.REM_

IDENTIFICATION

PRODUCT CODE: AC-F808B-MC

PRODUCT NAME: CXDMEB0 DMP/DMV11 DECX SLV MOD

PRODUCT DATE: AUGUST 1981

MAINTAINER: DIAGNOSTIC ENGINEERING CC:38P

AUTHORS: DAVID HOFFMAN
CHRIS BRIENEN

1. ABSTRACT

THERE ARE 2 DEC/X11 IOMODX MODULES WRITTEN FOR THE DMP/DMV11.
THESE ARE DMD AND DME. TOGETHER, THESE 2 MODULES CAN OPERATE UP TO
16 (DEC.) DMP11 DEVICES IN POINT-TO-POINT LINKS, OR A SINGLE
DEVICE CONFIGURED AS A MULTIPOINT CONTROL STATION COMMUNICATING WITH
UP TO 32 (DEC.) TRIBUTARIES, OR UP TO 16 (DEC.) DEVICES CONFIGURED AS
MULTIPOINT TRIBUTARIES ON THE SAME PDP-11 UNIBUS (16/16 W/DMV11 QBUS).
THE BASIC OPERATION IS TO TRANSMIT, RECEIVE, AND CHECK
32(DEC.) DATA MESSAGES OF 1024 (DEC.) BYTES EACH, ON A GIVEN
PHYSICAL LINK. BY DEFAULT, THIS WOULD INVOLVE A SINGLE PDP-11 SYSTEM
WITH 1 OR MORE DEVICES OPERATED IN INTERNAL OR EXTERNAL LOOPBACK MODE.
HOWEVER, BY OPERATOR SELECTION OF NON-DEFAULT MODES, ACTUAL POINT-TO-
POINT OR MULTIPOINT OPERATION IS POSSIBLE. IN ONE SUCH MODE, THE MODULE
CAN DRIVE THE OPERATION OF A CONTROL STATION WHICH SENDS AND RECEIVES
BACK MESSAGES FROM A SPECIFIED LIST OF TRIBUTARIES, ON THE SAME
OR DIFFERENT PDP-11 SYSTEMS. IN ANOTHER SUCH MODE, THE MODULE CAN DRIVE
THE OPERATION OF A GROUP OF TRIBUTARIES WHICH RESIDE ON A SINGLE SYSTEM
AND REFLECT BACK MESSAGES SENT TO THEM BY A CONTROL STATION , ON THE
SAME OR DIFFERENT SYSTEM.

DMD IS THE MASTER MODULE WHICH CAN OPERATE UP TO 16 (DEC.) DEVICES IN
LOOPED-BACK OR POINT-TO-POINT MASTER MODES, OR A SINGLE DEVICE IN
MULTIPOINT CONTROL MODE.
DME IS THE SLAVE MODULE WHICH WHICH CAN OPERATE UP TO 16 (DEC.) DEVICES
IN POINT-TO-POINT SLAVE OR MULTIPOINT TRIBUTARY MODES.
A MASTER MODULE CAN BE SELF-SUFFICIENT (IF LOOPBACK IS USED) OR IT
CAN COMMUNICATE WITH A SLAVE MODULE(S) ON THE SAME OR ANOTHER PROCESSOR.
THE REST OF THIS DOCUMENT WILL DESCRIBE THE USE AND OPERATION OF THE
SLAVE MODULE, DME.

IT IS REQUIRED THAT THE OPERATOR CONFIGURE THE DEC/X11 EXERCISER WITH
A SEPARATE COPY OF MODULE DME FOR EACH GROUP OF POINT-TO-POINT SLAVES
OR MULTIPOINT TRIBUTARY DEVICES ON THIS PDP-11 SYSTEM.
THE ACTUAL OPERATING MODE OF EACH COPY IS SELECTED BY THE
SETTINGS OF SW1-SW4 (SOFTWARE SWITCHES) FOR THAT COPY. FOR
MULTI-PROCESSOR CONFIGURATIONS, A DEC/X11 EXERCISER MUST BE CONFIGURED
AND RUN ON EACH PDP-11 PROCESSOR, SIMULTANEOUSLY. NOTE THAT THE DME
MODULE IS NEVER ALLOWED TO RELOCATE; USE A MODULE THAT DOES NOT ALLOW
RELOCATION OR USE THE RUN LOCK COMMAND (RUNL).

AFTER THE MASTER MODULE (DMD) HAS BEEN STARTED ON ONE PROCESSOR, THE
OPERATOR IS ALLOWED AT LEAST 5 MINUTES TO START THE SLAVE MODULE(S)
(DME) ON THE OTHER PROCESSOR(S), BEFORE GETTING A PROTOCOL TRANSMIT
THRESHOLD ERROR ON THE MASTER MODULE PROCESSOR.

THE M8203 LINE UNIT CONTAINS HARDWARE SWITCHES WHICH CAN DEFINE THE
PHYSICAL LINK AND DMP11 OPERATING MODE, AND A TRIBUTARY ADDRESS
(IF MULTIPOINT). IF THE MODE SWITCHES ARE ENABLED, THE MODE DEFINED
IN THE SWITCHES WILL BE USED, IF POSSIBLE. IF THE MODE SWITCHES ARE
NOT ENABLED, THE PROGRAM WILL DEFINE THE MODE AUTOMATICALLY.
IN ANY CASE, THE PROGRAM DOES NOT REQUIRE OPERATOR INTERVENTION TO
CHANGE HARDWARE SWITCH SETTINGS OR CONFIGURE CABLES, ETC.

2. REQUIREMENTS

HARDWARE: 1 TO 16 (DEC.) DMP/DMV11'S AND ASSOCIATED CABLES
AND CONNECTORS.
STORAGE: DME REQUIRES ABOUT 2K WORDS OF STORAGE

3. PASS DEFINITION

ONE PASS OF THE DME MODULE CONSISTS OF TRANSMITTING AND RECEIVING
A TOTAL OF 32,768 (DEC.) 8-BIT CHARACTERS, PER DMP/DMV11.

4. EXECUTION TIME

DME RUNNING ALONE ON A PDP11/40 PROCESSOR TAKES APPROXIMATELY
1 MINUTE PER SLAVE DEVICE TO COMPLETE ONE PASS AT 9600 BAUD. THE PASS
TIME IS ALSO INVERSELY PROPORTIONAL TO THE BAUD RATE.

5. CONFIGURATION PARAMETERS

DEFAULT PARAMETERS :

DEVADR: 160170, VECTOR:300, BR1:5, BR2:UNUSED, DEVCNT:1

SOFTWARE SWITCH REGISTER OPTIONS :

THE ALLOWABLE OCTAL VALUES OF SR4 ARE:
SR4=000000 : IF TESTING DMP11
SR4=000001 : IF TESTING DMV11
SR4=000002 : IF TESTING DMV11 (AND Q22 SOFTWARE MODE IS DESIRED)

THE ALLOWABLE OCTAL VALUES OF SR1 ARE: 000000 AND 000001.
A DESCRIPTION OF SR1-SR3 (AND THEIR MEANING) ARE GIVEN BELOW:

FOR SR1 = 000000:
ALL UNITS SELECTED IN DVID1 WILL BE RUN IN POINT-TO-POINT SLAVE,
FULL OR HALF-DUPLEX MODE (DEPENDING ON LINE UNIT SWITCH SETTINGS)
WITHOUT LOOPBACK. THE MODULE COMMUNICATES WITH MASTER MODULE(S)
ON THE SAME AND/OR OTHER PDP-11 SYSTEM(S).
** THIS IS THE DEFAULT MODE OF OPERATION **.
WHEN SR1 = 000000, SR2, SR3, AND SR4 ARE UNUSED, AND
THE HEADER ENTRY DVC MUST BE IN THE RANGE 1-20 (OCTAL).

FOR SR1 = 000001:
ALL UNITS SELECTED IN DVID1 WILL BE RUN IN MULTIPOINT TRIBUTARY,
FULL OR HALF-DUPLEX MODE (DEPENDING ON LINE UNIT SWITCH SETTINGS)
WITHOUT LOOPBACK. THE MODULE COMMUNICATES WITH A MASTER MODULE ON
THE SAME OR DIFFERENT PDP-11 SYSTEM.
WHEN SR1 = 000001, THE FOLLOWING MEANING IS GIVEN TO SR2 AND SR3 :
- SR2 = THE TOTAL NUMBER OF TRIBUTARIES (OCTAL) ON THE MULTIPOINT
LINK WHICH ARE ON THIS CPU. THE ALLOWABLE RANGE IS 000001-000040
(DMP) OR 000001-000020 (DMV).

- SR3 = THE STARTING TRIBUTARY ADDRESS (OCTAL). THE ALLOWABLE RANGE IS 000001-000377. THE PROGRAM WILL USE THIS STARTING TRIB ADRS TO COMPUTE THE OTHER TRIB ADDRESSES ON THE MULTIPOINT LINK, AND THE ADDRESSES CAN "WRAPAROUND" 377 TO 001, IF NECESSARY.

THE HEADER ENTRY DVC MUST BE IN THE RANGE 1-20 (OCTAL).

****
NOTE:: MEMORY RELOCATION IS NOT ALLOWED IN EITHER POINT-TO-POINT OR MULTIPOINT OPERATION. USE EITHER A MODULE THAT DOES NOT ALLOW FOR RELOCATION OR USE THE RUN LOCK COMMAND (RUNL).
****

6. DEVICE/OPTION SETUP

DME REQUIRES NO ADDITIONAL DEVICE SETUP.

7. DME MODULE OPERATION

TEST SEQUENCE:

A. INITIALIZE ALL SLAVE DEVICES
B. LOAD INPUT AND OUTPUT INTERRUPT VECTORS ON ALL SLAVES
C. PERFORM MODE DEFINITION ON ALL SLAVES
D. VERIFY OPERATION OF INPUT AND OUTPUT INTERRUPTS ON ALL SLAVES
E. START THE DDCMP PROTOCOL ON ALL SLAVES, TO ALL MASTERS, AND SCAN EACH LOGICAL LINK UNTIL IT ENTERS THE RUNNING STATE.
F. EXIT TO MONITOR WITH INTERRUPTS ENABLED
G. INPUT INTERRUPT SERVICE :
    1)IF LOGICAL LINK WAS JUST STARTED, INPUT A RCV BUFFER, AND DISABLE INPUTS ON THIS DEVICE.
    2)IF RCV BUFFER WAS PREVIOUSLY INPUT, INPUT A TRANSMIT BUFFER (SEND SAME DATA BACK TO MASTER).
H. OUTPUT INTERRUPT SERVICE :
    1)IF TRANSMIT BUFFER WAS JUST RETURNED, CHECK FOR CORRECT BA, EA, TRANSMIT CHAR COUNT BITS.
    2)IF RCV BUFFER WAS JUST RETURNED, CHECK FOR CORRECT BA, EA, RCV CHAR COUNT BITS, AND RE-ENABLE INPUTS ON THIS DEVICE.
    3)IF ATTEMPT TO RESTART PROTOCOL IS RECEIVED FROM MASTER(S) BY ALL SLAVES, REPORT AN END OF PASS. THEN RESTART THE PROTOCOL ON ALL SLAVES TO THE MASTER(S).
    4)REPORT ANY ERRORS PROVIDED BY ANY SLAVE DEVICE.
    NOTE THAT AT THE END OF EACH PASS THE PROGRAM WILL REPORT ANY SOFT (DATA OR HEADER CRC) ERRORS THAT HAVE OCCURRED DURING THE PASS.
I. REPEAT G AND H

8. OPERATION OPTIONS

A. LOCATION DVID1 (DME 14) MAY BE CHANGED TO SELECT ANY COMBINATION OF DEVICES BIT0=DEV0, BIT1=DEV1 .....BIT15=DEV15.

NOTE: IF DVID1 IS INITIALLY = 0, DME WILL BE DROPPED FROM TEST.

9. ERROR REPORTS

ALL ERROR REPORTS HAVE STANDARD FORMATS AS DESCRIBED IN THE DEC/X11 USER'S GUIDE. IN EACH ERROR PRINTOUT, THE SAME INFORMATION IS ALWAYS REPORTED, ALONG WITH THE ERROR TYPE AND PC, WHICH UNIQUELY IDENTIFY THE ERROR. THE FOLLOWING IS AN EXAMPLE PRINTOUT :

```
DMEA0  PA 00046706  APC 004462  PASS #00001  SOFT ERROR #1
CSRA: 160210  CSRC: 100000  ASTAT: 000602  ERRTYP: 000001
100000  000602  001005  000112  NNNNNN
```

