRB       ;ANY DR11L PRESENT FLAGS (FROM CON, FLAG),
001506' 046767  176546  176524          BIC     POINT,DVIDL     ;(FROM SINGULAR FLAG)
001514' 104403  000000' 003154'         MSGN$,BEGIN,MS4P        ;ASCII MESSAGE CALL WITH COMMON HEADER
001522' 000410                          BR      5$              ;GOTO 5$,

001524' 042777  000500  176476  4$:     BIC     #BIT06!BIT08,@DRLADR    ;DEVICE INTR, TO HERE, CLEAR INTR, ENABLE
001532' 005267  176530                  INC     IFLAG           ;INDICATE DEV HAS INTR,(IFLAG=1),
                                        ;-------------------------------------------------------------
001536' 000004  000000' 001544'         PIRQ$,BEGIN,5$          ; QUEUE UP TO CONTINUE AT 5$ AND RTI
                                        ;-------------------------------------------------------------

001544' 062767  000004  176456  5$:     ADD     #4,DRLADR       ;POINT TO ADDRESS OF NEXT DR11L
001552' 062767  000004  176456          ADD     #4,DRMDBR       ;POINT TO NEXT ADDRESS OF DR11M (SHOULD THEY BE CABLED)
001560' 006367  176474                  ASL     POINT           ;SET POINTER TO NEXT
001564' 103265                          BCC     1$              ;IF NOT DONE ALL, LOOP,

                                        ;
                                        ;THIS SECTION OF CODE WILL DETERMINE IF ALL DR11MS SELECTED FOR
                                        ;TEST WILL INTERRUPT,
                                        ;

001566' 016767  176214  176436  6$:     MOV     ADDR,DRLDBR     ;SET UP BASE DR11L ADDR
001574' 016767  176424  176432          MOV     ADDR2,DRMADR    ;SET UP BASE DR11M ADDR,
001602' 062767  000002  176422          ADD     #2,DRLDBR       ;LOOK AT DR11L DBR,
001610' 012767  000001  176442          MOV     #1,POINT        ;SET TO POINT AT FIRST DR11M,

001616' 036767  176436  176416  7$:     BIT     POINT,DVIDM     ;ANY DR11M AT THIS POINT?
001624' 001476                          BEQ     11$             ;NO - GOTO 11$,

001626' 016767  176402  176244          MOV     DRMADR,CSRA     ;DR11M ADDR GOES IN CSRA FOR GETVP,
001634' 004767  001064                  JSR     PC,GETVP        ;GET DR11M VECTOR,
001640' 012777  002002' 176406          MOV     #10$,@VCTCAL    ;SET INTR, VECTOR TO 10$,
001646' 005067  176414                  CLR     IFLAG           ;CLEAR HAS INTERRUPTED FLAG,

001652' 036767  176402  176370          BIT     POINT,SRB       ;IS DR11L CABLED TO DR11M?
001660' 001004                          BNE     8$              ;YES GOTO 8$,

001662' 052777  000400  176344          BIS     #BIT08,@DRMADR  ;NO - SET BIT08 = CSR TO GENERATE INTR,
001670' 000402                          BR      9$              ;GOTO 9$,

001672' 005777  176334          8$:     TST     @DRLDBR         ;READ DR11L'S DBR, THIS WILL
                                                                ;CAUSE DR11M TO INTR,

001676' 052777  000100  176330  9$:     BIS     #BIT06,@DRMADR  ;SET INTERRUPT ENABLE,

001704' 000240                          NOP                     ;GIVE TIME FOR DEV, TO INTR,
001706' 005767  176354                  TST     IFLAG           ;HAS DEV INTERRUPTED(IFLAG=1)?
```

```
001712' 001402                  BEQ     91$             ;NO-THEN REPORT AN ERROR.
001714' 104400  000000'         EXIT$,BEGIN                     ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.
                                                        ;CONTINUE AT THE POINT AFTER
                                                        ;THE INTERRUPT SERVICE ROUTINE.

