IDENTIFICATION

PRODUCT CODE:	AC-E953B-MC
PRODUCT NAME:	CXDMCB0 DMC-11 MODULE
PRODUCT DATE:	SEPTEMBER 1978
MAINTAINER:	DEC/X11 SUPPORT GROUP

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

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

MAIN DEC CHANGE NOTICE
MAY BE REQUIRED FOR
PROGRAM TO OPERATE

```
        1.      ABSTRACT

                DMC IS AN IOMOD THAT EXERCISES UP TO AND INCLUDING TWO
                CONSECUTIVELY ADDRESSED AND CONSECUTIVELY VECTORED DMC11
                SYNCHRONOUS INTERFACES. IT USES MAINTENANCE MODE OR A TURN
                AROUND CONNECTOR TO TRANSMIT AND RECEIVE A SET DATA
                PATTERN. THE RECEIVER AND TRANSMITTER ISR ARE ALWAYS AT
                PRIORITY FIVE (BR1,BR2). DATA CHECKING IS PERFORMED AT
                LEVEL 0 AND DONE OUTSIDE THE ISRS.

        2.      REQUIREMENTS

                HARDWARE:       AT LEAST 1 DMC11-AR WITH A
                                DMC11-DA OR DMC11-FA
                                        OR
                                AT LEAST 1 DMC11-AL WITH A
                                DMC11-MA OR A DMC11-MD

                STORAGE:: DMC REQUIRES:
                                1. DECIMAL WORDS: 1578
                                2. OCTAL WORDS: 03052
                                3. OCTAL BYTES: 6124


        3.      PASS DEFINITION

                ONE PASS OF THE DMC MODULE CONSISTS OF TRANSMITTING AND
                RECEIVING 7 BUFFERS OF 100 CHARACTERS 200 TIMES FOR EACH SELECTED
                DEVICE.

        4.      EXECUTION TIME

                RUNNING ALONE ON AN 11/45 ONE PASS TAKES APPROXIMATELY
                ONE MINUTE.

        5.      CONFIGURATION PARAMETERS.

                DEFAULT PARAMETERS:
                ADDR: 1,  VECTOR: 1,  BR1: 5,  BR2: 5,  DVID1: 1,  SR1:0
                DMC WILL RUN UP TO TWO CONSECUTIVELY ADDRESSED AND
                CONSECUTIVELY VECTORED DMC11'S. IF SR1 IS ZERO THEN
                LINE UNIT LOOP (MAINT MODE) IS USED AND UP TO 16
                DMC11S (8 DMC MODULES) MAY BE RUN AT A TIME. IF SR1
                BIT0 = 1 THEN LINE UNIT LOOP IS NOT USED AND A TURN-
                AROUND CONNECTOR MUST BE INSTALLED, THIS ENABLES THE
                DMC TO RUN AT SPEED. IF YOU ARE RUNNING THE DMC11S AT
                SPEED, NO MORE THAN A TOTAL SPEED OF 2 MEGABAUD AT
                FULL DUPLEX WILL RUN DEC/X AT A TIME.THAT IS TO SAY IF
                YOU ADD THE SPEEDS OF ALL DMC11S THAT ARE RUNNING
                THE TOTAL MUST NOT EXCEED 2 MEGABAUD. EXAMPLES:

                        AT THE BAUD OF ONE MEG A MAXIMUM OF TWO DMC11S
                        CAN BE RUN AT A TIME.

                        AT THE BAUD OF 500K A MAXIMUM OF FOUR
                        DMC11S CAN BE RUN AT A TIME.
                        OR ANY COMBINATIN SUCH AS 1 DMC11 AT A MEG
                        AND 2 DMC11S AT 500K BAUD CAN BE RUN AT A TIME,
                        ANY COMBINATION AS LONG AS THE TOTAL BAUD DOES
                        NOT EXCEED 2 MEG.

        6.      DEVICE/OPTION SETUP

                IF SR1 = 0 (LINE UNIT LOOP MODE) THEN THERE IS NO
                SPECIAL SET UP NECESSARY.