ERRTYP = 1 INDICATES A DATA ERROR (AS DESCRIBED IN THE DEC/X11 USER'S GUIDE), WHICH IS A SOFT ERROR. CSRA IS THE DEVICE CSR ADDRESS, CSRC AND ASTAT ARE THE CONTENTS OF THE FIRST TWO CSR REGISTERS (SEL0 AND SEL2). THE FOUR (5 IF DMV11) FIELDS OF NUMBERS ON THE THIRD LINE ARE THE CONTENTS OF ALL FOUR CSR REGISTERS (SEL0,2,4,6(AND 10 IF DMV)). WHENEVER ERRTYP = 0 IS REPORTED (UNDEFINED ERROR) THE USER MUST REFER TO THE DIAGNOSTIC LISTING FOR THE ACTUAL EXPLANATION OF THE ERROR. THIS IS RECOMMENDED FOR ALL ERRORS, BECAUSE THE DEC/X11 ERROR TYPE DEFINITIONS ARE QUITE GENERAL.

```
                                        .TITLE  DMEB DEC/X11 SYSTEM EXERCISER MODULE
                                ;       DDXCOM  VERSION  6      23-MAY-78
                                        .LIST   BIN
                                ;*****************************************************************************
000000'                         BEGIN:
000000' 046504  041105      040 MODNAM: .ASCII  /DMEB / ;MODULE NAME.
000005'     000                 XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 160170                  ADDR:   160170+0                ;1ST DEVICE ADDR.
000010' 000300                  VECTOR: 300+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' 150000                  STAT:   150000                  ;STATUS WORD.
000030' 000466'                 INIT:   START                   ;MODULE START ADDR.
000032' 000252'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                       ;PASS COUNTER.
000036' 000001                  ICONT:  1                       ;# OF ITERATIONS PER PASS=1
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' 002224'                 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' 000000                  IDNUM:  0                       ;MODULE IDENTIFICATION NUMBER=0
000124' 006244'                 RBUFVA: BUFIN                   ;READ BUFFER VIRTUAL ADDRESS
000126' 000000                  RBUFPA: OPEN                    ;READ BUFFER PHYSICAL ADDRESS
000130' 000000                  RBUFEA: OPEN                    ;READ BUFFER EA BITS
```

```
   292  000132' 001000          RBUFSZ: 512.                    ;SIZE OF THE READ BUFFER
   293  000134' 000000          WBUFPA: OPEN                    ;WRITE BUFFER PHYSICAL ADDRESS
   294  000136' 000000          WBUFEA: OPEN                    ;WRITE BUFFER EA BITS
   295  000140' 001000          WBUFRQ: 512.                    ;WRITE BUFFER SIZE REQUESTED
   296  000142' 000000          WBUFSZ: OPEN                    ;WRITE BUFFER SIZE AVAILABLE
   297  000144' 000000          CDERCT: OPEN                    ;CDATA/DATCK ERROR COUNT
   298  000146' 000000          CDWDCT: OPEN                    ;CDATA/DATCK WORD COUNT
   299  000150' 000000          FREE:   OPEN                    ;RESERVED FOR FUTURE USE
   300  000252'                 MODSP:
   301                          ;*****************************************************************************
```

```
CXDMEB.P11    25-MAR-81 08:25          DEC/X11-1 SYSTEM EXERCISER MACRO DEFINITION MODULE

                                       ;BIT DEFINITIONS
            100000                     BIT15=100000
            040000                     BIT14=40000
            020000                     BIT13=20000
            010000                     BIT12=10000
            004000                     BIT11=4000
            002000                     BIT10=2000
            001000                     BIT9=1000
            000400                     BIT8=400
            000200                     BIT7=200
            000100                     BIT6=100
            000040                     BIT5=40
            000020                     BIT4=20
            000010                     BIT3=10
            000004                     BIT2=4
            000002                     BIT1=2
            000001                     BIT0=1


            000305                     GOODIN  = 305           ;CODE LEFT IN BSEL6 AFTER INIT, IF NO ERRORS

            021344                     RDLU16  = 021344        ;M8207 INSTRUCTION TO MOVE LU REG 16 INTO BSEL4

            121000                     SWPBOT  = 121000        ;DMV-11 ADDRESS OF "SWPBOT" SWITCHES
            000200                     MRDY    = 200           ;DMV-11 BIT INDICATING M-LOOP IS READY FOR NEXT COMMAND
            000001                     REDLOC  = 1             ;DMV-11 M-LOOP COMMAND TO READ INTERNAL MEMORY
            000006                     TTLOOP  = 6             ;DMV-11 M-LOOP COMMAND TO SETUP 56K BPS/INT LOOPBACK

                                       ;*****************************************************************************
                                       ;* SWITCH REGISTER 1 (SR1) BIT DEFINITIONS
                                       ;*****************************************************************************
            000001                     NONLUP  = BIT0  ;LOOPBACK MODE IF BIT = 0

                                       ;*****************************************************************************
                                       ;* SWITCH REGISTER 4 (SR4) BIT DEFINITIONS
                                       ;*****************************************************************************
            000001                     DMVBIT  = BIT0  ;UNIT IS DMV. IF BIT = 1
            000002                     MODE22  = BIT1  ;UNIT IS IN Q22 MODE IF BIT = 1


                                       ;*****************************************************************************
                                       ;* DEC/X11 ERROR CODES
                                       ;*****************************************************************************
            000000                     NOTDFN  = 0     ;ERROR NOT DEFINED
            000001                     DATERR  = 1     ;DATA ERROR
            000007                     SELERR  = 7     ;SELECTION ERROR
            000011                     ILGINT  = 11    ;ILLEGAL INTERRUPT OCCURRED, OR DONE NOT SET
            000023                     NOINTR  = 23    ;DEVICE FAILED TO INTERRUPT
            000034                     NOINIT  = 34    ;DEVICE WILL NOT INITIALIZE
            000036                     BADRED  = 36    ;UNABLE TO EXECUTE READ FUNCTION
            000037                     BADWRT  = 37    ;UNABLE TO EXECUTE WRITE FUNCTION
```



```
CXDMEB.P11    25-MAR-81 08:25          DEC/X11-1 SYSTEM EXERCISER MACRO DEFINITION MODULE

                                       ;*****************************************************************************
                                       ;* OFFSETS FOR DMP/DMV11 CSR ADDRESSES
                                       ;*****************************************************************************
            000000                     SEL0    = 0
            000000                     BSEL0   = 0
            000001                     BSEL1   = 1
            000002                     SEL2    = 2
            000002                     BSEL2   = 2
            000003                     BSEL3   = 3
            000004                     SEL4    = 4
            000004                     BSEL4   = 4
            000005                     BSEL5   = 5
            000006                     SEL6    = 6
            000006                     BSEL6   = 6
            000007                     BSEL7   = 7
            000010                     SEL10   = 10
            000010                     BSEL10  = 10
            000011                     BSEL11  = 11

                                       ;*****************************************************************************
                                       ;* INPUT COMMAND TYPE CODES - BSEL2
                                       ;*****************************************************************************
            000002                     MODDEF  = 2
            000001                     CILIN   = 1
            000004                     BACCIT  = 4
            000000                     BACCIR  = 0


                                       ;*****************************************************************************
                                       ;* OUTPUT COMMAND TYPE CODES - BSEL2
                                       ;*****************************************************************************
            000001                     CTLOUT  = 1
            000004                     BACCOT  = 4
            000000                     BACCOR  = 0
            000003                     BACORU  = 3
            000006                     BACOTS  = 6
            000007                     BACOTN  = 7
            000002                     INFOUT  = 2


                                       ;*****************************************************************************
                                       ;* BIT MASKS FOR FIELDS IN COMMANDS
                                       ;*****************************************************************************
            000370                     CMDMSK  = 370           ;MASK FOR COMMAND TYPE IN BSEL2
            000370                     MODMSK  = 370           ;MASK FOR MODE FIELD IN BSEL6 FOR MODE DEF'N CMND
            000340                     REQMSK  = 340           ;MASK FOR REQUEST KEY IN BSEL6 FOR CTL IN CMND
            000340                     RTNMSK  = 340           ;MASK FOR RETURN KEY IN BSEL6 FOR INFO OUT CMND
            140000                     CCMSK   = 140000        ;MASK FOR CHARACTER COUNT IN SEL6
            000077                     EAMSK   = 077           ;MASK FOR BUS ADRS EA BITS IN BSEL7
            000361                     MODESW  = 361           ;MASK FOR DMP11 MODE SWITCHES IN IBUS REG 16


                                       ;*****************************************************************************
```

```
                        ;* COMMAND CONTROL, INTERRUPT ENABLE BITS - BSEL0
                        ;*****************************************************************************
000200                  RQI     = BIT7
000020                  IEO     = BIT4
000001                  IEI     = BIT0


                        ;*****************************************************************************
                        ;* COMMAND CONTROL BITS - BSEL2
                        ;*****************************************************************************
000200                  RDYO    = BIT7
000020                  RDYI    = BIT4
000010                  Q22BIT  = BIT3


                        ;*****************************************************************************
                        ;* MICROPROCESSOR CONTROL COMMAND - BSEL1
                        ;*****************************************************************************
000200                  RUN     = BIT7
000100                  MCLR    = BIT6
000020                  STEPLU  = BIT4
000010                  LULOOP  = BIT3
000004                  ROMO    = BIT2
000002                  ROM1    = BIT1
000001                  STEPMP  = BIT0


                        ;*****************************************************************************
                        ;* MODE DEFINITION COMMAND - BSEL6
                        ;*****************************************************************************
000000                  MDPPDC  = 0
000001                  FDPPDC  = 1
000002                  MDPP    = 2
000003                  FDPP    = 3
000004                  MDCS    = 4
000005                  FDCS    = 5
000006                  MDTS    = 6
000007                  FDTS    = 7


                        ;*****************************************************************************
                        ;* CONTROL IN COMMAND - BSEL6
                        ;*****************************************************************************
000200                  WRITSS  = BIT7
000100                  RDCTSS  = BIT6
000040                  REDTSS  = BIT5
000001                  ESTRIB  = 01
000002                  KILLTR  = 02
000003                  ISTART  = 03
000004                  MAINT   = 04
000005                  HALTST  = 05
000020                  REDMDM  = 20
000021                  WRIMDM  = 21
000022                  INTFDG  = 22
```



```
000112                  DERDTB  = 112
000113                  DERINB  = 113


                        ;*****************************************************************************
                        ;* CONTROL IN COMMAND - BSEL7
                        ;*****************************************************************************
000200                  DISPIG  = BIT7
000100                  ENBPIG  = BIT6
000040                  LATPS   = BIT5
000020                  ULATPS  = BIT4
000002                  DISCMP  = BIT1
000001                  ENACMP  = BIT0


                        ;*****************************************************************************
                        ;* BUFFER ADDRESS / CHARACTER COUNT IN/OUT COMMAND - SEL4
                        ;*****************************************************************************
100000                  BA15    = BIT15
040000                  BA14    = BIT14
020000                  BA13    = BIT13
010000                  BA12    = BIT12
004000                  BA11    = BIT11
002000                  BA10    = BIT10
001000                  BA9     = BIT9
000400                  BA8     = BIT8
000200                  BA7     = BIT7
000100                  BA6     = BIT6
000040                  BA5     = BIT5
000020                  BA4     = BIT4
000010                  BA3     = BIT3
000004                  BA2     = BIT2
000002                  BA1     = BIT1
000001                  BA0     = BIT0


                        ;*****************************************************************************
                        ;* BUFFER ADDRESS / CHARACTER COUNT IN/OUT - SEL6
                        ;*****************************************************************************
100000                  BA17    = BIT15
040000                  BA16    = BIT14
020000                  CC13    = BIT13
010000                  CC12    = BIT12
004000                  CC11    = BIT11
002000                  CC10    = BIT10
001000                  CC9     = BIT9
000400                  CC8     = BIT8
000200                  CC7     = BIT7
000100                  CC6     = BIT6
000040                  CC5     = BIT5
000020                  CC4     = BIT4
000010                  CC3     = BIT3
000004                  CC2     = BIT2
```

```
        000002          CC1     = BIT1
        000001          CC0     = BIT0



                        ;****************************************************************************
                        ;* CONTROL OUT COMMAND - BSEL6
                        ;****************************************************************************
        000002          RTHRER  = 2
        000004          TTHRER  = 4
        000006          STHRER  = 6
        000010          STARUN  = 10
        000012          MNTRUN  = 12
        000014          MNTHLT  = 14
        000020          RINGDT  = 20
        000022          DEADTR  = 22
        000024          RUNST   = 24
        000026          BABTRB  = 26
        000030          STRTRB  = 30
        000300          BUFTSM  = 300
        000302          NONEXM  = 302
        000304          DISCON  = 304
        000306          QUOVRN  = 306



                        ;****************************************************************************
                        ;* INFORMATION OUT COMMAND - BSEL6
                        ;****************************************************************************
        000100          RDCTSS  = BIT6
        000040          REDTSS  = BIT5
        000000          NORET   = 00
        000010          RETMST  = 10
        000020          BUFRTC  = 20
        000012          EROTB   = 12
        000013          ERINB   = 13



                        ;;***************************************************************************
                        ;* DMP OBUS REG 10 - TRANSMITTER BUFFER
                        ;;***************************************************************************
        000200          TX7     = BIT7
        000100          TX6     = BIT6
        000040          TX5     = BIT5
        000020          TX4     = BIT4
        000010          TX3     = BIT3
        000004          TX2     = BIT2
        000002          TX1     = BIT1
        000001          TX0     = BIT0

                        ;;***************************************************************************
                        ;* DMP OBUS REG 11
                        ;;***************************************************************************
        000200          OC      = BIT7
        000010          GOAH    = BIT3
```



```
        000004          ABORT   = BIT2
        000002          EOM     = BIT1
        000001          SOM     = BIT0

                        ;;***************************************************************************
                        ;* DMP OBUS REG 12
                        ;;***************************************************************************
        000200          IC      = BIT7
        000100          BPOLL   = BIT6
        000040          LOLP    = BIT5

                        ;;***************************************************************************
                        ;* DMP OBUS REG 13
                        ;;***************************************************************************
        000200          POLL    = BIT7
        000100          DIR     = BIT6
        000040          SELFR   = BIT5
        000020          HDX     = BIT4
        000010          MAINT1  = BIT3
        000004          MAINT2  = BIT2
        000002          SELSBY  = BIT1

                        ;;***************************************************************************
                        ;* DMP OBUS REG 14
                        ;;***************************************************************************
        000100          TXEN    = BIT6
        000040          DISSI   = BIT5
        000020          RDAX    = BIT4
        000010          WAX     = BIT3
        000004          ENAX    = BIT2
        000002          AX2     = BIT1
        000001          AX1     = BIT0

                        ;;***************************************************************************
                        ;* DMP OBUS REG 17
                        ;;***************************************************************************
        000200          CRC2    = BIT7
        000100          CRC1    = BIT6
        000040          IDLE    = BIT5
        000020          SECA    = BIT4
        000010          STRIP   = BIT3
        000004          RDALL   = BIT2
        000002          IERR    = BIT1
        000001          DDCMP   = BIT0

                        ;;***************************************************************************
                        ;* DMP IBUS REG 10 - RECEIVER BUFFER
                        ;;***************************************************************************
        000200          RX7     = BIT7
        000100          RX6     = BIT6
        000040          RX5     = BIT5
        000020          RX4     = BIT4
        000010          RX3     = BIT3
        000004          RX2     = BIT2
        000002          RX1     = BIT1
        000001          RX0     = BIT0
```

```
                        ;;*****************************************************************************
                        ;* DMP IBUS REG 12
                        ;;*****************************************************************************
000200                  IC      = BIT7
000100                  IACT    = BIT6
000040                  LULP    = BIT5
000020                  IRDY    = BIT4
000010                  OVRR    = BIT3
000004                  RAB     = BIT2
000002                  EBLK    = BIT1
000001                  BCC     = BIT0

                        ;;*****************************************************************************
                        ;* DMP IBUS REG 13
                        ;;*****************************************************************************
000200                  RING    = BIT7
000100                  DTR     = BIT6
000040                  RTS     = BIT5
000020                  MDX     = BIT4
000010                  MODR    = BIT3
000004                  CS      = BIT2
000002                  STBY    = BIT1
000001                  CARR    = BIT0

                        ;;*****************************************************************************
                        ;* DMP IBUS REG 14
                        ;;*****************************************************************************
000200                  READY   = BIT7
000100                  TXEN    = BIT6
000040                  DISSI   = BIT5
000020                  RDAX    = BIT4
000010                  WAX     = BIT3
000004                  ENAX    = BIT2
000002                  AX2     = BIT1
000001                  AX1     = BIT0

                        ;;*****************************************************************************
                        ;* DMP IBUS REG 16 -- DMV "SWPBUT" (W/BITS RE-ARRANGED)
                        ;;*****************************************************************************
000020                  MODSW0  = BIT4
000040                  MODSW1  = BIT5
000100                  MODSW2  = BIT6
000200                  ENABSW  = BIT7

                        ;;*****************************************************************************
                        ;* DMP IBUS REG 17
                        ;;*****************************************************************************
000200                  SIGR    = BIT7
000100                  SIGD    = BIT6
000040                  TXDATA  = BIT5
000020                  OCDR    = BIT4
000010                  ICIR    = BIT3
000004                  TESTMD  = BIT2
000002                  MCLK    = BIT1
000001                  DOCMP   = BIT0
```



```
                        ;*****************************************************************************
                        ;* INTERRUPT FLAG BIT DEFINITIONS IN "FLAGS"
                        ;*****************************************************************************
000001                  ININT   = BIT0          ; =1 IF INPUT INTRPT OCCURRED
000002                  OUTINT  = BIT1          ; =1 IF OUTPUT INTRPT OCCURRED
000004                  NONQI   = BIT2          ; =1 FOR NON-QUEUED (REQUEST-AND-WAIT) INPUT OPERATION
000010                  NONQO   = BIT3          ; =1 FOR NON-QUEUED (REQUEST-AND-WAIT) OUTPUT OPERATION


                        ;*****************************************************************************
                        ;* INTERRUPT STATUS WORD (INTABL) BIT DEFINITIONS
                        ;*****************************************************************************
000001                  MDFDON  = BIT0          ;MODE DEFINITION DONE
000004                  ISTDON  = BIT2          ;ISTART DONE
000010                  BIRDON  = BIT3          ;BACCIR DONE
000020                  BITDON  = BIT4          ;BACCIT DONE
000040                  BORDON  = BIT5          ;BACCOR DONE
000100                  BOTDON  = BIT6          ;BACCOT DONE


                        ;*****************************************************************************
                        ;* ERROR FLAG BIT DEFINITIONS IN "ERRORS"
                        ;*****************************************************************************
000001                  INTIMO  = BIT0          ;INPUT INTERRUPT TIMED-OUT
000002                  OUTIMO  = BIT1          ;OUTPUT INTERRUPT TIMED-OUT
000004                  BADINI  = BIT2          ;MASTER CLEAR FAILED ON DEVICE


000252' 000000          BUFFEX: .WORD   0       ;ADJUSTED EXTENDED R/W BUFFER ADDRESS BITS

000254' 000000          N.DEVS: .WORD   0       ;NUMBER OF DEVICES PRESENT
000256' 000000          TOTAL:  .WORD   0       ;NUMBER OF DMP11'S CURRENTLY ACTIVE
000260' 000000          COUNT:  .WORD   0       ;ITERATION COUNT FOR MODULE
000262' 000000          SAVBF:  .WORD   0       ;RCV ISR TEMPORARY STORAGE
000264' 000000          SELECT: .WORD   0       ;SOFTWARE BIT MAP OF ACTIVE DEVICES
000266' 000000          DEVPTR: .WORD   0       ;DEVICE SELECTION POINTER
000270' 000000          LUNG16: .WORD   0       ;STORAGE FOR LU IBUS REG 16
000272' 000000          CSEL0:  .WORD   0       ;STORAGE FOR INPUT COMMAND TO BE PERFORMED
000274' 000000          CSEL2:  .WORD   0
000276' 000000          CSEL4:  .WORD   0
000300' 000000          CSEL6:  .WORD   0
000302' 000000          CSEL10: .WORD   0
000304' 000000          RSEL0:  .WORD   0       ;STORAGE FOR CSR'S RETURNED ON OUTPUT CMND
000306' 000000          RSEL2:  .WORD   0
000310' 000000          RSEL4:  .WORD   0
000312' 000000          RSEL6:  .WORD   0
000314' 000000          RSEL10: .WORD   0
000316' 000000          FLAGS:  .WORD   0       ;INPUT AND OUTPUT INTERRUPT FLAG BITS
000320' 000000          ERRORS: .WORD   0       ;ERROR FLAG BITS
000322' 000000          INTIMR: .WORD   0       ;INPUT INTERRUPT TIMER
```

```
000324' 000000                  OUTIMR: .WORD   0        ;OUTPUT INTERRUPT TIMER
000326' 000000                  LLKCNT: .WORD   0        ;COUNT OF LOGICAL LINKS
                                                         ;   IF P-TO-P, THIS IS SAME AS NO. OF DEVICES.
                                                         ;   IF MULTIPOINT, THIS IS SAME AS NO. OF TRIBUTARIES.
000330' 000000                  ISTCNT: .WORD   0        ;COUNT OF ISTART ATTEMPTS ON LOGICAL LINK
000332' 000000                  SCRACH: .WORD   0        ;MISCELLANEOUS STORAGE LOCATION
000334' 000000                  TRBADR: .WORD   0        ;CURRENT TRIB ADDRESS
000336' 000000                  TRBMAX: .WORD   0        ;MAXIMUM TRIB ADDRESS
000340' 000000                  LNKDON: .WORD   0        ;NO. OF LOG. LNKS DONE WITH MSG
000342' 000000                  INDON:  .WORD   0        ;INPUT COUNT OF LOGICAL LINKS DONE WITH MSG
                                                         ;   PER ITERATION
000344' 000000                  OUTDON: .WORD   0        ;OUTPUT COUNT OF LOGICAL LINKS DONE WITH MSG
                                                         ;   PER ITERATION
000346' 000000                  WBFPA:  .WORD   0        ;WRITE BUFFER PHYSICAL ADDRESS
000350' 000000                  WBFEA:  .WORD   0        ;WRITE BUFFER EA BITS

                                ;*****************************************************************************
                                ;* INTABLE - INTERRUPT STATUS TABLE - 16 (MAX.) 2-WORD ENTRIES, 1 ENTRY
                                ;*  PER LOGICAL LINK.
                                ;*
                                ;* IN EITHER POINT-TO-POINT OR MULTIPOINT TRIB MODE, UP TO 16 ENTRIES CAN BE
                                ;*      USED, WHERE AN ENTRY CONSISTS OF 16 INTERRUPT STATUS FLAG BITS,
                                ;*      FOLLOWED BY A WORD CONTAINING THE TRIB ADDRESS IN THE LOWER BYTE.
                                ;*  IN ANY CASE, THE NO. OF OCCUPIED ENTRIES = THE NO. OF LOGICAL LINKS,
                                ;*     ON THIS PDP-11 SYSTEM, KEPT IN LLKCNT.
                                ;*****************************************************************************
000352' 000040                  INTABL: .BLKW   32.


                                ; REGISTER ADDRESS TABLE FOR ERROR REPORTS
000452' 160170                  REGTBL: .WORD   160170
000454' 160172                          .WORD   160172
000456' 160174                          .WORD   160174
000460' 160176                          .WORD   160176
000462' 160200                          .WORD   160200
000464' 177777                          .WORD   177777
```



```
                                ;-----------------------------------------------------------------------------
                                ; THIS IS START OF MODULE
                                ;-----------------------------------------------------------------------------
000466' 016767  177322  177570  START:  MOV     DVIDI,SELECT    ;GET ACTIVE DEVICES
000474' 001004                          BNE     SETUP           ;BR IF ANY ARE SELECTED
000476' 004767  004424          DRPMOD: JSR     PC,DISABL       ;DISABLE INTERRUPTS ON ALL SELECTED DEVICES
000502' 104410  000000'                 ENDS,BEGIN              ;;DROP THE DEVICE MODULE

                                ;-----------------------------------------------------------------------------
                                ;SETUP VECTORS FOR ACTIVE DEVICES, AND INITIALIZE THEM
                                ;-----------------------------------------------------------------------------
000506'                         SETUP:
000506' 005067  177542                  CLR     N.DEVS          ;CLEAR NO. OF DEVICES
000512' 005067  177602                  CLR     ERRORS          ;CLEAR ERROR FLAGS
000516' 012767  000010  177572          MOV     #NONQU,FLAGS    ;SET FLAG FOR NON-QUEUED OUTPUT INTRPTS
000524' 004767  004354                  JSR     PC,CLRCMD       ;CLEAR COMMAND STORAGE AREA
000530' 012704  000352'                 MOV     #INTABL,R4      ;INIT POINTER TO INTRPT STATUS TABLE
000534' 005024                  1$:     CLR     (R4)+           ;CLEAR A TABLE WORD
000536' 020427  000452'                 CMP     R4,#INTABL+64.  ;SEE IF ALL ENTRIES CLEARED YET
000542' 103774                          BLO     1$              ;BR IF NOT YET
000544' 016702  177240                  MOV     VECTOR,R2       ;GET INITIAL VECTOR
000550' 016700  177232                  MOV     ADDR,R0         ;GET INITIAL ADDRESS
000554' 012703  005304'                 MOV     #11ISR0,R3      ;INIT ISR POINTER
000560' 012767  000001  177500          MOV     #BIT0,DEVPTR    ;INIT SELECTION POINTER
000566' 005067  177534                  CLR     LLKCNT          ;INIT LOGICAL LINK CNT
                                ;FIND A SELECTED DEVICE
000572' 036767  177470  177464  2$:     BIT     DEVPTR,SELECT   ;SEE IF THIS DEVICE IS SELECTED
000600' 001021                          BNE     5$              ;BR IF SELECTED
000602' 006367  177460          3$:     ASL     DEVPTR          ;SHIFT SELECTION POINTER
000606' 001002                          BNE     4$              ;BR IF NOT ALL DEVICES SCANNED YET
000610' 000167  000410                  JMP     STRTUP          ;ALL DEVICES SCANNED - START LINKS
000614' 062703  000036          4$:     ADD     #36,R3          ;POP ISR POINTER
000620' 062702  000010                  ADD     #10,R2          ;POP VECTOR
000624' 005767  177174                  TST     SR4             ; IS THIS A DMV?
000630' 001402                          BEQ     25$
000632' 062700  000010                  ADD     #10,R0          ;IF YES: POP DMV11 ADDRESS (20)
000636' 062700  000010          25$:    ADD     #10,R0          ;IF NO: POP DMP11 ADDRESS (10)
000642' 000753                          BR      2$              ;CONTINUE
                                ; MASTER CLEAR THIS DEVICE
000644' 112760  000100  000001  5$:     MOVB    #MCLR,BSEL1(R0) ;SET MASTER CLEAR
000652' 005767  177146                  TST     SR4             ; IS THIS A DMV?
000656' 001011                          BNE     23$             ; IF YES: SKIP LULOOP/RUN STUFF
000660' 112760  000200  000001          MOVB    #RUN,BSEL1(R0)  ;SET RUN BIT
000666' 142760  000200  000001          BICB    #RUN,BSEL1(R0)  ;CLEAR RUN
000674' 152760  000010  000001          BISB    #LULOOP,BSEL1(R0) ;SET LULOOP
000702' 004767  003714          23$:    JSR     PC,READ16       ;READ LU REG 16 FOR SWITCHES
000706' 004767  003530                  JSR     PC,INIDMP       ;MASTER CLEAR DEVICE
000712' 132767  000004  177400          BITB    #BADINI,ERRORS  ;SEE IF INIT WAS BAD
000720' 001070                          BNE     8$              ;IF INIT BAD, GO DROP MODULE
                                ; LOAD INPUT AND OUTPUT VECTORS FOR THIS DEVICE
000722' 010312                          MOV     R3,(R2)         ;LOAD INPUT ISR VECTOR
000724' 116762  177062  000002          MOVB    BR1,2(R2)       ;LOAD PRIORITY
000732' 105062  000003                  CLRB    3(R2)           ;CLEAR HI BYTE OF PROGRAM STATUS WORD
000736' 010063  000034                  MOV     R0,34(R3)       ;LOAD DMP11 ADDRESS
```

```
                        MOV     R3,4(R2)        ;LOAD OUTPUT ISR VECTOR
                        ADD     #16,4(R2)       ;POINT TO OUTPUT ISR
                        MOVB    BR1,6(R2)       ;LOAD PRIORITY
                        CLRB    7(R2)           ;CLEAR HI BYTE OF STATUS WORD
                ;DEFINE THE MODE OF THIS DEVICE (FDX, HDX, P-P, M-P TRIB)
                        MOVB    #MODDEF,CSEL2   ;SET MODE DEFINITION COMMAND
                        BISB    #LULOOP,BSEL1(R0) ;SET LULOOP TO FORCE MODE