001720' 017767  176310  176154  91$:    MOV     @DRMADR,ACSR    ;SAVE CONTENTS OF CSR FOR TYPEOUT.
001726' 042777  000500  176300          BIC     #BIT06!BIT08,@DRMADR    ;HERE
001734' 012767  000023  176144          MOV     #23,ERRTYP              ;NO INTERRUPT OCCURRED
                                        ;*************************************************************
001742' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;;DR11M FAILED TO INTERRUPT
                                        ;*************************************************************

001750' 046767  176304  176264          BIC     POINT,DVIDM     ;DROP THIS DR11M FROM ALL SOFTWARES
001756' 046767  176276  176262          BIC     POINT,SRM       ;FLAGS.
001764' 046767  176270  176256          BIC     POINT,SRB
001772' 104403  000000' 003154'         MSGN$,BEGIN,MS4P        ;ASCII MESSAGE CALL WITH COMMON HEADER
002000' 000410                          BR      11$             ;GOTO 11$.

002002' 042777  000500  176224  10$:    BIC     #BIT06!BIT08,@DRMADR    ;DEVICE INTR, CLEAR INTR ENABLE.
002010' 005267  176252                  INC     IFLAG           ;INDICATE DR11M INTERRUPTED.
                                        ;-------------------------------------------------------------
002014' 000004  000000' 002022'         PIRQ$,BEGIN,11$         ; QUEUE UP TO CONTINUE AT 11$ AND RTI
                                        ;-------------------------------------------------------------

002022' 062767  000004  176204  11$:    ADD     #4,DRMADR       ;ADD 10 ADDRESS BASE TO
002030' 062767  000004  176174          ADD     #4,DRLDBR       ;LOOK AT NEXT ADDRESS RANGE.
002036' 006367  176216                  ASL     POINT           ;SET POINTER TO NEXT.
002042' 103265                          BCC     7$              ;IF NOT DONE ALL, LOOP.

                                        ;
                                        ;INTERRUPT + DATA CHECKING SECTION
                                        ;REENTRY POINT AFTER "END PASS".

002044' 016767  176214  176210  RESTRT: MOV     ITSET,ITCNT     ;PRESET ITERATION COUNT.

                                        ;
                                        ;THIS SECTION OF CODE WILL FIX ITCNT SO THAT IF RUNNING
                                        ;MULTIPLE DEVICES, IT SHOULD ONLY TAKE THIS MODULE 1 MIN. PER PASS.
                                        ;

002052' 005001                          CLR     R1              ;COUNT DEVICES IN R1.
002054' 012767  000001  176176          MOV     #1,POINT        ;FIX POINTER.
002062' 036767  176172  176154  1$:     BIT     POINT,SRL       ;ANY DR11L'S AT THIS POINT?
002070' 001014                          BNE     3$              ;YES THEN COUNT IT.
002072' 036767  176162  176146          BIT     POINT,  SRM     ;ANY DR11M'S AT THIS POINT?
002100' 001010                          BNE     3$              ;YES THEN COUNT IT.
002102' 036767  176152  176140          BIT     POINT,SRB       ;ANY PAIR AT THIS POINT?
002110' 001004                          BNE     3$              ;YES THEN COUNT IT

002112' 006367  176142          2$:     ASL     POINT           ;FIX POINT TO LOOK AT NEXT SET.
002116' 001361                          BNE     1$              ;IF NOT LOOKED AT ALL POINTS,LOOP,ELSE EXIT.
002120' 000402                          BR      4$              ;EXIT IF ALL POINTS TESTED.

002122' 005201                  3$:     INC     R1              ;COUNT THIS DEVICE
002124' 000772                          BR      2$
```



```
002126' 162701  000002          4$:     SUB     #2,R1           ;ITCNT ALREADY SET FOR FIRST 2 DEVICES.

002132' 162701  000002          5$:     SUB     #2,R1           ;ANY MORE DEVICES?
002136' 100404                          BMI     6$              ;NO THEN EXIT RIGHT COUNT IN ITCNT.
002140' 062767  006000  176114          ADD     #6000,ITCNT     ;YES UPDATE ITCNT FOR LESS ITERATIONS.
002146' 000771                          BR      5$              ;LOOK FOR MORE.