                IF SR1 BIT0 = 1 (RUNNING THE DMC AT SPEED) THEN
                A TURN-AROUND CONNECTOR MUST BE INSTALLED.

                NOTE:   SR1 CAN BE SET UP AT CONFIGURATION TIME OR
                        AT RUN TIME WITH A MOD COMMAND.
        7.      MODULE OPERATION

                1.  LOAD SOFTWARE POINTERS IN LINK TABLE.
                2.  LOAD VECTORS AND PRIORITIES IN TABLE
                3.  ENABLE SELECTED DEVICES.
                4.  SCAN FOR ALL DEVICES TO FINISH
                5.  IF NOT DONE GO TO 4.
                    IF HUNG REPORT SO AND DROP HUNG DEVICE.
                6.  CHECK DATA FOR ALL DEVICES SELECTED.
                7.  DECREMENT ITERATION COUNT
                8.  IF NOT = 0 GO TO 1
                9.  SIGNAL ENDPASS.

                IISR:   INPUT INTERRUPT SERVICE ROUTINE.
                I1. GET INTERRUPTING DMCSCR.
                I2. IF BASE I WAS REQUESTED, LOAD BASE ADDRESS.
                I3. IF RECEIVE BA/CC WAS REQUESTED, LOAD REC BA/CC.
                I4. IF XMIT BA/CC WAS REQUESTED, LOAD XMIT BA/CC.
                15. RTI

                OISR:     OUTPUT INTERRUPT SERVICE ROUTINE.
                O1. GET INTERRUPTING DMCSCR
                O2. IF ERROR, REPORT IT AND EXIT.
                O3. IF XMIT DONE OR REC DONE, SET APPROPRIATE BITS IN
                    THE ENDPASS FLAG FOR THE DEVICE.
                O4. RTI
```

```
9.      NON-STANDARD PRINTOUTS

        IF THE MODULE "HANGS" IN WHICH NOT ALL SELECTED DEVICES
        HAVE FINISHED, THEN A "HUNG" MESSAGE IS PRINTED OUT.
        CHECK THE ENDPASS FLAGS FOR EACH SELECTED DEVICE IN
        THE LINK TABLE TO DETERMINE WHICH DEVICE FAILED TO
        FINISH AND HOW FAR IT GOT.

        FOR EXAMPLE:

        THE TWO ENDPASS FLAGS ARE LOCATED IN THE LINK TABLE
        (INTLNK) AT THE FOLLOWING LOCATIONS.
        XX11:
        XX21:

        ONLY BITS 0 THRU 4 ARE USED AND ARE DEFINED AS FOLLOWS:
        BIT0 = 1        THE BASE ADDRESS WAS LOADED.
        BIT1 = 1        7 RECEIVE BA/CC'S WERE LOADED.
        BIT2 = 1        7 TRANSMIT BA/CC'S WERE LOADED.
        BIT3 = 1        7 TRANSMIT DONE'S WERE RECEIVED.
        BIT4 = 1        7 RECEIVE DONE'S WERE RECEIVED.

        A CORRECT END PASS FLAG = 37, WHEN THE ENDPASS FLAGS
        = 37 FOR THE SELECTED DEVICES, THE DATA IS CHECKED. IF
        A "HUNG" MESSAGE IS TYPED IT IS BECAUSE ONE OR BOTH
        DEVICES DID NOT FINISH. TO FIND WHICH ONE, CHECK THE
        END PASS FLAGS, ANY THAT ARE NOT EQUAL TO 37 ARE THE
        HUNG DEVICES. CHECK WHICH BITS OF THE ENDPASS FLAG ARE
        CLEAR TO SEE WHAT IT WAS TRYING TO DO.