                        CMPB    SR1,#0          ;SEE IF GRP OF POINT-TO-POINT DEVICES
                        BNE     9$              ;BR IF NOT
                        BITB    #ENABSW,LURG16  ;SEE IF LU SWITCHES ENABLED
                        BEQ     6$              ;BR IF NOT
                        BITB    #MODSW0,LURG16  ;SEE IF HDX IN SWITCHES
                        BEQ     6$              ;BR IF HDX
                        MOVB    #FDPP,CSEL6     ;SET FDX P-TO-P MODE
                        BR      7$              ;GO SET MODE
                6$:     MOVB    #HDPP,CSEL6     ;SET HDX P-TO-P MODE
                7$:     JSR     PC,INCMND       ;PERFORM MODE DEF'N CMND
                        BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
                        BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
                        BICB    #LULOOP,BSEL1(R0) ;CLEAR LULOOP
                        BIT     #INTIMO,ERRORS  ;SEE IF INPUT INTRPT TIMED-OUT
                        BEQ     12$             ;BR IF NO TIME-OUT
                8$:     BICB    #RUN,BSEL1(R0)  ;CLEAR RUN ON THIS DEVICE
                        JMP     DRPMOD          ;GO DROP MODULE
                9$:     CMPB    SR1,#1          ;SEE IF GRP OF M-P TRIBUTARIES
                        BNE     11$             ;BR IF NOT
                        BITB    #ENABSW,LURG16  ;SEE IF LU SWITCHES ENABLED
                        BEQ     10$             ;BR IF NOT
                        BITB    #MODSW0,LURG16  ;SET HDX IN SWITCHES
                        BEQ     10$             ;BR IF HDX
                        MOVB    #FDTS,CSEL6     ;SET FDX TRIB MODE
                        BR      7$              ;GO SET MODE
                10$:    MOVB    #HDTS,CSEL6     ;SET HDX TRIB MODE
                        BR      7$              ;GO SET MODE
                11$:    JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
                        MOV     #SELERR,ERRTYP  ;SET CODE FOR SELECTION ERROR
                        ;*************************************************************
                        HRDER$,BEGIN,NULL       ;OPERATOR SPEC'D INVALID SR1 VALUE
                        ;*************************************************************
                        JMP     DRPMOD          ;GO DROP THE MODULE
                12$:    INC     N.DEVS          ;UPDATE THE NO. OF DEVICES TO RUN
                        CLRB    BSEL0(R0)       ;DISABLE INTRPTS FROM THIS DEVICE
                        JMP     3$              ;CONTINUE SETUP

                ;-------------------------------------------------------------------------
                ; ATTEMPT TO START THE PROTOCOL ON EACH LOGICAL LINK, AND IF A LOGICAL LINK
                ; CAN'T BE STARTED, DROP THE DEVICE WHICH CONTROLS THAT LINK, FROM TESTING
                ;-------------------------------------------------------------------------
                STRTUP:
                        MOV     N.DEVS,LLKCNT   ;SET TOTAL NO. OF LOGICAL LINKS
                        MOV     #1,TRBADR       ;SET P-TO-P TRIB ADRS = 1
                        MOV     #INTABL+2,R5    ;INIT STATUS TABLE TRIB ADRS POINTER
                        CMPB    SR1,#1          ;SEE IF MULTIPOINT
                        BNE     1$              ;BR IF NOT MULTIPOINT
                        MOV     SR3,TRBADR      ;INIT MULTIPOINT TRIB ADRS
                1$:     MOV     ADDR,R0         ;INIT DEVICE ADRS
```



```
                        MOV     #BIT0,DEVPTR    ;INIT SELECTION POINTER
                ;FIND A SELECTED DEVICE
                2$:     BIT     DEVPTR,SELECT   ;SEE IF THIS DEVICE IS SELECTED
                        BNE     22$             ;BR IF SELECTED
                20$:    ASL     DEVPTR          ;SHIFT SELECTION POINTER
                        BNE     21$             ;BR TO KEEP SCANNING
                        JMP     19$             ;ALL DEVICES SCANNED - PROCEED

                21$:    TST     SR4             ; IS THIS A DMV ?
                        BEQ     25$
                        ADD     #10,R0          ;IF YES: INCREMENT DMV11 ADDRESS (20)
                25$:    ADD     #10,R0          ;IF NO: INCREMENT DMP11 ADDRESS (10)
                        ADD     #4,R5           ;INCR STATUS TABLE TRIB ADRS POINTER
                        BR      2$              ;CONTINUE
                ;SET UP FOR CONTROL IN COMMAND
                22$:    MOVB    #CTLIN,CSEL2    ;SET CONTROL IN CMND
                        MOVB    TRBADR,CSEL2+1  ;SET TRIB ADRS
                        MOV     TRBADR,(R5)     ;STORE THE TRIB ADRS IN STATUS TABLE
                        CLR     CSEL4
                        CLR     CSEL6
                ;ESTABLISH A LOGICAL LINK
                        MOVB    #ESTRIB,CSEL6   ;SET CODE FOR ESTABLISH TRIB CMND
                        BIC     #OUTINT,FLAGS   ;CLEAR OUTPUT INTRPT FLAG
                        JSR     PC,INCMND       ;ESTABLISH THIS TRIBUTARY
                ;CHANGE THE SELECTION INTERVAL TO 5 SECONDS
                        TST     SR4             ; IS THIS A DMV ?
                        BEQ     23$
                        MOV     #500.,CSEL4     ;IF YES: CONSTANT = 500.
                        BR      24$
                23$:    MOV     #5000.,CSEL4    ;IF NO: CONSTANT = 5000.
                24$:    MOV     #WRITSS+36,CSEL6 ;WRITE TSS ADDRESS 36 (SEL INTERVAL).
                        BIC     #OUTINT,FLAGS   ;CLEAR OUTPUT INTRPT FLAG.
                        JSR     PC,INCMND       ;CHANGE THE INTERVAL.
                ;ISTART THE LOGICAL LINK
                        CLR     CSEL4           ;CLEAR INTERVAL.
                        MOV     #0.,ISTCNT      ;INIT ISTART COUNTER FOR APPROX. 5 MINUTES
                                                ;   STARTUP (EACH COUNT = 35. SEC.)
                3$:     MOVB    #ISTART,CSEL6   ;SET CODE FOR ISTART REQUEST
                        BIC     #OUTINT,FLAGS   ;CLEAR OUTPUT INTRPT FLAG
                        JSR     PC,INCMND       ;PERFORM ISTART CMND
                        MOV     #-1,OUTIMR      ;INIT OUTPUT INTERRUPT TIMER
                4$:
                        BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
                        BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
                        BITB    #OUTINT,FLAGS   ;SEE IF OUTPUT INTRPT SERVICED YET
                        BNE     5$              ;BR IF YES
                        DEC     OUTIMR          ;DECREMENT TIMER
                        BNE     4$              ;BR IF OUTPUT INTRPT DIDN'T TIME-OUT YET
                        JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
                        MOV     #NOINTR,ERRTYP  ;CODE FOR DEVICE FAILED TO INTRPT
                        ;*************************************************************
                        HRDER$,BEGIN,REGTBL     ;OUTPUT INTERRUPT TIMED-OUT
                        ;*************************************************************
                        JMP     14$             ;GO DROP MODULE
                5$:     BICB    #CMDMSK,RSEL2   ;MASK FOR OUTPUT COMMAND FIELD
                        CMPB    RSEL2,#CTLOUT   ;CHK FOR CTL OUT CMND RETURNED
```

```
                        BEQ     6$              ;BR IF YES
001604' 001412          JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
001606' 004767 003356   MOV     #NOTDFN,ERRTYP  ;CODE FOR ERROR NOT DEFINED (MUST INTERPRET
001612' 012767 000000 176266    ;               ;   CSR CONTENTS)
                        ;***************************************************************
001620' 104405 000000' 000452'  HRDERS,BEGIN,REGTBL     ;UNEXPECTED OUTPUT CMND
                        ;***************************************************************
001626' 000167 000316   JMP     14$             ;GO DROP MODULE
                    ;CHECK FOR LOGICAL LINK RUNNING YET
001632' 126727 176454 000024 6$:    CMPB    RSEL6,#RUNST    ;SEE IF PROTOCOL RUNNING ON THIS LOG LINK YET
001640' 001002          BNE     7$              ;BR IF NOT
001642' 000167 000312   JMP     15$             ;GO SELECT NEXT LOGICAL LINK
                    ;CHECK FOR TRANSMIT THRESHOLD ERROR
001646' 126727 176440 000004 7$:    CMPB    RSEL6,#TTHRER   ;SEE IF TRANSMIT THRESHOLD ERROR (TRIB NOT RESPONDING)
001654' 001416          BEQ     8$              ;BR IF YES
001656' 126727 176430 000006    CMPB    RSEL6,#STHRER   ;SEE IF SELECT THRESHOLD ERROR
001664' 001412          BEQ     8$              ;BR IF YES
001666' 004767 003276   JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
001672' 012767 000000 176206    MOV     #NOTDFN,ERRTYP  ;CODE FOR ERROR NOT DEFINED (MUST INTERPRET
                        ;               ;   CSR CONTENTS)
                        ;***************************************************************
001700' 104405 000000' 000452'  HRDERS,BEGIN,REGTBL     ;UNEXPECTED CTL OUT CMND
                        ;***************************************************************
001706' 000167 000236   JMP     14$             ;GO DROP MODULE
001712' 005367 176412 8$:   DEC     ISTCNT          ;DECREMENT ISTART COUNTER
001716' 001012          BNE     9$              ;BR IF COUNT NOT OVERFLOWED YET
001720' 004767 003244   JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
001724' 012767 000000 176154    MOV     #NOTDFN,ERRTYP  ;CODE FOR ERROR NOT DEFINED (MUST INTERPRET
                        ;               ;   CSR CONTENTS)
                        ;***************************************************************
001732' 104405 000000' 000452'  HRDERS,BEGIN,REGTBL     ;TIMED-OUT ATTEMPTING TO START LINK
                        ;***************************************************************
001740' 000167 000204   JMP     14$             ;GO DROP MODULE
                    ;HALT PROTOCOL PRIOR TO ISTART RETRY
001744' 112767 000005 176326 9$:    MOVB    #HALTST,CSEL6   ;SET CODE FOR HALT REQUEST
001752' 042767 000002 176336    BIC     #OUTINT,FLAGS   ;CLEAR OUTPUT INTRPT FLAG
001760' 004767 002756   JSR     PC,INCMND       ;PERFORM HALT CMND TO RE-INIT LINK STATUS
001764' 012767 020000 176332    MOV     #20000,OUTIMR   ;INIT OUTPUT INTERRUPT TIMER
001772'                 10$:
001772' 104407 000000'          BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
001776' 104407 000000'          BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002002' 032767 000002 176306    BIT     #OUTINT,FLAGS   ;SEE IF OUTPUT INTRPT SERVICED YET
002010' 001015          BNE     11$             ;BR IF YES
002012' 005367 176306   DEC     OUTIMR          ;DECREMENT TIMER
002016' 001365          BNE     10$             ;BR IF OUTPUT INTRPT DIDN'T TIME-OUT YET
002020' 004767 003144   JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
002024' 012767 000023 176054    MOV     #NOINTR,ERRTYP  ;CODE FOR DEVICE FAILED TO INTRPT
                        ;***************************************************************
002032' 104405 000000' 000452'  HRDERS,BEGIN,REGTBL     ;OUTPUT INTERRUPT TIMED-OUT
                        ;***************************************************************
002040' 000167 000104   JMP     14$             ;GO DROP MODULE
002044' 142767 000370 176234 11$:   BICB    #CMDMSK,RSEL2   ;MASK FOR OUT CMND RETURNED
                    ;CHECK FOR INFO OUT WITH "BUFFERS RETURNED" CODE
002052' 126727 176230 000002    CMPB    RSEL2,#INFOUT   ;CHK FOR INFO OUT CMND RETURNED
002060' 001022          BNE     13$             ;BR IF NOT
002062' 142767 000340 176222    BICB    #RTNMSK,RSEL6   ;MASK FOR RETURN KEY
```



```
002070' 126727 176216 000020    CMPB    RSEL6,#BUFRTC   ;CHK FOR BUFFER RETURN COMPLETE
002076' 001002          BNE     12$             ;BR IF NOT BUFFER RETURN COMPLETE
002100' 000167 177364   JMP     3$              ;GO TRY TO ISTART AGAIN
002104' 004767 003060 12$:  JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
002110' 012767 000000 175770    MOV     #NOTDFN,ERRTYP  ;CODE FOR ERROR NOT DEFINED (MUST INTERPRET
                        ;               ;   CSR CONTENTS)
                        ;***************************************************************
002116' 104405 000000' 000452'  HRDERS,BEGIN,REGTBL     ;UNEXPECTED INFO OUT CMND
                        ;***************************************************************
002124' 000411          BR      14$             ;GO DROP MODULE
002126' 042767 000002 176162 13$:   BIC     #OUTINT,FLAGS   ;CLEAR OUTPUT INTRPT FLAG
002134' 012767 020000 176162    MOV     #20000,OUTIMR   ;INIT OUTPUT INTRPT TIMER
002142' 105060 000002   CLRB    BSEL2(R0)       ;RELEASE THE PORT AGAIN
002146' 000711          BR      10$             ;KEEP LOOKING FOR INFO OUT CMND
002150' 005060 000000 14$:  CLR     SEL0(R0)        ;CLEAR RUN, DISABLE INTRPTS ON THIS DEVICE
002154' 000167 176316   JMP     DRPMOD          ;GO DROP MODULE
                    ;SELECT NEXT LOGICAL LINK TO START
002160' 105060 000002 15$:  CLRB    BSEL2(R0)       ;CLEAR BSEL2 (INCLUDING RDYO)
002164' 126727 175626 000001    CMPB    SR1,#1          ;SEE IF MULTIPOINT
002172' 001007          BNE     16$             ;BR IF NOT MULTIPOINT
002174' 005267 176134   INC     TRBADR          ;INCREMENT TRIB ADRS
002200' 105767 176130   TSTB    TRBADR          ;SEE IF TRIB ADRS OVERFLOW
002204' 001002          BNE     16$             ;BR IF NOT
002206' 005267 176122   INC     TRBADR          ;IF TRIB ADRS = 0, MAKE IT 1
                        ;               ;   TRIB ADRS = 0 IS ILLEGAL
002212' 105060 000000 16$:  CLRB    BSEL0(R0)       ;DISABLE INTRPTS FROM THIS DEVICE
002216' 000167 177062   JMP     20$             ;GO START NEXT LOGICAL LINK
002222' 000413          19$:    BR      REENTR          ;GO BEGIN TESTING ON ALL LOGICAL LINKS
                        ;-------------------------------------------------------------------------
                        ; RESTART ADDRESS FOR A PASS ON ALL DEVICES
                        ;-------------------------------------------------------------------------
002224' 012705 000020 RESTRT: MOV   #16.,R5           ;INIT INTRPT STATUS TABLE ENTRY COUNTER
002230' 012704 000352'  MOV     #INTABL,R4      ;INIT TABLE POINTER
002234' 012714 000001 1$:   MOV     #MDFDON,(R4)    ;SET MODE DEF'N DONE FLAG IN STATUS BYTE
002240' 062704 000004   ADD     #4,R4           ;INCR STATUS TABLE INDEX
002244' 005305          DEC     R5              ;DECR TABLE ENTRY COUNTER
002246' 001372          BNE     1$              ;BR IF MORE TO DO
002250' 000412          BR      ENABLE

                        ;-------------------------------------------------------------------------
                        ; PROGRAM ONLY COMES HERE AT START OF FIRST PASS
                        ;-------------------------------------------------------------------------
002252' 012705 000020 REENTR: MOV   #16.,R5           ;INIT INTRPT STATUS TABLE ENTRY COUNTER
002256' 012704 000352'  MOV     #INTABL,R4      ;INIT TABLE POINTER
002262' 012714 000005 1$:   MOV     #MDFDON!1STDUN,(R4)     ;INIT DONE FLAGS FOR ITERATION
002266' 062704 000004   ADD     #4,R4           ;INCR STATUS TABLE INDEX
002272' 005305          DEC     R5              ;DECREMENT COUNTER
002274' 001372          BNE     1$              ;BR IF MORE TO DO

                        ;-------------------------------------------------------------------------
                        ;GET A WRITE BUFFER, ENABLE INTERRUPTS, REQUEST INPUTS ON ALL DEVICES
                        ;-------------------------------------------------------------------------
002276' 005067 176040 ENABLE: CLR   INDON           ;INIT INPUT DONE COUNT TO 0
002302' 005067 176036   CLR     OUTDON          ;INIT OUTPUT DONE COUNT TO 0
002306' 104415 000000' 000124'  GETPA$,BEGIN, RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
002314' 016767 175610 175730    MOV     RBUFEA,BUFFEX
```

```
1069  002322' 006367  175724                ASL     BUFFEX          ;GET RCV BUF EA BITS INTO BITS 6,7
1070  002326' 006367  175720                ASL     BUFFEX
1071  002332' 032767  000002  175464        BIT     #MODE22,SR4     ;* DMV IN Q22 MODE ??
1072  002340' 001406                        BEQ     5$              ;* NO: DONE ADJUSTING...
1073  002342' 006367  175704                ASL     BUFFEX          ;* YES: ADJUST FOR Q22 MODE
1074  002346' 006367  175700                ASL     BUFFEX          ;*
1075  002352' 000367  175674                SWAB    BUFFEX          ;*
1076  002356' 016767  175544  175762  5$:   MOV     RBUFPA,#BFPA    ;SET UP SAME BUFFER FOR READ AND WRITE
1077  002364' 016767  175662  175756        MOV     BUFFEX,#BFEA
1078  002372' 016700  175410                MOV     ADDR,R0         ;INIT DEVICE ADDRESS
1079  002376' 012767  000001  175662        MOV     #BIT0,DEVPTR    ;INIT SELECTION POINTER
1080  002404' 042767  000010  175704        BIC     #NONQU,FLAGS    ;CLEAR FLAG TO GET QUEUED OUTPUT INTRPTS AGAIN
1081  002412' 036767  175650  175644  1$:   BIT     DEVPTR,SELECT   ;SEE IF THIS DEVICE IS SELECTED
1082  002420' 001403                        BEQ     2$              ;BR IF NOT SELECTED
1083  002422' 112760  000221  000000        MOVB    #RQI!IE!!IEO,BSEL0(R0) ;ENABLE INTRPTS, SET RQI ON THIS DEVICE
1084  002430' 005767  175370        2$:     TST     SR4             ; IS THIS A DMV ?
1085  002434' 001402                        BEQ     3$
1086  002436' 062700  000010                ADD     #10,R0          ;IF YES: INCREMENT DEVICE ADDRESS (20)
1087  002442' 062700  000010        3$:     ADD     #10,R0          ;IF NO: INCREMENT DMP11 ADDRESS (10)
1088  002446' 006367  175614                ASL     DEVPTR          ;SHIFT SELECTION POINTER
1089  002452' 001357                        BNE     1$              ;BR IF NOT ALL ENABLED YET
1090  002454'                       RTNMON:
1091  002454' 104400  000000'               EXIT$,BEGIN                     ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.
1092
1093
1094
```



```
1095
1096
1097
1098
1099
1100                                  ;*****************************************************************************
1101                                  ;* INISR - THIS IS THE INPUT INTERRUPT SERVICE ROUTINE.
1102                                  ;*       WHEN THE INPUT INTERRUPT OCCURS, THE PROGRAM LOADS THE APPROPRIATE
1103                                  ;*       INPUT COMMAND INTO THE DEVICE CSR'S, AND MONITORS THE PROGRESS OF THE
1104                                  ;*       MODULE ON THE CURRENT ITERATION.
1105                                  ;*****************************************************************************
1106  002460'                         INISR:
1107  002460' 142760  000021  000000        BICB    #IE!!IEO,BSEL0(R0) ;DISABLE INTERRUPTS
1108  002466' 032767  000004  175622        BIT     #NONQI,FLAGS    ;SEE IF NON-QUEUED INPUT OPERATION REQUESTED
1109  002474' 001421                        BEQ     1$              ;BR FOR QUEUED OPERATION
1110  002476' 116760  175573  000003        MOVB    CSEL2+1,BSEL3(R0) ;LOAD BSEL3
1111  002504' 016760  175566  000004        MOV     CSEL4,SEL4(R0)  ;LOAD SEL4
1112  002512' 016760  175562  000006        MOV     CSEL6,SEL6(R0)  ;LOAD SEL6
1113  002520' 152767  000001  175570        BISB    #ININT,FLAGS    ;SET INPUT INTRPT FLAG
1114  002526' 142760  000200  000000        BICB    #RQI,BSEL0(R0)  ;CLEAR RQI
1115  002534' 000167  000416                JMP     11$
1116  002540' 010004                1$:     MOV     R0,R4           ;GET DEVICE ADDRESS
1117  002542' 166704  175240                SUB     ADDR,R4         ;COMPUTE INTRPT STATUS TABLE INDEX
1118  002546' 005767  175252                TST     SR4             ; IS THIS A DMV?
1119  002552' 001401                        BEQ     8$
1120  002554' 006204                        ASR     R4
1121  002556' 006204                8$:     ASR     R4              ;IF NO: DIVIDE BY 2 FOR INDEX (DMP)
1122  002560' 016467  000354' 175546        MOV     INTABL+2(R4),TRBADR ;GET TRIB ADRS FROM TABLE
1123  002566' 032764  000020  000352' 2$:    BIT     #BITDON,INTABL(R4) ;SEE IF BACCIT JUST COMPLETED
1124  002574' 001031                        BNE     4$              ;GO HANDLE UNSOLICITED INTERRUPT
1125  002576' 032764  000010  000352'        BIT     #BINDON,INTABL(R4) ;SEE IF BACCIR JUST COMPLETED
1126  002604' 001117                        BNE     7$              ;BR IF YES, TO ISSUE BACCIT
1127  002606' 032764  000004  000352'        BIT     #ISTDON,INTABL(R4) ;SEE IF ISTART JUST COMPLETED
1128  002614' 001053                        BNE     6$              ;BR IF YES, TO ISSUE BACCIR
1129  002616' 032764  000001  000352'        BIT     #MDFDON,INTABL(R4) ;SEE IF MODE DEFINITION JUST COMPLETED
1130  002624' 001026                        BNE     5$              ;BR IF YES, TO ISSUE ISTART
1131  002626' 004767  002336                JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
1132  002632' 012767  000011  175246        MOV     #ILGINT,ERRTYP  ;CODE FOR ILLEGAL INTRPT OCCURRED
1133                                        ;*****************************************************************
1134  002640' 104405  000000' 000452'      HRDER$,BEGIN,REGTBL     ;UNSOLICITED INPUT INTERRUPT OCCURRED
1135                                        ;*****************************************************************
1136  002646' 142760  000201  000000 3$:    BICB    #RQI!IEI,BSEL0(R0)      ;UNSOLICITED INPUT INTRPT - CLR RQI,IEI
1137  002654' 000167  177574                JMP     RTNMON          ;RETURN TO DEC/X11 MONITOR TO AWAIT INTERRUPT
1138  002660' 004767  002304        4$:     JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
1139  002664' 012767  000011  175214        MOV     #ILGINT,ERRTYP  ;CODE FOR ILLEGAL INTRPT OCCURRED
1140                                        ;*****************************************************************
1141  002672' 104405  000000' 000452'      HRDER$,BEGIN,REGTBL     ;UNSOLICITED INPUT INTERRUPT OCCURRED
1142                                        ;*****************************************************************
1143  002700' 000762                        BR      3$
1144                                ;ISSUE ISTART CTL IN
1145  002702' 112767  000001  175364 5$:    MOVB    #CTLIN,CSEL2    ;SET UP CONTROL IN COMMAND
1146  002710' 005060  000004                CLR     SEL4(R0)        ;CLEAR SEL4
1147  002714' 005060  000006                CLR     SEL6(R0)        ;CLEAR SEL6
1148  002720' 112760  000003  000006        MOVB    #ISTART,BSEL6(R0) ;SET ISTART STATE
1149  002726' 052764  000004  000352'       BIS     #ISTDON,INTABL(R4) ;SET ISTART DONE IN STATUS WORD
1150  002734' 142760  000200  000000        BICB    #RQI,BSEL0(R0)  ;DISABLE INPUTS TO AWAIT OUTPUT
```

```
1151  002742' 000502                          BR      10$
1152                                  ;ISSUE BACCIR
1153  002744' 112767  000000  175322  6$:     MOVB    #BACCIR,CSEL2   ;SET UP BACCIR COMMAND
1154  002752' 016760  175150  000004          MOV     RBUFPA,SEL4(R0) ;LOAD RCV BUF ADRS LO BITS
1155  002760' 012760  002000  000006          MOV     #1024.,SEL6(R0) ;SET RCV CHAR COUNT = 1024.
1156  002766' 156760  175260  000007          BISB    BUFFEX,BSEL7(R0) ;LOAD RCV BUF EA BITS
1157  002774' 032767  000002  175022          BIT     #MODE22,SR4     ;* IS THIS A DMV IN Q22 MODE
1158  003002' 001411                          BEQ     10$             ;* IF YES:
1159  003004' 052767  000010  175262          BIS     #Q22BIT,CSEL2   ;*  SET Q22 MODE BIT
1160  003012' 012760  002000  000010          MOV     #1024.,SEL10(R0);*  SET RX CHAR CNT=1024.
1161  003020' 016760  175226  000006          MOV     BUFFEX,SEL6(R0) ;*  LOAD RCV BUF EA BITS
1162  003026' 052764  000010  000352' 16$:    BIS     #BIRDUN,INTABL(R4)  ;  SET BACCIR DONE IN STATUS WORD
1163  003034' 142760  000200  000000          BICB    #RQI,BSEL0(R0)      ;DISABLE INPUT REQUESTS, UNTIL BUFFER IS RCV'D
1164  003042' 000442                          BR      10$
1165                                  ;ISSUE BACCIT
1166  003044' 112767  000004  175222  7$:     MOVB    #BACCIT,CSEL2   ;SET UP BACCIT COMMAND
1167  003052' 016760  175270  000004          MOV     #BFPA,SEL4(R0)  ;LOAD TX BUFFER ADDRESS LO BITS
1168  003060' 012760  002000  000006          MOV     #1024.,SEL6(R0) ;SET TX CHAR CNT = 1024.
1169  003066' 156760  175256  000007          BISB    #BFEA,BSEL7(R0) ;LOAD TX BUF EA BITS
1170  003074' 032767  000002  174722          BIT     #MODE22,SR4     ;* IS THIS A DMV IN Q22 MODE
1171  003102' 001411                          BEQ     12$             ;* IF YES:
1172  003104' 052767  000010  175162          BIS     #Q22BIT,CSEL2   ;*  SET Q22 MODE BIT
1173  003112' 012760  002000  000010          MOV     #1024.,SEL10(R0);*  SET RX CHAR CNT=1024.
1174  003120' 016760  175224  000006          MOV     #BFEA,SEL6(R0)  ;*  LOAD RCV BUF EA BITS
1175  003126'                         12$:                            ;* IF NO: LEAVE ALONE.
1176  003126' 052764  000020  000352'         BIS     #BITDUN,INTABL(R4)  ;SET BACCIT DONE IN STATUS WORD
1177  003134' 142760  000200  000000          BICB    #RQI,BSEL0(R0)  ;DISABLE INPUT REQUESTS, UNTIL BUFFER IS TX'D
1178  003142' 005267  175174                  INC     INDUN           ;INCR NO. OF LOGICAL LNKS DONE WITH MSG
1179  003146' 000400                          BR      10$
1180                                  ;LOAD TRIB ADRS AND COMMAND INTO CSR'S
1181  003150' 116760  175160  000003  10$:    MOVB    TRBADR,BSEL3(R0) ;LOAD TRIB ADRS
1182  003156' 116760  175112  000002  11$:    MOVB    CSEL2,BSEL2(R0) ;LOAD COMMAND, CLEAR RDYI
1183
1184  003164' 132760  000020  000002  CHKMOR: BITB    #RDYI,BSEL2(R0) ;SEE IF RDYI SET AGAIN
1185  003172' 001402                          BEQ     1$              ;BR IF RDYI NOT SET AGAIN
1186  003174' 000167  177260                  JMP     INISR           ;GO HANDLE NEXT INPUT
1187  003200' 132760  000200  000002  1$:     BITB    #RDYO,BSEL2(R0) ;SEE IF AN OUTPUT IS PENDING
1188  003206' 001402                          BEQ     RTNINT          ;BR IF NOT
1189  003210' 000167  000012                  JMP     OUTISR          ;GO HANDLE OUTPUT COMMAND
1190  003214' 152760  000021  000000  RTNINT: BISB    #IEI!IEO,BSEL0(R0) ;SET IEI AND IEO AGAIN
1191  003222' 000167  177226                  JMP     RTNMON          ;RETURN TO DEC/X11 TO AWAIT INTERRUPT
1192
1193
1194
1195
1196
1197                                  ;*************************************************************************
1198                                  ;* OUTISR - THIS IS THE OUTPUT INTERRUPT SERVICE ROUTINE. WHEN THE OUTPUT
1199                                  ;*       INTERRUPT OCCURS, THIS ROUTINE READS THE DEVICE CSR'S,
1200                                  ;*       CHECKS THE OUTPUT, AND MONITORS THE PROGRESS OF THE
1201                                  ;*       MODULE ON THE CURRENT ITERATION. IF SUFFICIENT ITERATIONS HAVE BEEN
1202                                  ;*       PERFORMED, AN END OF PASS IS INDICATED, AND A NEW PASS IS BEGUN.
1203                                  ;*************************************************************************
1204  003226'                         OUTISR:
1205  003226' 142760  000021  000000          BICB    #IEI!IEO,BSEL0(R0) ;CLEAR IEI, IEO
1206  003234' 032767  000010  175054          BIT     #NONQU,FLAGS    ;SEE IF NON-QUEUED OUTPUT INTRPT MODE
```



```
1207  003242' 001427                          BEQ     1$              ;BR IF NOT
1208  003244' 016067  000000  175032          MOV     SEL0(R0),RSEL0  ;READ AND STORE CSR'S
1209  003252' 016067  000002  175026          MOV     SEL2(R0),RSEL2
1210  003260' 016067  000004  175022          MOV     SEL4(R0),RSEL4
1211  003266' 016067  000006  175016          MOV     SEL6(R0),RSEL6
1212  003274' 005767  174524                  TST     SR4             ; IS THIS A DMV ?
1213  003300' 001403                          BEQ     26$             ; IF NO: SKIP SEL10 READ
1214  003302' 016067  000010  175004          MOV     SEL10(R0),RSEL10
1215  003310' 052767  000002  175000  26$:    BIS     #OUTINT,FLAGS   ;SET OUTPUT INTERRUPT FLAG
1216  003316' 000167  177672                  JMP     RTNINT          ;GO RETURN TO DEC/X11 MONITOR
1217  003322' 010004                  1$:     MOV     R0,R4           ;GET DEVICE ADDRESS
1218  003324' 166704  174456                  SUB     ADDR,R4         ;COMPUTE INTRPT STATUS TABLE INDEX
1219  003330' 005767  174470                  TST     SR4             ; IS THIS A DMV?
1220  003334' 001401                          BEQ     22$
1221  003336' 006204                          ASR     R4              ;IF YES: DIVIDE BY 4 FOR INDEX (DMV)
1222  003340' 006204                  22$:    ASR     R4              ;IF NO: DIVIDE BY 2 FOR INDEX (DMP)
1223  003342' 116001  000002                  MOVB    BSEL2(R0),R1    ;GET BSEL2 CONTENTS
1224  003346' 142701  000370                  BICB    #CMDMSK,R1      ;MASK OFF ALL BUT CMND BITS
1225  003352' 120127  000004                  CMPB    R1,#BACCOT      ;SEE IF BACCOT CMND JUST COMPLETED
1226  003356' 001437                          BEQ     4$              ;BR IF YES
1227  003360' 120127  000000                  CMPB    R1,#BACCOR      ;SEE IF BACCOR CMND JUST COMPLETED
1228  003364' 001534                          BEQ     8$              ;BR IF YES
1229  003366' 120127  000003                  CMPB    R1,#BACORU      ;SEE IF UNUSED RCV BUFFER RETURNED
1230  003372' 001002                          BNE     20$             ;BR IF NOT
1231  003374' 000167  000632                  JMP     21$             ;PROCEED
1232  003400' 120127  000001          20$:    CMPB    R1,#CTLOUT      ;SEE IF CONTROL OUT CMND JUST COMPLETED
1233  003404' 001002                          BNE     2$              ;BR IF NOT
1234  003406' 000167  000444                  JMP     12$             ;GO HANDLE CONTROL OUT COMMAND
1235  003412' 120127  000002          2$:     CMPB    R1,#INFOUT      ;SEE IF INFORMATION OUT CMND JUST COMPLETED
1236  003416' 001002                          BNE     3$              ;BR IF NOT
1237  003420' 000167  000514                  JMP     15$             ;GO HANDLE INFO OUT COMMAND
1238  003424' 004767  001540          3$:     JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
1239  003430' 012767  000011  174450          MOV     #ILGINT,ERRTYP  ;SET CODE FOR ILLEGAL INTRPT OCCURRED
1240                                          ;*************************************************************
1241  003436' 104405  000000' 000452'         HRDERS,BEGIN,REGTBL     ;UNSOLICITED OUTPUT INTRPT OCCURRED
1242                                          ;*************************************************************
1243  003444' 142760  000020  000000          BICB    #IEO,BSEL0(R0)  ;UNSOLICITED OUTPUT INTRPT - CLEAR IEO
1244  003452' 000167  176776                  JMP     RTNMON          ;RETURN TO DEC/X11 MONITOR TO AWAIT INTRPT
1245                                  ;HANDLE BACCUT
1246  003456' 026067  000004  174662  4$:     CMP     SEL4(R0),#BFPA  ;CHK FOR CORRECT TX BUFFER LO BITS RETURNED
1247  003464' 001411                          BEQ     5$              ;BR IF CORRECT
1248  003466' 004767  001476                  JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
1249  003472' 012767  000037  174406          MOV     #BADWRT,ERRTYP  ;CODE FOR UNABLE TO EXEC WRITE FUNCTION
1250                                          ;*************************************************************
1251  003500' 104405  000000' 000452'         HRDER$,BEGIN,REGTBL     ;INCORRECT TX BA BITS RETURNED
1252                                          ;*************************************************************
1253  003506' 000453                          BR      7$
1254  003510' 116001  000007          5$:     MOVB    BSEL7(R0),R1    ;GET BSEL7 CONTENTS
1255  003514' 142701  000077                  BICB    #EAMSK,R1       ;MASK FOR BUS ADRS EA BITS
1256  003520' 032767  000002  174276          BIT     #MODE22,SR4     ;* IF THIS IS A DMV IN Q22 MODE...
1257  003526' 001402                          BEQ     30$             ;* THEN GET EXTENDED ADDRESS (EA)
1258  003530' 116001  000006                  MOVB    BSEL6(R0),R1    ;* BITS FROM A DIFFERENT CSR.
1259  003534' 120167  174610          30$:    CMPB    R1,WBFEA        ;CHK FOR CORRECT EA BITS RETURNED
1260  003540' 001411                          BEQ     6$              ;BR IF CORRECT
1261  003542' 004767  001422                  JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
1262  003546' 012767  000037  174332          MOV     #BADWRT,ERRTYP  ;CODE FOR UNABLE TO EXEC WRITE FUNCTION
```

```
1263
1264  003554' 104405  000000' 000452'          ;*****************************************************************
                                               HRDER$,BEGIN,REGTBL     ;INCORRECT TX EA BITS RETURNED
1265                                           ;*****************************************************************
1266  003562' 000425                           BR      7$
1267  003564' 016001  000006          6$:     MOV     SEL6(R0),R1     ;GET SEL6 CONTENTS
1268  003570' 032767  000002  174226           BIT     #MODE22,SR4     ;* IF THIS IS A DMV IN Q22 MODE...
1269  003576' 001402                           BEQ     28$             ;* THEN GET CHARACTER COUNT