002150' 005767  175662          6$:     TST     ICONT           ;FIRST TIME THROUGH?
002154' 001003                          BNE     7$              ;NO BRANCH
002156' 016767  176100  175652          MOV     ITCNT,ICONT     ;YES - SET UP ICONT
002164' 012767  003042' 176100  7$:     MOV     #PATLST,PATPNT  ;SET UP PATTERN POINTER.

002172' 016767  175612  176030  LOOP:   MOV     ADDR,DRLADR     ;SET BASE ADDR. OF DR11L CSR.
002200' 016767  175602  176024          MOV     ADDR,DRLDBR     ;SET BASE ADDR OF DR11 DBR.
002206' 062767  000002  176016          ADD     #2,DRLDBR       ;EQUAL CSR+2.
002214' 016767  176004  176012          MOV     ADDR2,DRMADR    ;SET BASE ADDR. OF DR11M CSR.
002222' 016767  175776  176006          MOV     ADDR2,DRMDBR    ;SET BASE ADDR. OF DR11M DBR
002230' 062767  000002  176000          ADD     #2,DRMDBR       ;EQUAL CSR+2.
002236' 012767  000001  176014          MOV     #1,POINT        ;POINT TO FIRST DR11L AND/OR DR11M.

002241' 005767  175774                  TST     SRL             ;ANY DR11LS LEFT TO TEST?
002250' 001013                          BNE     MWORK           ;YES CONTINUE TESTING.
002252' 005767  175770                  TST     SRM             ;NO - BUT ARE THE ANY DR11MS LEFT?
002256' 001010                          BNE     MWORK           ;YES - GO WORK ON THEM.
                                                                ;NO,
002260' 005767  175764                  TST     SRB             ;ANY CABLED TOGETHER?
002264' 001005                          BNE     MWORK           ;YES-GO WORK ON THEM.
                                                                ;NO,

002266'                         DROP:
002266' 104403  000000' 003150'         MSGN$,BEGIN,MS3P        ;ASCII MESSAGE CALL WITH COMMON HEADER
                                                                ;TEXT: "DREA - NO MORE WORDS TO
                                                                ;EXERCISE".
002274' 104410  000000'                 END$,BEGIN              ;
                                                                ;DEC/X11 END CALL TO DROP THIS SOFTWARE
                                                                ;MODULE FROM CURRENT RUN.

002300' 036767  175754  175740  MWORK:  BIT     POINT,SRM       ;ANY SINGULAR DR11MS FOR TEST?
002306' 001404                          BEQ     1$              ;NO - GOTO 1$.

002310' 052777  000400  175716          BIS     #BIT08,@DRMADR  ;YES, SET BIT08 IN CSR TO GENERATE AN INTR.
002316' 000406                          BR      2$              ;GOTO 2$.

002320' 036767  175734  175722  1$:     BIT     POINT,SRB       ;ANY DR11L CABLE TO DR11M AT THIS POINT?
002326' 001430                          BEQ     LWORK           ;NO - GOTO "LWORK".

002330' 005777  175676                  TST     @DRLDBR         ;YES - READ DR11LS DBR THIS WILL CAUSE
                                                                ;DR11M TO INTR, WHEN ITS INTR, ENABLE IS SET
002334' 016767  175674  175536  2$:     MOV     DRMADR,CSRA     ;PUT CURRENT ADDR. IN CSRA FOR GETVP.

002342' 004767  000356                  JSR     PC,GETVP        ;GET DR11M'S VECTOR ADDR.

002346' 012777  002366' 175700          MOV     #MSERV,@VCTCAL  ;SET UP VECTOR ADDR, INTR TO MSERV.

002354' 052777  000100  175652          BIS     #BIT06,@DRMADR  ;SET INTERRUPT ENABLE.
002362' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.
```

```
                                                ;SHOULD INTERRUPT TO MSERV.
                                        ;DR11M INTERRUPTS TO HERE
002366' 042777  000500  175640  MSERV:  BIC     #BIT06!BIT00,@DRMADR ;CLEAR DEVICES INTR, ENABLE
                                        ;-------------------------------------------------------------
002374' 000004  000000' 002402'         PIRQ$,BEGIN,MDEL                ; QUEUE UP TO CONTINUE AT MDEL AND RTI
                                        ;-------------------------------------------------------------
002402' 017777  175664  175626  MDEL:   MOV     @PATPNT,@DRMDBR ;LOAD PATTERN INTO DBR OF DR11M.
002410' 036767  175644  175626  LWORK:  BIT     POINT,SRL       ;SINGULAR DR11L FOR TEST (W/H8913)?
002416' 001004                          BNE     1$              ;YES = GOTO 1$.
002420' 036767  175634  175622          BIT     POINT,SRB       ;ANY DR11L/M PAIR AT THIS POINT?
002426' 001501                          BEQ     LOOPE           ;NO = GOTO LOOP END.
002430' 016767  175574  175442  1$:     MOV     DRLADR,CSRA     ;PUT DR11L ADDR IN CSRA FOR GETVP.
002436' 004767  000262                  JSR     PC,GETVP        ;GO GET DR11L'S VECTOR ADDRESS.
002442' 012777  002462' 175604  2$:     MOV     #LSERV,@VCTCAL  ;SET UP DR11L VECTOR TO INTR. TO LSERV.
002450' 052777  000500  175552          BIS     #BIT00!BIT06,@DRLADR ;SET INTERRUPT ENABLE (06) AND BIT00.
                                                ;00 NO EFFECT IF CABLED PAIR; HOWEVER WILL
                                                ;CAUSE DR11L W/H8913 TO INTR.
002456' 104400  000000'                 FXIT$,BEGIN             ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.
                                        ;DR11L INTRS. TO HERE
002462' 042777  000500  175540  LSERV:  BIC     #BIT06!BIT00,@DRLADR ;CLEAR INTR, ENABLE.
                                        ;-------------------------------------------------------------
002470' 000004  000000' 002476'         PIRQ$,BEGIN,LDEL                ; QUEUE UP TO CONTINUE AT LDEL AND RTI
                                        ;-------------------------------------------------------------
002476' 036767  175556  175544  LDEL:   BIT     POINT,SRB       ;SERVICING DR11L/M PAIR?
002504' 001434                          BEQ     2$              ;NO = GOTO 2$.
                                                                ;YES.
002506' 017767  175520  175374          MOV     @DRLDBR,AWAS    ;READ DATA IN DR11L'S DBR.
002514' 016767  175512  175362          MOV     DPLDBR,WASADR   ;SET FOR ERROR TYPEOUT IF ANY.
002522' 016767  175544  175352  1$:     MOV     PATPNT,SBADR    ;FOR ERROR TYPEOUT = ADDR OF TEST DATA.
002530' 017767  175536  175350          MOV     @PATPNT,ASR     ;FOR ERROR TEST DATA.
002536' 026767  175344  175344          CMP     ASR,AWAS        ;DATA SENT = DATA RECEIVED?
002544' 001432                          BEQ     LOOPE           ;YES = THEN LOOP END.
                                                                ;NO.
                                        ;*************************************************************
002546' 104404  000000'                 DATER$,BEGIN            ;DATA ERROR!!!
                                        ;*************************************************************
                                                ;NOTE: IF TESTING SINGULAR DR11M W/H8913,
                                                ;DATA WAS WRITE/READ FROM DR11M'S DBR.
                                                ;IF DR11L/M PAIR, DATA WAS WRITTEN
                                                ;INTO DR11M'S DBR AND READ FROM DR11L'S DBR
002552' 046767  175502  175470          BIC     POINT,SRB       ;CLEAR ANY FLAGS ASSOC. WITH DR11L
002560' 046767  175474  175460          BIC     POINT,SRM       ;OR DR11L/M PAIR
002566' 104403  000000' 003154'         MSGN$,BEGIN,MS4P        ;ASCII MESSAGE CALL WITH COMMON HEADER
002574' 000416                          BR      LOOPE           ;GOTO LOOP END.
```