1270  003600' 016001  000010                   MOV     SEL10(R0),R1    ;* BITS FROM A DIFFERENT CSR.
1271  003604' 042701  140000          28$:    BIC     #CCMSK,R1       ;MASK FOR CHAR COUNT BITS
1272  003610' 020127  002000                   CMP     R1,#1024.       ;CHK FOR CORRECT CHAR CNT RETURNED
1273  003614' 001410                           BEQ     7$              ;BR IF CORRECT
1274  003616' 004767  001346                   JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
1275  003622' 012767  000037  174256           MOV     #BADWRT,ERRTYP  ;CODE FOR UNABLE TO EXEC WRITE FUNCTION
1276                                           ;*****************************************************************
1277  003630' 104405  000000' 000452'          HRDER$,BEGIN,REGTBL     ;INCORRECT TX CHAR CNT BITS RETURNED
1278                                           ;*****************************************************************
1279  003636' 012764  000005  000352' 7$:      MOV     #MDFDON!ISTDON,INTABL(R4) ;RESET STATUS FOR NEXT ITERATION
1280  003644' 152760  000200  000000           BISB    #RQI,BSEL0(R0)  ;ENABLE INPUTS AGAIN
1281  003652' 000167  000354                   JMP     21$
1282                                   ;HANDLE BACCOR
1283  003656' 026067  000004  174242  8$:      CMP     SEL4(R0),RBUFPA ;CHK FOR CORRECT RCV BUFFER LO BITS RETURNED
1284  003664' 001411                           BEQ     9$              ;BR IF CORRECT
1285  003666' 004767  001276                   JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
1286  003672' 012767  000036  174206           MOV     #BADRED,ERRTYP  ;CODE FOR UNABLE TO EXEC READ FUNCTION
1287                                           ;*****************************************************************
1288  003700' 104405  000000' 000452'          HRDER$,BEGIN,REGTBL     ;INCORRECT RCV BA BITS RETURNED
1289                                           ;*****************************************************************
1290  003706' 000454                           BR      11$
1291  003710' 116001  000007          9$:     MOVB    BSEL7(R0),R1    ;GET BSEL7 CONTENTS
1292  003714' 142701  000077                   BICB    #EAMSK,R1       ;MASK FOR BUS ADRS EA BITS
1293  003720' 032767  000002  174076           BIT     #MODE22,SR4     ;* IF THIS IS A DMV IN Q22 MODE...
1294  003726' 001402                           BEQ     31$             ;* THEN GET EXTENDED ADDRESS (EA)
1295  003730' 116001  000006                   MOVB    BSEL6(R0),R1    ;* BITS FROM A DIFFERENT CSR.
1296  003734' 120167  174312          31$:    CMPB    R1,BUFFEX       ;CHK FOR CORRECT EA BITS RETURNED
1297  003740' 001411                           BEQ     10$             ;BR IF CORRECT
1298  003742' 004767  001222                   JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
1299  003746' 012767  000036  174132           MOV     #BADRED,ERRTYP  ;CODE FOR UNABLE TO EXEC READ FUNCTION
1300                                           ;*****************************************************************
1301  003754' 104405  000000' 000452'          HRDER$,BEGIN,REGTBL     ;INCORRECT RCV EA BITS RETURNED
1302                                           ;*****************************************************************
1303  003762' 000426                           BR      11$
1304  003764' 016001  000006          10$:    MOV     SEL6(R0),R1     ;GET SEL6 CONTENTS
1305  003770' 032767  000002  174026           BIT     #MODE22,SR4     ;* IF THIS IS A DMV IN Q22 MODE...
1306  003776' 001402                           BEQ     29$             ;* THEN GET CHARACTER COUNT
1307  004000' 016001  000010                   MOV     SEL10(R0),R1    ;* BITS FROM A DIFFERENT CSR.
1308  004004' 042701  140000          29$:    BIC     #CCMSK,R1       ;MASK FOR CHAR COUNT BITS
1309  004010' 020127  002000                   CMP     R1,#1024.       ;CHK FOR CORRECT CHAR CNT RETURNED
1310  004014' 001411                           BEQ     11$             ;BR IF CORRECT
1311  004016' 004767  001146                   JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
1312  004022' 012767  000036  174056           MOV     #BADRED,ERRTYP  ;CODE FOR UNABLE TO EXEC READ FUNCTION
1313                                           ;*****************************************************************
1314  004030' 104405  000000' 000452'          HRDER$,BEGIN,REGTBL     ;INCORRECT RCV CHAR CNT BITS RETURNED
1315                                           ;*****************************************************************
1316  004036' 000400                           BR      11$
1317  004040' 052764  000040  000352' 11$:    BIS     #BURDON,INTABL(R4) ;SET BACCOR DONE IN STATUS WORD
1318  004046' 152760  000200  000000           BISB    #RQI,BSEL0(R0)  ;ENABLE INPUTS AGAIN
```



```
1319  004054' 000466                           BR      21$             ;PROCEED
1320                                   ;HANDLE CTL OUT
1321  004056' 126027  000006  000010  12$:     CMPB    BSEL6(R0),#STARUN ;SEE IF START RECEIVED FROM MASTER
1322                                                                   ;    WHILE RUNNING
1323  004064' 001004                           BNE     13$             ;BR IF NOT
1324  004066' 142760  000200  000000           BICB    #RQI,BSEL0(R0)  ;DISABLE INPUTS TO AWAIT OUTPUT
1325  004074' 000456                           BR      21$             ;PROCEED
1326  004076' 126027  000006  000024  13$:     CMPB    BSEL6(R0),#RUNST ;SEE IF PROTOCOL JUST ENTERED THE RUNNING
1327                                                                   ;   STATE
1328  004104' 001004                           BNE     14$             ;BR IF NOT
1329  004106' 152760  000200  000000           BISB    #RQI,BSEL0(R0)  ;ENABLE INPUTS AGAIN
1330  004114' 000446                           BR      21$             ;PROCEED
1331  004116' 004767  001046  14$:     JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
1332  004122' 012767  000000  173756           MOV     #NOTDFN,ERRTYP  ;CODE FOR ERROR NOT DEFINED (MUST INTERPRET
1333                                                                   ;   CSR CONTENTS)
1334                                           ;*****************************************************************
1335  004130' 104405  000000' 000452'          HRDER$,BEGIN,REGTBL     ;UNEXPECTED CTL OUT CMND
1336                                           ;*****************************************************************
1337  004136' 000435                           BR      21$
1338                                   ;HANDLE INFO OUT
1339  004140' 116001  000006  15$:     MOVB    BSEL6(R0),R1    ;GET BSEL6 CODE RETURNED
1340  004144' 142701  000340                   BICB    #RTNMSK,R1      ;MASK FOR RETURN CODE
1341  004150' 120127  000020                   CMPB    R1,#BUFRTC      ;CHECK FOR BUFFER RETURN COMPLETE
1342  004154' 001016                           BNE     19$             ;BR IF NOT
1343  004156' 005267  174162                   INC     OUTDUN          ;INCR CNT OF LOG LNKS RESTARTED BY MASTER
1344  004162' 026767  174156  174136           CMP     OUTDUN,LLKCNT   ;SEE IF ALL LOG LNKS DONE YET
1345  004170' 002420                           BLT     21$             ;BR IF ALL NOT DONE
1346  004172' 105060  000002                   CLRB    BSEL2(R0)       ;CLEAR BSEL2 (INCLUDING RDYO)
1347  004176' 004767  000724                   JSR     PC,DISABL       ;DISABLE INTRPTS ON ALL DEVICES
1348  004202' 104413  000000'                  ENDIT$,BEGIN            ;SIGNAL END OF ITERATION,
                                                                       ;MONITOR SHALL TEST END OF PASS
1349
1350  004206' 000167  176012                   JMP     RESTRT          ;GO START NEW PASS
1351  004212' 004767  000752  19$:     JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
1352  004216' 012767  000000  173662           MOV     #NOTDFN,ERRTYP  ;CODE FOR ERROR NOT DEFINED (MUST INTERPRET
1353                                                                   ;   CSR CONTENTS)
1354                                           ;*****************************************************************
1355  004224' 104405  000000' 000452'          HRDER$,BEGIN,REGTBL     ;UNEXPECTED INFO OUT CMND
1356                                           ;*****************************************************************
1357  004232' 105060  000002           21$:    CLRB    BSEL2(R0)       ;CLEAR BSEL2 (INCLUDING RDYO)
1358  004236' 000167  176722                   JMP     CHKMOR          ;GO SEE IF RDYI OR RDYO STILL SET,
                                                                       ;   AND RETURN TO DEC/X11 MONITOR IF NOT
1359
1360
1361
1362                                   ;*****************************************************************************
1363                                   ;* DMVLUP - THIS SUBROUTINE ENTERS M-LOOP, SETS TTL LOOPBACK AND TIMER 1 FOR
1364                                   ;*      56K BPS, AND EXITS MLOOP. (DMV ONLY)
1365                                   ;*****************************************************************************
1366  004242' 112760  000301  000001  DMVLUP: MOVB    #301,BSEL1(R0)  ;ENTER MAINTENANCE LOOP
1367  004250'                          1$:
1368  004250' 104407  000000'                  BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
1369  004254' 104407  000000'                  BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
1370  004260' 132760  000200  000002           BITB    #RDY,BSEL2(R0)  ;MAKE SURE MLOOP COMPLETE
1371  004266' 001770                           BEQ     1$
1372  004270' 012760  000006  000002           MOV     #TTLOOP,BSEL2(R0) ;ISSUE TTLOOP CONFIGURE COMMAND
1373  004276' 104407  000000'                  BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
1374  004302' 104407  000000'                  BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
```

```
1375  004306' 000240                    NOP
1376  004310' 000240                    NOP
1377  004312' 000207                    RTS     PC              ;RETURN
1378
1379
1380                                    ;*************************************************************************
1381                                    ;* DMVRD - THIS SUBROUTINE FORCES THE DMV WHOSE ADRS IS PASSED IN R0 ON ENTRY
1382                                    ;*      TO ENTER THE M-LOOP AND READ THE INTERNAL DMV LOCATION PASSED IN THE
1383                                    ;*      WORD FOLLOWING THE CALL.
1384                                    ;*************************************************************************
1385  004314' 112760  000301  000001   DMVRD:  MOVB    #301,BSEL1(R0)  ;ENTER MAINTENANCE LOOP
1386  004322'
1387  004322' 104407  000000'           1$:
1388  004326' 104407  000000'                   BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
1389  004332' 132760  000200  000002           BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
1390  004340' 001770                    BITB    #MRDY,BSEL2(R0) ;MAKE SURE MLOOP COMPLETE
1391  004342' 017660  000000  000004           BEQ     1$
1392  004350' 012760  000001  000002   MOV     @(SP),SEL4(R0)  ;PASS DMV INTERNAL ADDRESS
1393  004356'                           MOV     #REDLOC,BSEL2(R0) ;ISSUE READ COMMAND
1394  004356' 104407  000000'           2$:
1395  004362' 104407  000000'                   BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
1396  004366' 132760  000200  000002           BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
1397  004374' 001770                    BITB    #MRDY,BSEL2(R0) ;MAKE SURE MLOOP READ IS COMPLETE
1398  004376' 062716  000002                   BEQ     2$
1399  004402' 000207                    ADD     #2,(SP)         ;FIX UP RETURN PC
1400                                     RTS     PC              ;RETURN
1401
1402
1403                                    ;*************************************************************************
1404                                    ;* EXECUT - THIS SUBROUTINE FORCES THE MICROPROCESSOR WHOSE ADRS IS PASSED
1405                                    ;*      IN R0 ON ENTRY TO EXECUTE AN INSTRUCTION WHICH IS PASSED IN THE WORD
1406                                    ;*      FOLLOWING THE CALL.
1407  004404' 152760  000006  000001   ;*************************************************************************
1408  004412' 017660  000000  000006   EXECUT: BISB    #ROMO!ROMI,BSEL1(R0)    ;SET ROMO, ROMI BITS IN BSEL1
1409  004420' 152760  000007  000001           MOV     @(SP),SEL6(R0)  ;PUT INSTRUCTION INTO SEL6
1410  004426' 142760  000007  000001           BISB    #ROMO!ROMI!STEPMP,BSEL1(R0) ;SET ROMO, ROMI, STEPMP IN BSEL1
1411  004434' 062716  000002                   BICB    #ROMO!ROMI!STEPMP,BSEL1(R0) ;CLEAR ROMO, ROMI, STEPMP IN BSEL1
1412  004440' 000207                    ADD     #2,(SP)         ;FIX UP RETURN PC
1413                                     RTS     PC              ;RETURN
1414
1415
1416                                    ;*************************************************************************
1417                                    ;* INIDMP - THIS SUBROUTINE ISSUES A MASTER CLEAR AND CLEARS THE CSR'S,
1418                                    ;*      FOR THE DEVICE WHOSE ADDRESS IS PASSED IN R0 ON ENTRY.
1419                                    ;*************************************************************************
1420  004442' 010146                    INIDMP: MOV     R1,-(SP)        ;SAVE R1
1421  004444' 004767  000236                   JSR     PC,CLRCSR       ;CLEAR BSEL0, SEL2,4,6
1422  004450' 112760  000100  000001           MOVB    #MCLR,BSEL1(R0) ;SET MASTER CLEAR BIT
1423  004456' 005767  173342                   TST     SR4             ; IF DMV...
1424  004462' 001003                    BNE     6$              ; THEN  DON'T SET RUN
1425                                    ;-------------------------------------------------------------------------
1426                                    ;;;     MOVB    #RUN,BSEL1(R0)  ;PATCH TO SET RUN BIT IF USING M8206
1427  004464' 000240                    NOP                     ; 112760
1428  004466' 000240                    NOP                     ;    200
1429  004470' 000240                    NOP                     ;      1
1430                                    ;-------------------------------------------------------------------------
```



```
1431  004472' 042767  000004  173620   6$:     BIC     #BADINI,ERRORS  ;CLEAR BAD INIT ERROR FLAG
1432  004500' 012701  020000                   MOV     #20000,R1       ;INITIALIZE INIT TIMER
1433  004504' 132760  000200  000001   1$:     BITB    #RUN,BSEL1(R0)  ;SEE IF RUN IS SET AGAIN
1434  004512' 001020                    BNE     3$              ;BR IF NOT SET
1435  004514' 005301                    DEC     R1              ;DECR TIMER
1436  004516' 001405                    BEQ     2$              ;BR IF INIT TIMED-OUT
1437  004520' 104407  000000'                   BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
1438  004524' 104407  000000'                   BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
1439  004530' 000765                    BR      1$              ;GO CHECK AGAIN
1440  004532' 004767  000432   2$:     JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
1441  004536' 012767  000034  173342           MOV     #NOINIT,ERRTYP  ;CODE FOR DEVICE WILL NOT INIT
1442                                    ;*****************************************************************
1443  004544' 104405  000000' 000452'          HRDERS,BEGIN,REGTBL     ;TIMED-OUT WAITING FOR RUN TO SET
1444                                    ;*****************************************************************
1445  004552' 000414                    BR      4$
1446  004554' 126027  000006  000305   3$:     CMPB    BSEL6(R0),#GOODIN ;SEE IF INIT COMPLETE CODE SET BY MICRO-CPU
1447  004562' 001413                    BEQ     5$              ;BR IF YES
1448  004564' 004767  000400                   JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
1449  004570' 012767  000034  173310           MOV     #NOINIT,ERRTYP  ;CODE FOR DEVICE WILL NOT INIT
1450                                    ;*****************************************************************
1451  004576' 104405  000000' 000452'          HRDERS,BEGIN,REGTBL     ;INIT COMPLETE CODE NOT SET BY MICRO-CPU
1452                                    ;*****************************************************************
1453  004604' 052767  000004  173506   4$:     BIS     #BADINI,ERRORS  ;SET BAD INIT ERROR FLAG
1454  004612' 004767  000070           5$:     JSR     PC,CLRCSR       ;CLEAR BSEL0, SEL2,4,6
1455  004616' 012601                    MOV     (SP)+,R1        ;RESTORE R1
1456  004620' 000207                    RTS     PC              ;RETURN
1457
1458
1459
1460                                    ;*************************************************************************
1461                                    ;* READ16 - THIS SUBROUTINE FORCES THE MICROPROCESSOR WHOSE ADDRESS IS
1462                                    ;*      PASSED IN R0 ON ENTRY, TO EXECUTE AN INSTRUCTION WHICH READS DMP LINE
1463                                    ;*      UNIT IBUS REG 16 INTO BSEL4, AND THEN IT PLACES THE CONTENTS INTO LOC.
1464                                    ;*      LURG16 (IF DMV: EXECUTE MLOOP, "SWPBOT" => LURG16, AND ADJUST).
1465                                    ;*************************************************************************
1466  004622' 005767  173176   READ16: TST     SR4             ;IS THIS A DMV ?
1467  004626' 001007                    BNE     1$              ;BRANCH IF YES
1468
1469  004630' 004767  177550                   JSR     PC,EXECUT       ;DMP: EXECUTE MOVE INSTRUCTION
1470  004634' 021344                    .WORD   RDLU16
1471  004636' 116067  000004  173424           MOVB    BSEL4(R0),LURG16 ;GET REG 16 CONTENTS INTO LURG16
1472  004644' 000415                    BR      2$
1473
1474  004646' 004767  177442   1$:     JSR     PC,DMVRD        ;DMV: READ "SWPBOT" SWITCHES
1475  004652' 121000                    .WORD   SWPBOT
1476  004654' 116067  000006  173406           MOVB    BSEL6(R0),LURG16 ;PUT IT IN LURG16
1477  004662' 000241                    CLC                     ;ADJUST "SWPBOT" BITS
1478  004664' 106267  173400                   ASRB    LURG16          ; TO LOOK LIKE THE DMP
1479  004670' 103003                    BCC     2$              ;  REG 16 SWITCHES
1480  004672' 152767  000200  173370           BISB    #ENABSW,LURG16
1481
1482  004700' 105067  173365   2$:     CLRB    LURG16+1        ;CLEAR HI BYTE
1483  004704' 000207                    RTS     PC              ;RETURN
1484
1485
1486
```

```
                                        ;*******************************************************************************
                                        ;* CLRCSR - THIS SUBROUTINE CLEARS BSEL0, SEL2,4,6 (AND SEL10 IF DMV11), FOR
                                        ;*       THE DEVICE WHOSE ADRS IS PASSED IN R0 ON ENTRY
                                        ;*******************************************************************************
004706' 105060  000000          CLRCSR: CLRB    BSEL0(R0)       ;CLEAR BSEL0
004712' 005060  000002                  CLR     SEL2(R0)        ;CLEAR SEL2
004716' 005060  000004                  CLR     SEL4(R0)        ;CLEAR SEL4
004722' 005060  000006                  CLR     SEL6(R0)        ;CLEAR SEL6
004726' 005767  173072                  TST     SR4
004732' 001402                          BEQ     1$
004734' 005060  000010                  CLR     SEL10(R0)       ;IF DMV: CLEAR SEL10
004740' 000207                  1$:     RTS     PC              ;RETURN


                                        ;*******************************************************************************
                                        ;* INCMND - THIS SUBROUTINE ISSUES AN INPUT COMMAND TO THE
                                        ;*       DEVICE WHOSE ADDRESS IS PASSED IN R0, BY ENABLING INTERRUPTS
                                        ;*       AND SETTING RQI. THEN, IT SETS NONQI IN FLAGS, FOR NON-QUEUED
                                        ;*       (REQUEST-AND-WAIT) INPUT INTERRUPT OPERATION.
                                        ;*       IF THE PROGRAM TIMES-OUT WAITING FOR COMMAND COMPLETION,
                                        ;*       THE ERROR IS REPORTED, AND A RETURN IS MADE WITH THE INPUT INTERRUPT
                                        ;*       TIME-OUT FLAG IN "ERRORS" SET TO 1.
                                        ;*******************************************************************************
004742' 042767  000001  173350  INCMND: BIC     #INTIMO,ERRORS  ;CLR INPUT INTRPT TIME-OUT FLAG
004750' 052767  000004  173340          BIS     #NONQI,FLAGS    ;SET BIT FOR NON-QUEUED INPUT INTRPT OPERATION
004756' 042767  000001  173332          BIC     #ININT,FLAGS    ;CLR INPUT INTRPT FLAG
004764' 005067  173314                  CLR     RSEL0           ;CLEAR OUTPUT CMND RETURN AREA
004770' 005067  173312                  CLR     RSEL2
004774' 005067  173310                  CLR     RSEL4
005000' 005067  173306                  CLR     RSEL6
005004' 012767  030000  173310          MOV     #30000,INTIMR   ;INIT INPUT INTRPT TIMER
005012' 112760  000221  000000          MOVB    #RQI!IEI!IEO,BSEL0(R0) ;REQUEST INPUT, ENABLE INTERRUPTS ON THIS DEVICE
005020'                         1$:
005020' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
005024' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
005030' 132767  000001  173260          BITB    #ININT,FLAGS    ;SEE IF INPUT INTRPT SERVICED YET
005036' 001016                          BNE     3$              ;BR IF YES
005040' 005367  173256                  DEC     INTIMR          ;DECREMENT TIMER
005044' 001365                          BNE     1$              ;BR IF INPUT INTERRUPT DIDN'T TIME-OUT YET
005046' 004767  000116          2$:     JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
005052' 012767  000023  173026          MOV     #NOINTR,ERRTYP  ;CODE FOR DEV FAILED TO INTRPT
                                        ;*****************************************************************
005060' 104405  000000' 000452'         HRDERR$,BEGIN,REGTBL    ;INPUT INTERRUPT TIMED-OUT
                                        ;*****************************************************************
005066' 052767  000001  173224          BIS     #INTIMO,ERRORS  ;SET INPUT INTERRUPT TIME-OUT FLAG
005074' 042767  000004  173214  3$:     BIC     #NONQI,FLAGS    ;CLEAR NON-QUEUED BIT
005102' 000207                          RTS     PC      ;RETURN


                                        ;*******************************************************************************
                                        ;* CLRCMD - THIS SUBROUTINE CLEARS THE COMMAND AREA : CSEL2,CSEL4,CSEL6,CSEL10
                                        ;*******************************************************************************
```



```
005104' 005067  173164          CLRCMD: CLR     CSEL2
005110' 005067  173162                  CLR     CSEL4
005114' 005067  173160                  CLR     CSEL6
005120' 005067  173156                  CLR     CSEL10
005124' 000207                          RTS     PC              ;RETURN


                                        ;*******************************************************************************
                                        ;* DISABL - DISABLE INTERRUPTS ON ALL DEVICES
                                        ;*******************************************************************************
005126' 016700  172654          DISABL: MOV     ADDR,R0         ;INIT DEVICE ADDRESS
005132' 012767  000001  173126          MOV     #B1IO,DEVPTR    ;INIT SELECTION POINTER
005140' 036767  173122  173116  1$:     BIT     DEVPTR,SELECT   ;SEE IF THIS DEVICE IS SELECTED
005146' 001402                          BEQ     2$              ;BR IF NOT SELECTED
005150' 105060  000000                  CLRB    BSEL0(R0)       ;CLEAR RQI, IEI, IEO
005154' 062700  000010          2$:     ADD     #10,R0          ;INCR DEVICE ADDRESS
005160' 006367  173102                  ASL     DEVPTR          ;SHIFT SELECTION POINTER
005164' 001365                          BNE     1$              ;BR IF MORE TO SCAN
005166' 000207                          RTS     PC              ;RETURN


                                        ;*******************************************************************************
                                        ;* GETERR - THIS SUBROUTINE LOADS THE CSR ADRS, SEL0 AND SEL2 CONTENTS INTO
                                        ;*       CSRA, ACSR, AND ASTAT FOR ERROR PRINTOUT. THE DEVICE ADRS IS ASSUMED
                                        ;*       TO BE IN R0 ON ENTRY. THEN, IT LOADS THE CSR ADDRESSES INTO REGTBL.
                                        ;*******************************************************************************
005170' 010067  172704          GETERR: MOV     R0,CSRA         ;LOAD CSR ADDRESS
005174' 016067  000000  172700          MOV     SEL0(R0),ACSR   ;LOAD SEL0 CONTENTS
005202' 016067  000002  172674          MOV     SEL2(R0),ASTAT  ;LOAD SEL2 CONTENTS
005210' 010067  173236                  MOV     R0,REGTBL       ;GET CSR ADDRESSES FOR DEVICE INTO REGTBL
005214' 010067  173234                  MOV     R0,REGTBL+2
005220' 062767  000002  173226          ADD     #2,REGTBL+2
005226' 010067  173224                  MOV     R0,REGTBL+4
005232' 062767  000004  173216          ADD     #4,REGTBL+4
005240' 010067  173214                  MOV     R0,REGTBL+6
005244' 062767  000006  173206          ADD     #6,REGTBL+6
005252' 005767  172546                  TST     SR4             ; IS THIS A DMV ?
005256' 001004                          BNE     1$
005260' 012767  177777  173174          MOV     #177777,REGTBL+10       ;NO: DISABLE SEL10
005266' 000405                          BR      2$              ;       AND GOTO RETURN
005270' 010067  173166          1$:     MOV     R0,REGTBL+10    ;YES: ENABLE SEL10
005274' 062767  000010  173160          ADD     #10,REGTBL+10
005302' 000207                  2$:     RTS     PC              ;RETURN


                                        ;*******************************************************************************
```

```
                                        ;* SERVICE CODE FOR LINKING A PARTICULAR DEVICE TO
                                        ;* A COMMON INPUT OR OUTPUT INTERRUPT SERVICE ROUTINE.
                                        ;*****************************************************************************


005304'                         IISR0:
                                        ;-----------------------------------------------------------------
005304' 000004  000000' 005312'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-----------------------------------------------------------------
005312' 016700  000022          1$:     MOV     ADDR0,R0        ;PUT DEVICE 0 CSR ADDRESS IN R0
005316' 000167  175136                  JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 0

005322'                         OISR0:
                                        ;-----------------------------------------------------------------
005322' 000004  000000' 005330'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-----------------------------------------------------------------
005330' 016700  000004          1$:     MOV     ADDR0,R0        ;PUT DEVICE 0 CSR ADDRESS IN R0
005334' 000167  175666                  JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 0

005340' 000000                  ADDR0:  .WORD   0               ;DEVICE CSR ADDRESS STORED HERE


005342'                         IISR1:
                                        ;-----------------------------------------------------------------
005342' 000004  000000' 005350'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-----------------------------------------------------------------
005350' 016700  000022          1$:     MOV     ADDR1,R0        ;PUT DEVICE 1 CSR ADDRESS IN R0
005354' 000167  175100                  JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 1

005360'                         OISR1:
                                        ;-----------------------------------------------------------------
005360' 000004  000000' 005366'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-----------------------------------------------------------------
005366' 016700  000004          1$:     MOV     ADDR1,R0        ;PUT DEVICE 1 CSR ADDRESS IN R0
005372' 000167  175630                  JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 1

005376' 000000                  ADDR1:  .WORD   0               ;DEVICE CSR ADDRESS STORED HERE


005400'                         IISR2:
                                        ;-----------------------------------------------------------------
005400' 000004  000000' 005406'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-----------------------------------------------------------------
005406' 016700  000022          1$:     MOV     ADDR2,R0        ;PUT DEVICE 2 CSR ADDRESS IN R0
005412' 000167  175042                  JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 2

005416'                         OISR2:
                                        ;-----------------------------------------------------------------
005416' 000004  000000' 005424'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-----------------------------------------------------------------
005424' 016700  000004          1$:     MOV     ADDR2,R0        ;PUT DEVICE 2 CSR ADDRESS IN R0
005430' 000167  175572                  JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 2

005434' 000000                  ADDR2:  .WORD   0               ;DEVICE CSR ADDRESS STORED HERE
```



```


005436'                         IISR3:
                                        ;-----------------------------------------------------------------
005436' 000004  000000' 005444'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-----------------------------------------------------------------
005444' 016700  000022          1$:     MOV     ADDR3,R0        ;PUT DEVICE 3 CSR ADDRESS IN R0
005450' 000167  175004                  JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 3

005454'                         OISR3:
                                        ;-----------------------------------------------------------------
005454' 000004  000000' 005462'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-----------------------------------------------------------------
005462' 016700  000004          1$:     MOV     ADDR3,R0        ;PUT DEVICE 3 CSR ADDRESS IN R0
005466' 000167  175534                  JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 3

005472' 000000                  ADDR3:  .WORD   0               ;DEVICE CSR ADDRESS STORED HERE


005474'                         IISR4:
                                        ;-----------------------------------------------------------------
005474' 000004  000000' 005502'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-----------------------------------------------------------------
005502' 016700  000022          1$:     MOV     ADDR4,R0        ;PUT DEVICE 4 CSR ADDRESS IN R0
005506' 000167  174746                  JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 4

005512'                         OISR4:
                                        ;-----------------------------------------------------------------
005512' 000004  000000' 005520'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-----------------------------------------------------------------
005520' 016700  000004          1$:     MOV     ADDR4,R0        ;PUT DEVICE 4 CSR ADDRESS IN R0
005524' 000167  175476                  JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 4

005530' 000000                  ADDR4:  .WORD   0               ;DEVICE CSR ADDRESS STORED HERE


005532'                         IISR5:
                                        ;-----------------------------------------------------------------
005532' 000004  000000' 005540'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-----------------------------------------------------------------
005540' 016700  000022          1$:     MOV     ADDR5,R0        ;PUT DEVICE 5 CSR ADDRESS IN R0
005544' 000167  174710                  JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 5

005550'                         OISR5:
                                        ;-----------------------------------------------------------------
005550' 000004  000000' 005556'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-----------------------------------------------------------------
005556' 016700  000004          1$:     MOV     ADDR5,R0        ;PUT DEVICE 5 CSR ADDRESS IN R0
005562' 000167  175440                  JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 5

005566' 000000                  ADDR5:  .WORD   0               ;DEVICE CSR ADDRESS STORED HERE