```
002576' 036767  175456  175442  2$:     BIT     POINT,SRM       ;SINGLUAR DR11M W/H8913?
002604' 001412                          BEQ     LOOPE           ;NO = JUST SINGLUAR DR11L - GOTO LOOP END.
002606' 017767  175424  175274          MOV     @DRMDBR,AWAS    ;YES READ DATA FROM DR11M'S DBR.
002614' 016767  175416  175262          MOV     DPMDBR,WASADR   ;SET FOR ERROR TYPEOUT (IF ANY).
002622' 016767  175406  175250          MOV     DRMADR,CSRA     ;RECORD ADDR. OF DR11M.
002630' 000734                          BR      1$              ;GO DO DATA CHECK.
                                        ;
                                        ;LOGIC LOOP END FOR SINGULAR OR PAIRED DEVICES.
                                        ;
002632' 062767  000004  175370  LOOPE:  ADD     #4,DRLADR       ;UPDATE ALL ADDRESS BASES
002640' 062767  000004  175364          ADD     #4,DRLDBR       ;TO LOOK AT NEXT
002646' 062767  000004  175360          ADD     #4,DRMADR       ;SINGULAR DR11L OR DR11M
002654' 062767  000004  175354          ADD     #4,DRMDBR       ;OR DR11L/M PAIR.
002662' 006367  175372                  ASL     POINT           ;POINT TO NEXT GROUP.
002666' 103204                          BCC     MWORK           ;IF NOT DONE ALL=LOOP.
002670' 062767  000002  175374  1$:     ADD     #2,PATPNT       ;UPDATE TO LOOK AT NEXT PATTERN.
002676' 026727  175370  003142'         CMP     PATPNT,#PATEND  ;DONE ALL PATTERNS?
002704' 001003                          BNE     2$              ;NO = GOTO 2$.
002706' 012767  003042' 175356          MOV     #PATLST,PATPNT  ;YES = RESET PATTERN POINTER.
002714'                         2$:
002714' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
002720' 000167  177246                  JMP     LOOP            ;NO=LOOP WHOLE THING.
                                        ;
                                        ;"GETVP"        SUBROUTINE TO CALCULATE A VECTOR ADDRESS AND
                                        ;               PRIORITY OF A DR11L OR DR11M BASED ON INFORMATION
                                        ;               IN THE DEVICE'S CSR.
                                        ;               PRIORITY IS CALCULATED FROM CSR BITS 4 + 5
                                        ;                       STATE OF        BIT5    BIT4    PRIORITY
                                        ;                                         0       0       4
                                        ;                                         0       1       5
                                        ;                                         1       0       6
                                        ;                                         1       1       7
                                        ;               VECTOR ADDRESS IS FOUND IN CSR BITS 09-14, THEY
                                        ;               REPRESENT REAL ADDR, BITS 03-08
                                        ;
                                        ;       TO CALL THIS ROUTINE:
                                        ;               MOV     "ADDRESS", CSRA
                                        ;               JSR     PC,GETVP
                                        ;       RETURNS WITH:
                                        ;               VECTOR ADDRESS IN VCTCAL
                                        ;               PRIORITY IN PRIOR
                                        ;
002724' 017767  175150  175324  GETVP:  MOV     @CSRA,PRIOR     ;GET CSR INFORMATION
002732' 042767  177717  175316          BIC     #177717,PRIOR   ;PRIORITY IN CSR BITS 4 AND 5
002740' 006267  175312                  ASR     PRIOR           ;RIGHT JUSTIFY 4+5 INTO
```