```

```
 1711
 1712  005570'                          IISR6:
 1713
 1714  005570' 000004  000000' 005576'          PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
 1715                                           ;-------------------------------------------------------------
 1716  005576' 016700  000022           1$:     MOV     ADDR6,R0        ;PUT DEVICE 6 CSR ADDRESS IN R0
 1717  005602' 000167  174652                   JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 6
 1718
 1719  005606'                          OISR6:
 1720                                           ;-------------------------------------------------------------
 1721  005606' 000004  000000' 005614'          PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
 1722                                           ;-------------------------------------------------------------
 1723  005614' 016700  000004           1$:     MOV     ADDR6,R0        ;PUT DEVICE 6 CSR ADDRESS IN R0
 1724  005620' 000167  175402                   JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 6
 1725
 1726  005624' 000000                   ADDR6:  .WORD   0               ;DEVICE CSR ADDRESS STORED HERE
 1727
 1728
 1729
 1730  005626'                          IISR7:
 1731                                           ;-------------------------------------------------------------
 1732  005626' 000004  000000' 005634'          PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
 1733                                           ;-------------------------------------------------------------
 1734  005634' 016700  000022           1$:     MOV     ADDR7,R0        ;PUT DEVICE 7 CSR ADDRESS IN R0
 1735  005640' 000167  174614                   JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 7
 1736
 1737  005644'                          OISR7:
 1738                                           ;-------------------------------------------------------------
 1739  005644' 000004  000000' 005652'          PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
 1740                                           ;-------------------------------------------------------------
 1741  005652' 016700  000004           1$:     MOV     ADDR7,R0        ;PUT DEVICE 7 CSR ADDRESS IN R0
 1742  005656' 000167  175344                   JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 7
 1743
 1744  005662' 000000                   ADDR7:  .WORD   0               ;DEVICE CSR ADDRESS STORED HERE
 1745
 1746
 1747
 1748  005664'                          IISR10:
 1749                                           ;-------------------------------------------------------------
 1750  005664' 000004  000000' 005672'          PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
 1751                                           ;-------------------------------------------------------------
 1752  005672' 016700  000022           1$:     MOV     ADDR10,R0       ;PUT DEVICE 10 CSR ADDRESS IN R0
 1753  005676' 000167  174556                   JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 10
 1754
 1755  005702'                          OISR10:
 1756                                           ;-------------------------------------------------------------
 1757  005702' 000004  000000' 005710'          PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
 1758                                           ;-------------------------------------------------------------
 1759  005710' 016700  000004           1$:     MOV     ADDR10,R0       ;PUT DEVICE 10 CSR ADDRESS IN R0
 1760  005714' 000167  175306                   JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 10
 1761
 1762  005720' 000000                   ADDR10: .WORD   0               ;DEVICE CSR ADDRESS STORED HERE
 1763
 1764
 1765
 1766  005722'                          IISR11:
```



```
 1767
 1768  005722' 000004  000000' 005730'          PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
 1769                                           ;-------------------------------------------------------------
 1770  005730' 016700  000022           1$:     MOV     ADDR11,R0       ;PUT DEVICE 11 CSR ADDRESS IN R0
 1771  005734' 000167  174520                   JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 11
 1772
 1773  005740'                          OISR11:
 1774                                           ;-------------------------------------------------------------
 1775  005740' 000004  000000' 005746'          PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
 1776                                           ;-------------------------------------------------------------
 1777  005746' 016700  000004           1$:     MOV     ADDR11,R0       ;PUT DEVICE 11 CSR ADDRESS IN R0
 1778  005752' 000167  175250                   JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 11
 1779
 1780  005756' 000000                   ADDR11: .WORD   0               ;DEVICE CSR ADDRESS STORED HERE
 1781
 1782
 1783
 1784  005760'                          IISR12:
 1785                                           ;-------------------------------------------------------------
 1786  005760' 000004  000000' 005766'          PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
 1787                                           ;-------------------------------------------------------------
 1788  005766' 016700  000022           1$:     MOV     ADDR12,R0       ;PUT DEVICE 12 CSR ADDRESS IN R0
 1789  005772' 000167  174462                   JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 12
 1790
 1791  005776'                          OISR12:
 1792                                           ;-------------------------------------------------------------
 1793  005776' 000004  000000' 006004'          PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
 1794                                           ;-------------------------------------------------------------
 1795  006004' 016700  000004           1$:     MOV     ADDR12,R0       ;PUT DEVICE 12 CSR ADDRESS IN R0
 1796  006010' 000167  175212                   JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 12
 1797
 1798  006014' 000000                   ADDR12: .WORD   0               ;DEVICE CSR ADDRESS STORED HERE
 1799
 1800
 1801
 1802  006016'                          IISR13:
 1803                                           ;-------------------------------------------------------------
 1804  006016' 000004  000000' 006024'          PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
 1805                                           ;-------------------------------------------------------------
 1806  006024' 016700  000022           1$:     MOV     ADDR13,R0       ;PUT DEVICE 13 CSR ADDRESS IN R0
 1807  006030' 000167  174424                   JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 13
 1808
 1809  006034'                          OISR13:
 1810                                           ;-------------------------------------------------------------
 1811  006034' 000004  000000' 006042'          PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
 1812                                           ;-------------------------------------------------------------
 1813  006042' 016700  000004           1$:     MOV     ADDR13,R0       ;PUT DEVICE 13 CSR ADDRESS IN R0
 1814  006046' 000167  175154                   JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 13
 1815
 1816  006052' 000000                   ADDR13: .WORD   0               ;DEVICE CSR ADDRESS STORED HERE
 1817
 1818
 1819
 1820  006054'                          IISR14:
 1821                                           ;-------------------------------------------------------------
 1822  006054' 000004  000000' 006062'          PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
```

```
                                        ;-------------------------------------------------------------
006062' 016700  000022          1$:     MOV     ADDR14,R0       ;PUT DEVICE 14 CSR ADDRESS IN R0
006066' 000167  174366                  JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 14

006072'                         OISR14:
                                        ;-------------------------------------------------------------
006072' 000004  000000' 006100'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-------------------------------------------------------------
006100' 016700  000004          1$:     MOV     ADDR14,R0       ;PUT DEVICE 14 CSR ADDRESS IN R0
006104' 000167  175116                  JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 14

006110' 000000                  ADDR14: .WORD   0               ;DEVICE CSR ADDRESS STORED HERE


006112'                         IISR15:
                                        ;-------------------------------------------------------------
006112' 000004  000000' 006120'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-------------------------------------------------------------
006120' 016700  000022          1$:     MOV     ADDR15,R0       ;PUT DEVICE 15 CSR ADDRESS IN R0
006124' 000167  174330                  JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 15

006130'                         OISR15:
                                        ;-------------------------------------------------------------
006130' 000004  000000' 006136'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-------------------------------------------------------------
006136' 016700  000004          1$:     MOV     ADDR15,R0       ;PUT DEVICE 15 CSR ADDRESS IN R0
006142' 000167  175060                  JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 15

006146' 000000                  ADDR15: .WORD   0               ;DEVICE CSR ADDRESS STORED HERE


006150'                         IISR16:
                                        ;-------------------------------------------------------------
006150' 000004  000000' 006156'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-------------------------------------------------------------
006156' 016700  000022          1$:     MOV     ADDR16,R0       ;PUT DEVICE 16 CSR ADDRESS IN R0
006162' 000167  174272                  JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 16

006166'                         OISR16:
                                        ;-------------------------------------------------------------
006166' 000004  000000' 006174'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-------------------------------------------------------------
006174' 016700  000004          1$:     MOV     ADDR16,R0       ;PUT DEVICE 16 CSR ADDRESS IN R0
006200' 000167  175022                  JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 16

006204' 000000                  ADDR16: .WORD   0               ;DEVICE CSR ADDRESS STORED HERE


006206'                         IISR17:
                                        ;-------------------------------------------------------------
006206' 000004  000000' 006214'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-------------------------------------------------------------
006214' 016700  000022          1$:     MOV     ADDR17,R0       ;PUT DEVICE 17 CSR ADDRESS IN R0
```



```
006220' 000167  174234                  JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 17

006224'                         OISR17:
                                        ;-------------------------------------------------------------
006224' 000004  000000' 006232'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-------------------------------------------------------------
006232' 016700  000004          1$:     MOV     ADDR17,R0       ;PUT DEVICE 17 CSR ADDRESS IN R0
006236' 000167  174764                  JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 17

006242' 000000                  ADDR17: .WORD   0               ;DEVICE CSR ADDRESS STORED HERE
```

```
 1892
 1893
 1894
 1895
 1896
 1897  006244' 002000                BUFIN:  .BLKB   1024.   ;INPUT BUFFER (1024 BYTES)
 1898
 1899
 1900
 1901
 1902
 1903                                ;********** PATCH AREA FOR DEBUG *****************
 1904  010244' 000036                PATCH:  .BLKW   30.
 1905                                ;***********************************************
 1906
 1907
 1908
 1909
 1910
 1911          000001                .END
```



```
ABORT = 000004            582#
ACSR    000102R           278#  1576#
ADDR    000006R           244#   812    900   1078   1117   1218   1556
ADDR0   005340R          1608   1615   1618#
ADDR1   005376R          1626   1633   1636#
ADDR10  005720R          1752   1759   1762#
ADDR11  005756R          1770   1777   1780#
ADDR12  006014R          1788   1795   1798#
ADDR13  006052R          1806   1813   1816#
ADDR14  006110R          1824   1831   1834#
ADDR15  006146R          1842   1849   1852#
ADDR16  006204R          1860   1867   1870#
ADDR17  006242R          1878   1885   1888#
ADDR2   005434R          1644   1651   1654#
ADDR22= 001000            302#
ADDR3   005472R          1662   1669   1672#
ADDR4   005530R          1680   1687   1690#
ADDR5   005566R          1698   1705   1708#
ADDR6   005624R          1716   1723   1726#
ADDR7   005662R          1734   1741   1744#
ASB     000106R           282#
ASTAT   000104R           280#  1577*
AWAS    000110R           283#
AX1   = 000001            613#   673#
AX2   = 000002            612#   672#
BABTRB= 000026            543#
BACCIR= 000000            383#  1153
BACCIT= 000004            382#  1166
BACCOR= 000000            392#  1227
BACCOT= 000004            391#  1225
BACORU= 000003            393#  1229
BACOTN= 000007            395#
BACOTS= 000006            394#
BADIN1= 000004            724#   838   1431   1453
BADRED= 000036            353#  1286   1299   1312
BADWRT= 000037            354#  1249   1262   1275
BA0   = 000001            505#
BA1   = 000002            504#
BA10  = 002000            495#
BA11  = 004000            494#
BA12  = 010000            493#
BA13  = 020000            492#
BA14  = 040000            491#
BA15  = 100000            490#
BA16  = 040000            513#
BA17  = 100000            512#
BA2   = 000004            503#
BA3   = 000010            502#
BA4   = 000020            501#
BA5   = 000040            500#
BA6   = 000100            499#
BA7   = 000200            498#
BA8   = 000400            497#
BA9   = 001000            496#
BCC   = 000001            649#
BEGIN   000000R           241#   797    862    863    882    943    944    952    962    978    987    996    997
```

```
                    1005  1020  1067  1091  1134  1141  1241  1251  1264  1277  1288  1301  1314
                    1335  1348  1355  1368  1369  1373  1374  1387  1388  1394  1395  1437  1438
                    1443  1451  1521  1522  1530  1606  1613  1624  1631  1642  1649  1660  1667
                    1678  1685  1696  1703  1714  1721  1732  1739  1750  1757  1768  1775  1786
                    1793  1804  1811  1822  1829  1840  1847  1858  1865  1876  1883
BIRDON= 000010        712#  1125  1162
BITDON= 000020        713#  1123  1176
BIT0  = 000001        302#   319#   335   340   418   438   483   505   527   575   584   613   625
                      637   649   661   673   693   700   710   722   814   901  1079  1557
BIT1  = 000002        302#   318#   341   437   482   504   526   574   583   602   612   624   636
                      648   660   672   692   701   723
BIT10 = 002000        302#   309#   495   517
BIT11 = 004000        302#   308#   494   516
BIT12 = 010000        302#   307#   493   515
BIT13 = 020000        302#   306#   492   514
BIT14 = 040000        302#   305#   491   513
BIT15 = 100000        302#   304#   490   512
BIT2  = 000004        302#   317#   436   503   525   573   582   601   611   623   635   647   659
                      671   691   702   711   724
BIT3  = 000010        302#   316#   426   435   502   524   572   581   600   610   622   634   646
                      658   670   690   703   712
BIT4  = 000020        302#   315#   417   425   434   481   501   523   571   599   609   621   633
                      645   657   669   678   689   713
BIT5  = 000040        302#   314#   461   480   500   522   556   570   591   598   608   620   632
                      644   656   668   679   688   714
BIT6  = 000100        302#   313#   433   460   479   499   521   555   569   590   597   607   619
                      631   643   655   667   680   687   715
BIT7  = 000200        302#   312#   416   424   432   459   478   498   520   568   580   589   596
                      618   630   642   654   666   681   686
BIT8  = 000400        302#   311#   497   519
BIT9  = 001000        302#   310#   496   518
BORDON= 000040        714#  1317
BOTDON= 000100        715#
BPOLL = 000100        590#
BREAKS= 104407        302#   862   863   943   944   996   997  1368  1369  1373  1374  1387  1388
                     1394  1395  1437  1438  1521  1522
BR1     000012R       246#   842   847
BR2     000013R       247#
BSEL0 = 000000        362#   886*  1038*  1083*  1107*  1114*  1136*  1150*  1163*  1177*  1190*  1205*  1243*
                     1280*  1318*  1324*  1329*  1491*  1519*  1560*
BSEL1 = 000001        363#   830*   833*   834*   835*   851*   864*   867*  1366*  1385*  1407*  1409*  1410*
                     1422*  1433
BSEL10= 000010        374#
BSEL11= 000011        375#
BSEL2 = 000002        365#  1025*  1030*  1182*  1184  1187  1223  1346*  1357*  1370  1372*  1389  1392*
                     1396
BSEL3 = 000003        366#  1110*  1181*
BSEL4 = 000004        368#  1471
BSEL5 = 000005        369#
BSEL6 = 000006        371#  1148*  1258  1295  1321  1326  1339  1446  1476
BSEL7 = 000007        372#  1156*  1169*  1254  1291
BTOD$ = 104421        302#
BUFFEX  000252R       728#  1068*  1069*  1070*  1073*  1074*  1075*  1077  1156  1161  1296
BUFIN   006244R       289   1897#
BUFRTC= 000020        559#  1013  1341
BUFISM= 000300        545#
```



```
CARR  = 000001        661#
CCMSK = 140000        407#  1271  1308
CC0   = 000001        527#
CC1   = 000002        526#
CC10  = 002000        517#
CC11  = 004000        516#
CC12  = 010000        515#
CC13  = 020000        514#
CC2   = 000004        525#
CC3   = 000010        524#
CC4   = 000020        523#
CC5   = 000040        522#
CC6   = 000100        521#
CC7   = 000200        520#
CC8   = 000400        519#
CC9   = 001000        518#
CDATA$= 104412        302#
CDERCT  000144R       297#
CDWDCT  000146R       298#
CHKMOR  003164R      1184#  1358
CLRCMD  005104R       806   1543#
CLRCSR  004706R      1421   1454  1491#
CMDMSK= 000370        403#   955  1008  1224
CONFIG  000056R       266#
COUNT   000260R       732#
CRC1  = 000100        619#
CRC2  = 000200        618#
CS    = 000004        659#
CSEL0   000272R       737#
CSEL10  000302R       741#  1546#
CSEL2   000274R       738#   850*   916*   917*  1110  1145*  1153*  1159*  1166*  1172*  1182  1543*
CSEL4   000276R       739#   919*   928*   930*   935*  1111  1544*
CSEL6   000300R       740#   858*   860*   875*   877*   920*   922*   931*   938*   991*  1112  1545*
CSRA    000100R       276#  1575*
CTLIN = 000001        381#   916  1145
CTLOUT= 000001        390#   956  1232
DATCK$= 104411        302#
DATERR= 000001        348#
DATER$= 104404        302#
DDCMP = 000001        625#   693#
DEADTR= 000022        541#
DERINB= 000113        471#
DEROTB= 000112        470#
DEVPTR  000266R       735#   814*   817   819*   901*   903   905*  1079*  1081  1088*  1557*  1558  1562*
DISABL  005126R       796   1347  1556#
DISCMP= 000002        482#
DISCON= 000304        547#
DISPIG= 000200        478#
DISSI = 000040        608#   668#
DMVBIT= 000001        340#
DMVLUP  004242R      1366#
DMVRD   004314R      1385#  1474
DRPMOD  000476R       796#   868   884  1028
DTR   = 000100        597#   655#
DVID1   000014R       248#   794
EAMSK = 000077        408#  1255  1292
```

```
EBLK   = 000002          648#
ENABLE  002276R         1050    1065#
ENABSW= 000200           681#    854     871     1480
ENACMP= 000001           483#
ENAX   = 000004          611#    671#
ENBPIG= 000100           479#
ENDIT$= 104413           302#    1348
END$   = 104410          302#    797
EOM    = 000002          583#
ERINB  = 000013          561#
EROTB  = 000012          560#
ERRORS  000320R          748#    804*    838     865     1431*   1453*   1511*   1532*
ERRTYP  000106R          281#    880*    950*    959*    975*    984*    1003*   1017*   1132*   1139*   1239*   1249*   1262*
                        1275*   1286*   1299*   1312*   1332*   1352*   1441*   1449*   1528#
ESTRIB= 000001           462#    922
EXECUT  004404R         1407#   1469
EXIT$ = 104400           302#    1091
FDCS   = 000005          450#
FDPP   = 000003          448#    858
FDPPDC= 000001           446#
FDTS   = 000007          452#    875
FLAGS   000316R          747#    805*    923*    932*    939*    945     992*    998     1023*   1080*   1108    1113*   1206
                        1215*   1512*   1513*   1523    1533*
FREE    000150R          299#
GETERR  005170R          879     949     958     974     983     1002    1016    1131    1138    1238    1248    1261    1274
                        1285    1298    1311    1331    1351    1440    1448    1527    1575#
GETPA$= 104415           302#    1067
GOAH   = 000010          581#
GOODIN= 000305           323#    1446
GWBUF$= 104414           302#
HALTST= 000005           466#    991
HDCS   = 000004          449#
HDPP   = 000002          447#    860
HDPPDC= 000000           445#
HDTS   = 000006          451#    877
HDX    = 000020          599#    657#
HRDCNT  000044R          261#
HRDER$= 104405           302#    882     952     962     978     987     1005    1020    1134    1141    1241    1251    1264
                        1277    1288    1301    1314    1335    1355    1443    1451    1530
HRDPAS  000050R          263#
IACT   = 000100          643#
IC     = 000200          589#    642#
ICIR   = 000010          690#
ICONT   000036R          258#
ICOUNT  000040R          259#
IDLE   = 000040          620#
IDNUM   000122R          288#
IEI    = 000001          418#    1083    1107    1136    1190    1205    1519
IEO    = 000020          417#    1083    1107    1190    1205    1243    1519
IERR   = 000002          624#
IISR0   005304R          813     1604#
IISR1   005342R         1622#
IISR10  005664R         1748#
IISR11  005722R         1766#
IISR12  005760R         1784#
IISR13  006016R         1802#
```



```
IISR14  006054R         1820#
IISR15  006112R         1838#
IISR16  006150R         1856#
IISR17  006206R         1874#
IISR2   005400R         1640#
IISR3   005436R         1658#
IISR4   005474R         1676#
IISR5   005532R         1694#
IISR6   005570R         1712#
IISR7   005626R         1730#
ILGINT= 000011           350#    1132    1139    1239
IMODX.= 000000           300#
INCMND  004742R          861     924     933     940     993     1511#
INDON   000342R          759#    1065*   1178*
INFOUT= 000002           396#    1010    1235
INIDMP  004442R          837     1420#
ININT  = 000001          700#    1113    1513    1523
INISR   002460R         1106#   1186    1609    1627    1645    1663    1681    1699    1717    1735    1753    1771    1789
                        1807    1825    1843    1861    1879
INIT    000030R          255#
INTABL  000352R          776#    807     809     896     1045    1056    1122    1123    1125    1127    1129    1149*   1162*
                        1176*   1279*   1317*
INTFDG= 000022           469#
INTIMU= 000001           722#    865     1511    1532
INTIMR  000322R          749#    1518*   1525*
INTR    000120R          287#
IRDY   = 000020          645#
ISTART= 000003           464#    938     1148
ISTCNT  000330R          754#    936*    981*
ISTDON= 000004           711#    1057    1127    1149    1279
KILLTR= 000002           463#
LATPS  = 000040          480#
LLKCNT  000326R          751#    815*    894*    1344
LNKDON  000340R          758#
LULOOP= 000010           435#    835     851     864
LULP   = 000040          591#    644#
LURG16  000270R          736#    854     856     871     873     1471*   1476*   1478*   1480*   1482*
MAINT  = 000004          465#
MAINT1= 000010           600#
MAINT2= 000004           601#
MAP22$= 104416           302#
MCLK   = 000002          692#
MCLR   = 000100          433#    830     1422
MDFDON= 000001           710#    1046    1057    1129    1279
MNTHLT= 000014           539#
MNTRUN= 000012           538#
MODDEF= 000002           380#    850
MODESW= 000361           409#
MODE22= 000002           341#    1071    1157    1170    1256    1268    1293    1305
MODMSK= 000370           404#
MODNAM  000000R          242#
MODR   = 000010          658#
MODSP   000252R          256     300#
MODSW0= 000020           678#    856     873
MODSW1= 000040           679#
MODSW2= 000100           680#
```

```
MRDY   = 000200          328#  1370   1389   1396
MSGN$  = 104403          302#
MSGS$  = 104402          302#
MSG$   = 104401          302#
NOINIT= 000034           352#  1441   1449
NOINTR= 000023           351#   950   1003   1528
NONEXM= 000302           546#
NONLUP= 000001           335#
NONQI  = 000004          702#  1108   1512   1533
NONQO  = 000010          703#   805   1080   1206
NORET  = 000000          557#
NOTDFN= 000000           347#   959    975    984   1017   1332   1352
NULL   = 000000          302#   882
N.DEVS  000254R          730#   803*   885*   894
OC     = 000200          580#
OCOR   = 000020          689#
OISR0   005322R         1611#
OISR1   005360R         1629#
OISR10  005702R         1755#
OISR11  005740R         1773#
OISR12  005776R         1791#
OISR13  006034R         1809#
OISR14  006072R         1827#
OISR15  006130R         1845#
OISR16  006166R         1863#
OISR17  006224R         1881#
OISR2   005416R         1647#
OISR3   005454R         1665#
OISR4   005512R         1683#
OISR5   005550R         1701#
OISR6   005606R         1719#
OISR7   005644R         1737#
OPEN   = 000000           243   249    250    251    252    269    270    271    272    273    274    275    276
                          278   280    282    283    285    286    287    290    291    293    294    296    297
                          298   299   302#
OTUA$  = 104420           302#
OUTDON  000344R          761#  1066*  1343*  1344
OUTIMU= 000002           723#
OUTIMR  000324R          750#   941*   947*   994*  1000*  1024*
OUTINT= 000002           701#   923    932    939    945    992    998   1023   1215
OUTISR  003226R         1189  1204#  1616   1634   1652   1670   1688   1706   1724   1742   1760   1778   1796
                        1814   1832   1850   1868   1886
OVRR   = 000010          646#
PASCNT  000034R          257#
PATCH   010244R         1904#
PIRQ$  = 000004           302#  1606   1613   1624   1631   1642   1649   1660   1667   1678   1685   1696   1703
                         1714  1721   1732   1739   1750   1757   1768   1775   1786   1793   1804   1811   1822
                         1829  1840   1847   1858   1865   1876   1883
POLL   = 000200          596#
POPSP  = 005726          302#
POPSP2= 022626           302#
PRTY   = 000000          302#
PRTY0  = 000000          302#
PRTY1  = 000040          302#
PRTY2  = 000100          302#
PRTY3  = 000140          302#
```



```
PRTY4  = 000200          302#
PRTY5  = 000240          246    247    302#
PRTY6  = 000300          302#
PRTY7  = 000340          302#
PS     = 177776          302#
PSW    = 177776          302#
PUSH   = 005746          302#
PUSH2  = 024646          302#
PWRFLG= 000002           302#
QUOVRN= 000306           548#
Q22BIT= 000010           426#  1159   1172
RAB    = 000004          647#
RAND$  = 104417          302#
RANNUM  000054R          265#
RBUFEA  000130R          291#  1068
RBUFPA  000126R          290#  1076   1154   1283
RBUFSZ  000132R          292#
RBUFVA  000124R          289#  1067
RDALL  = 000004          623#
RDAX   = 000020          609#   669#
RDCTSS= 000100           460#   555#
RDLU16= 021344           325#  1470
RDYI   = 000020          425#  1184
RDYO   = 000200          424#  1187
READY  = 000200          666#
READ16  004622R          836   1466#
REDLUC= 000001           329#  1392
REDMDM= 000020           467#
REDTSS= 000040           461#   556#
REENTR  002252R         1040  1055#
REGTBL  000452R          781#   952    962    978    987   1005   1020   1134   1141   1241   1251   1264   1277
                        1288  1301   1314   1335   1355   1443   1451   1530   1578*  1579*  1580*  1581*  1582*
                        1583*  1584*  1587*  1589*  1590*
REQMSK= 000340           405#
RESTRT  002224R          284   1044#  1350
RES1    000056R          267#
RES2    000060R          268#
RETMSI= 000010           558#
RING   = 000200          654#
RINGDI= 000020           540#
ROMI   = 000002          437#  1407   1409   1410
ROMO   = 000004          436#  1407   1409   1410
RQI    = 000200          416#  1083   1114   1136   1150   1163   1177   1280   1318   1324   1329   1519
RSEL0   000304R          742#  1208*  1514*
RSEL10  000314R          746#  1214*
RSEL2   000306R          743#   955*   956   1008*  1010   1209*  1515*
RSEL4   000310R          744#  1210*  1516*
RSEL6   000312R          745#   966    970    972   1012*  1013   1211*  1517*
RSTRT   000112R          284#
RTHRER= 000002           534#
RTNINT  003214R         1188  1190#  1216
RTNMON  002454R         1090#  1137   1191   1244
RTNMSK= 000340           406#  1012   1340
RTS    = 000040          656#
RUN    = 000200          432#   833    834    867   1433
RUNST  = 000024          542#   966   1326
```

CXDMEB.P11 25-MAR-81 08:25 CROSS REFERENCE TABLE -- USER SYMBOLS

```
RX0    = 000001          637#
RX1    = 000002          636#
RX2    = 000004          635#
RX3    = 000010          634#
RX4    = 000020          633#
RX5    = 000040          632#
RX6    = 000100          631#
RX7    = 000200          630#
SAVBF    000262R         733#
SBADR    000102R         277#
SCRACH   000332R         755#
SECA   = 000020          621#
SELECT   000264R         734#   794*   817    903    1081   1558
SELERR= 000007           349#   880
SELFR  = 000040          598#
SELSBY= 000002           602#
SEL0   = 000000          361#  1027*  1208   1576
SEL10  = 000010          373#  1160*  1173*  1214   1270   1307   1497*
SEL2   = 000002          364#  1209   1492*  1577
SEL4   = 000004          367#  1111*  1146*  1154*  1167*  1210   1246   1283   1391*  1493*
SEL6   = 000006          370#  1112*  1147*  1155*  1161*  1168*  1174*  1211   1267   1304   1408*  1494*
SETUP    000506R         795    802#
SIGQ   = 000100          687#
SIGR   = 000200          686#
SUFCNT   000042R         260#
SUFERS= 104406           302#
SUFPAS   000046R         262#
SOM    = 000001          584#
SPOINT   000032R         256#
SPSIZ  = 000040            1#   300
SR1      000016R         249#   852    869    897    1031
SR2      000020R         250#
SR3      000022R         251#   899
SR4      000024R         252#   824    831    909    926    1071   1084   1118   1157   1170   1212   1219   1256
                        1268   1293   1305   1423   1466   1495   1585
START    000466R         255    794#
STARUN= 000010           537#  1321
STAT     000026R         254#
STBY   = 000002          660#
STEPLU= 000020           434#
STEPMP= 000001           438#  1409   1410
STHRER= 000006           536#   972
STRIP  = 000010          622#
STRTRB= 000030           544#
STRTUP   001224R         821    893#
SVR0     000062R         269#
SVR1     000064R         270#
SVR2     000066R         271#
SVR3     000070R         272#
SVR4     000072R         273#
SVR5     000074R         274#
SVR6     000076R         275#
SWPBOT= 121000           327#  1475
SYSCNT   000052R         264#
TESTMO= 000004           691#
TOTAL    000256R         731#
```