```
  997  002744' 006267  175306          ASR     PRIOR           ;BIT POSITIONS 1 AND 0.
  998  002750' 006267  175302          ASR     PRIOR
  999  002754' 006267  175276          ASR     PRIOR           ;DONE WITH PRIORITY.
 1000  002760' 052767  000004  175270  BIS     #4,PRIOR
 1001
 1002  002766' 017767  175106  175260  MOV     @CSRA,VCTCAL    ;GET CSR INFORMATION AGAIN.
 1003  002774' 042767  100777  175252  BIC     #100777,VCTCAL  ;STRIPT FOR VECTOR INFORMATION IN BITS 9-14.
 1004  003002' 000367  175246          SWAB    VCTCAL          ;CSR BITS 9 TO 14 REPRESENT REAL
 1005  003006' 006367  175242          ASL     VCTCAL          ;VECTOR ADDRESS BITS 03 TO 08.
 1006  003012' 006367  175236          ASL     VCTCAL          ;SO WE HAD TO PUT THEM IN THAT POSITION.
 1007
 1008  003016' 017746  175056          MOV     @CSRA,-(SP)     ;OK, NOW WE NEW BIT02 OF THE VECTOR ADDR.
 1009  003022' 042716  177773          BIC     #177773,(SP)    ;THIS IS REPRESENTED BY CSR BIT02 SO WE
 1010  003026' 052667  175222          BIS     (SP)+,VCTCAL    ;MUST ADD THAT TO THE VECTOR ADDR.
 1011  003032' 017767  175042  175042  MOV     @CSRA,ACSP      ;RECORD STATUS OF CSR.
 1012  003040' 000207                  PTS     PC              ;EXIT, WORK ALL DONE
 1013
 1014                                  ;
 1015                                  ;THIS IS A LIST OF ALL PATTERNS USED BY THIS MODULE.
 1016                                  ;IT USES A "FLOAT ZERO-COMPLEMENT" PATTERN.
 1017                                  ;
 1018  003042' 177777  000000  PATLST: .WORD   -1,0
 1019  003046' 177776  000002          .WORD   -2,2
 1020  003052' 177774  000004          .WORD   -4,4
 1021  003056' 177770  000010          .WORD   -10,10
 1022  003062' 177760  000020          .WORD   -20,20
 1023  003066' 177740  000040          .WORD   -40,40
 1024  003072' 177700  000100          .WORD   -100,100
 1025  003076' 177600  000200          .WORD   -200,200
 1026  003102' 177400  000400          .WORD   -400,400
 1027  003106' 177000  001000          .WORD   -1000,1000
 1028  003112' 176000  002000          .WORD   -2000,2000
 1029  003116' 174000  004000          .WORD   -4000,4000
 1030  003122' 170000  010000          .WORD   -10000,10000
 1031  003126' 160000  020000          .WORD   -20000,20000
 1032  003132' 140000  040000          .WORD   -40000,40000
 1033  003136' 100000  100000          .WORD   -100000,100000
 1034  003142' 000000          PATEND: .WORD   0
```



```
 1035
 1036                                  ;
 1037                                  ;ASCIZ MESSAGE POINTERS + MESSAGES
 1038                                  ;
 1039
 1040  003144' 003160' 177777          MS2P:   .WORD   MES2,-1
 1041  003150' 003345' 177777          MS3P:   .WORD   MES3,-1
 1042  003154' 003424' 177777          MS4P:   .WORD   MES4,-1
 1043
 1044  003160' 053445  051101  044516  MES2:   .ASCII  '%WARNING! DR11L/M PRIORITY IS GREATER THAN 5.'
 1045  003166' 043516  020041  051104
 1046  003174' 030461  027514  020115
 1047  003202' 051120  047511  044522
 1048  003210' 054524  044440  020123
 1049  003216' 051107  040505  042524
 1050  003224' 020122  044124  047101
 1051  003232' 032440      056
 1052  003235'     045  044124  051511          .ASCIZ  '%THIS MAY INTERFERE WITH ERROR FREE EXECUTION OF OTHER DEC/X11 MODULES%
 1053  003242' 046440  054501  044440
 1054  003250' 052116  051105  042506
 1055  003256' 042522  053440  052111
 1056  003264' 020110  051105  047522
 1057  003272' 020122  051106  042505
 1058  003300' 042440  042530  052503
 1059  003306' 044524  047117  047440
 1060  003314' 020106  052117  042510
 1061  003322' 020122  042504  027503
 1062  003330' 030530  020061  047515
 1063  003336' 052504  042514  022523
 1064  003344'     000
 1065
 1066  003345'     045  051104  040505  MES3:   .ASCIZ  '%DREA  -NO MORE WORDS (DR11L OR M) TO EXERCISE%'
 1067  003352' 026411  047516  046440
 1068  003360' 051117  020105  047527
 1069  003366' 042122  020123  042050
 1070  003374' 030522  046061  047440
 1071  003402' 020122  024515  052040
 1072  003410' 020117  054105  051105
 1073  003416' 044503  042523  000045
 1074
 1075  003424' 042045  042522  004501  MES4:   .ASCIZ          '%DREA  -ERRORING DR11(L OR M) WORD DROPPED FROM TEST%'
 1076  003432' 042455  051122  051117
 1077  003440' 047111  020107  051104
 1078  003446' 030461  046050  047440
 1079  003454' 020122  024515  053440
 1080  003462' 051117  020104  051104
 1081  003470' 050117  042520  020104
 1082  003476' 051106  046517  052040
 1083  003504' 051505  022524     000
 1084          000001                  .END
```