CXDMEB.P11 25-MAR-81 08:25 CROSS REFERENCE TABLE -- USER SYMBOLS

```
TRBADR   000334R         756#   895*   899*   917    918    1033*  1034   1036*  1122*  1181
TRBMAX   000336R         757#
TRPDFD= 000022           302#
TTHRER= 000004           535#   970
TTLOOP= 000006           330#  1372
TXDATA= 000040           688#
TXEN   = 000100          607#   667#
TX0    = 000001          575#
TX1    = 000002          574#
TX2    = 000004          573#
TX3    = 000010          572#
TX4    = 000020          571#
TX5    = 000040          570#
TX6    = 000100          569#
TX7    = 000200          568#
ULAIPS= 000020           481#
VECTOR   000010R         245#   811
WASADR   000104R         279#
WAX    = 000010          610#   670#
WBFEA    000350R         764#  1077*  1169   1174   1259
WBFPA    000346R         763#  1076*  1167   1246
WBUFEA   000136R         294#
WBUFPA   000134R         293#
WBUFRQ   000140R         295#
WBUFSZ   000142R         296#
WDFR     000116R         286#
WDTO     000114R         285#
WRIMDM= 000021           468#
WRITSS= 000200           459#   931
XFLAG    000005R         243#
.      = 010340R         776#  1897#  1904#
```