```
AC8R    000102R  419*  716*  778*  1011*
ADDR    000006R  385*  484   493   517   553   688   747   841   842
ADDR2   000224R  438*  487   500   534   621   689   748   844   845
ADDR22= 001000   437#
ASB     000106R  423*  925*  926
ASTAT   000104P  421#
AWAS    000110R  424*  922*  926   945*
BEGIN   000000R  382*  568   577   587   593   602   635   644   656   662   672   713   722
                 728   734   774   782   788   794   859   862   884   891   910   916   931
                 939   969
BIT0  = 000001   437#
BIT06 = 000100   468*  708   718   731   769   779   791   883   889   907   914
BIT08 = 000400   468*  702   708   718   731   763   779   791   869   889   907   914
BIT1  = 000002   437#
BIT10 = 002000   437#
BIT11 = 004000   437#
BIT12 = 010000   437#
BIT13 = 020000   437#
BIT14 = 040000   437#
BIT15 = 100000   437#
BIT2  = 000004   437#
BIT3  = 000010   437#
BIT4  = 000020   437#
BIT5  = 000040   437#
BIT6  = 000100   437#
BIT7  = 000200   437#
BIT8  = 000400   437#
BIT9  = 001000   437#
BREAK$= 104407   437#
BR1     000012R  387#
BR2     000013R  388#
BTOD$ = 104421   437#
CDATA$= 104412   437#
CKADR   000456R  516#
CKVTA   000566R  552#
CONFIG  000056R  407#
CSRA    000100R  417*  561*  579*  628*  647*  695*  755*  877*  902*  947*  994   1002  1008
                 1011
DATCK$= 104411   437#
DATER$= 104404   437#  931
DRLADR  000230R  445*  517*  521   523*  553*  561   579   608*  688*  695   702*  708*  716
                 718*  731*  737*  841*  902   907*  914*  955*
DRLDBR  000232R  446*  747*  749*  766   798*  842*  843*  875   922   923   956*
DRMADR  000234R  447*  534*  539   541*  621*  628   647   679*  748*  755   763*  769*  778
                 779*  791*  797*  844*  869*  877   883*  889*  947   957*
DRMDBR  000236R  448*  689*  690*  705*  738*  845*  846*  894*  945   946   958*
DROP    002266R  490   858#
DRVCT   000252R  454*  554*  563   582   609*  622*  629   650   680*
DVIDL   000240R  449*  498*  518   559   574*  590*  693   727*
DVIDM   000242R  450*  505*  536   625   641*  659*  752   785*
DVID1   000014R  389*  475   496   498   503   505   507
ENDIT$= 104413   437#  969
END$  = 104410   437#  862
ERRTYP  000106R  422#  566*  633*  654*  720*  780*
EXIT$ = 104400   437#  713   774   884   910
GETPA$= 104415   437#
```