CXDMEB.P11 25-MAR-81 08:25 CROSS REFERENCE TABLE -- MACRO NAMES

```
BKMOD       1#
BREAK       1#    862    942    995   1367   1373   1386   1393   1437   1520
BTOD        1#
CKDATA      1#
DATACK      1#
DATERR      1#
DFSEVN      1#    302
DSEVNT      1#    302
END         1#
ENDIT       1#   1348
ENDMOD      1#    797
EQUATS      1#    302
EXIT        1#   1090
GETPA       1#   1067
GWBUFF      1#
HRDER       1#    881    951    961    977    986   1004   1019   1133   1140   1240   1250   1263   1276   1287
          1300   1313   1334   1354   1442   1450   1529
IOMOD       1#
IOMODP      1#
IOMODR      1#
IOMODX      1#    237
ISRS     1603#   1604   1622   1640   1658   1676   1694   1712   1730   1748   1766   1784   1802   1820   1838
          1856   1874
MAP22       1#
MODULE      1#    237
MSG         1#
MSGN        1#
MSGS        1#
NBKMOD      1#
OTOA        1#
PIRQ        1#   1604   1611   1622   1629   1640   1647   1658   1665   1676   1683   1694   1701   1712   1719
          1730   1737   1748   1755   1766   1773   1784   1791   1802   1809   1820   1827   1838   1845   1856
          1863   1874   1881
RAND        1#
SBKMOD      1#
SOFER       1#

. ABS.  000000    000
        010340    001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XDMEB0,CXDMEB,SEQ/CHF/SOL/NL:TOC=DDXCOM.P11,CXDMEB.P11
 RUN-TIME: 2 4 .8 SECONDS
 RUN-TIME RATIO: 57/8=6.9
 CORE USED:  10K  (19 PAGES)
```