```
GETVP   002724R  580   648   696   756   879   904   994*
GWBUF$= 104414   437#
HRDCNT  000044R  402#
HRDER$= 104405   437#  568   587   635   656   722   782
HRDPAS  000050R  404#
ICONT   000036R  399*  836   838*
ICOUNT  000040R  400#
IDNUM   000122R  429#
IDRINT  001310R  645   688#
IFLAG   000266R  463*  698*  711   732*  758*  772   792*
INIT    000030R  396#
INTR    000128R  428#  480#
ITCNT   000262R  458#  806*  833*  838
ITSET   000264R  459#  806
LDEL    002476R  916   919#
LOOP    002172R  841#  971
LOOPE   002632R  900   927   940   943   955*
LSERV   002462R  906   914#
LWORK   002410R  873   896#
MAP22$= 104416   437#
MDEL    002402P  891   894#
MES2    003160R  1040  1044#
MES3    003345P  1041  1066#
MES4    003424R  1042  1075#
MODNAM  000000R  383*
MODSP   000224R  397   435#
MSERV   002366R  881   889#
MSG$  = 104403   437#  577   593   602   644   662   672   728   788   859   939
MSGS$ = 104402   437#
MSG$  = 104401   437#
MS2P    003144R  602   672   1040*
MS3P    003150P  859   1041#
MS4P    003154R  577   593   644   662   728   788   939   1042*
MWORK   002300R  850   852   855   866*  960
NULL  = 000000   437#  568   587   635   656   722   782
OPEN  = 000000   384   390   391   392   393   410   411   412   413   414   415   416   417
                 419   421   423   424   426   427   428   437*  445   446   447   448   449
                 450   451   452   453   454   455   456   457   458   463   465   466
OTOA$ = 104420   437#
PASCNT  000034R  398#
PATEND  003142R  963   1034#
PATLST  003042R  839   966   1018#
PATPNT  000272R  466*  839*  894   924   925   962*  963   966*
PIRQ$ = 000004   437#  734   794   891   916
POINT   000260R  457#  516*  518   524*  533*  536   542*  552*  559   590   591   592   610*
                 620*  625   659   660   661   681*  691*  693   699   725   726   727   739*
                 750*  752   760   785   786   787   799*  814*  815   817   819   822*  847*
                 866   872   896   899   919   937   938   942   959*
POPSP = 005726   437#
POPSP2= 022626   437#
PRIOR   000256R  456*  599   668   994*  995*  996*  997*  998*  999*  1000*
PRTY  = 000000   437#
PRTY0 = 000000   437#
PRTY1 = 000040   437#
PRTY2 = 000100   437#
PRTY3 = 000140   437#
```

```
PRTY4 = 000200            387     388     437#
PRTY5 = 000240            437#
PRTY6 = 000300            437#
PRTY7 = 000340            437#
PS    = 177776            437#
PSW   = 177776            437#
PUSH  = 005746            437#
PUSH2 = 024646            437#
RAND$ = 104417            437#
RANNUM  000054R           406#
RESTPT  002044R           425     806#
RES1    000056R           408#
RES2    000060R           409#
RSTRT   000112R           425#
SBADR   000102P           418#    924#
SOFCNT  000042R           401#
SOFER$= 104406            437#
SOFPAS  000046R           403#
SPOINT  000032P           397#
SPSIZ = 000040              1#    430
SRB     000250R           453#    507#    508#    509#    576#    592#    643#    661#    699     726#    760     787#    819
                          854     872     899     919     937#
SRL     000244R           451#    496#    497#    508     575#    591#    725#    815     849     896
SRM     000246R           452#    503#    504#    509     642#    660#    786#    817     851     866     938#    942
SR1     000016R           390#    497     504
SR2     000020R           391#
SR3     000022R           392#
SR4     000024R           393#
START   000274R           396     475#
STAT    000026R           395#
ST2     000354R           485     488     492#
SVR0    000062R           410#
SVR1    000064R           411#
SVR2    000066R           412#
SVR3    000070R           413#
SVR4    000072R           414#
SVR5    000074R           415#
SVR6    000076R           416#
SYSCNT  000052R           405#
TRPDFD= 000022            437#
VCTCAL  000254R           455#    582     650     697#    757#    881#    906#    1002#   1003#   1004#   1005#   1006#   1010#
VECTOR  000010R           386#    554
VECT2   000226R           440#    622
WARN    000270R           465#    556#    596     606#    665     677#
WASADR  000104R           420#    923#    946#
WDFR    000116R           427#    482#
WDTO    000114R           426#    481#
XFLAG   000005R           384#

. ABS.  000000      000
        003511      001

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



```
 XDREC0,XDREC0/SOL/CRF:SYM=DDXCOM,XDREC0
 RUN-TIME:  5  9 .9 SECONDS
 RUN-TIME RATIO: 92/16=5.5
 CORE USED:  7K  (13 PAGES)
```