        SOFT ERROR
        ----------

        IF THE DMC'S PROTOCOL CHECKERS DETECT AN ERROR IN THE TRANSMISSION
        OF A MESSAGE, IT WILL RETRANSMIT THE ENTIRE MESSAGE, UPDATING
        AN ERROR COUNTER IN IT'S RAM.  IF THIS COUNTER EXCEEDS 7 ON ANY
        GIVEN MESSAGE, IT WILL DECLARE A HARD ERROR.  HOWEVER, IF FEWER
        THAN 7 OCCUR, IT WILL TAKE NO NOTICE OF THE CONDITION.  FOR DEC/
        X11 PURPOSES, HOWEVER, THE DMC MODULE WILL CHECK THE ERROR COUNTER
        AFTER EACH MESSAGE; IF IT HAS BEEN INCREMENTED AT ALL, I.E., IF
        AT LEAST ONE RE-TRANSMISSION WAS MADE, DMC WILL DECLARE A SOFT
        ERROR.
                THE SOFT ERROR MESSAGE MAY INDICATE AN INTERMITTANT DE-
        VICE FAILURE OR OTHER HARDWARE PROBLEM; HOWEVER, IF THE MESSAGE
        OCCURS IN A HEAVILY LOADED SYSTEM, IT MAY BE THAT THE PROBLEM
        IS DUE TO BUS LATENCY (THE DMC-11 DOES NOT RECOGNISE A DISTINCT
        "DATA LATE" ERROR--IT CONSIDERS THE CONDITION MERELY ANOTHER
        TRANSMISSION PROBLEM).  ESPECIALLY IF THERE ARE OTHER FAST DIRECT
        MEMORY ACCESS DEVICES SELECTED, IT COULD BE THAT THE DMC-11'S
        NPR'S ARE NOT BEING HONORED QUICKLY ENOUGH TO PREVENT BIT-DROPPING.
                TO VERIFY WHETHER THIS IS THE CONDITION, RUN A SINGLE
        DMC MODULE, WITH A SINGLE DMC-11 DEVICE SELECTED.  THE SOFT ERROR
        MESSAGE SHOULD NOT OCCUR UNDER THESE CONDITIONS.  IF
        IT DOES OCCUR, THE PROBLEM IS PROBABLY IN DMC-11 HARDWARE OR A
        CABLE FAULT.
```

```
000000'                                IOMOD <DMCB >,1,1,5,5,0,200,127
000000'                                MODULE  140000,DMCB ,1,1,5,5,0,200,127
                                       .TITLE  DMCB DEC/X11 SYSTEM EXERCISER MODULE
                               ;       DDXCOM  VERSION  6      23-MAY-78
                                       .LIST   BIN
                               ;*********************************************************************
000000'                        BEGIN:
000000' 046504  041103     040 MODNAM: .ASCII  /DMCB / ;MODULE NAME.
000005'    000                 XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 000001                 ADDR:   1+0             ;1ST DEVICE ADDR.
000010' 000001                 VECTOR: 1+0                     ;1ST DEVICE VECTOR.
000012'    240                 BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013'    240                 BR2:    .BYTE   PRTY5+0         ;2ND BR LEVEL.
000014' 000001                 DVID1:  0+1                     ;DEVICE INDICATOR 1.
000016' 000000                 SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000                 SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000                 SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000                 SR4:    OPEN                    ;SWITCH REGISTER 4
                               ;********************************************************************
000026' 140000                 STAT:   140000                  ;STATUS WORD.
000030' 000256'                INIT:   START                   ;MODULE START ADDR.
000032' 000224'                SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                 PASCNT: 0                       ;PASS COUNTER.
000036' 000200                 ICONT:  200                     ;# OF ITERATIONS PER PASS=200
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' 000324'                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' 000127                 IDNUM:  127                     ;MODULE IDENTIFICATION NUMBER=127
000224'                        MODSP:
                               ;*********************************************************************
```

```

                               ;VARIABLES FOR DMC11

000224' 000000                 DLY1:   0
000226' 000000                 DLY2:   0
000230' 000000                 SELECT: 0
000232' 000037                 FLAGB:  37
000234' 000000                 FIRST:  0
000236' 000000                 MASK:   0
000240' 000100                 RCOUNT: 100
000242' 000100                 TCOUNT: 100
000244' 000000                 VA:     0
000246' 000000                 PA:     0
000250' 000000                 EA:     0
000252' 000000                 SAR0:   0
000254' 000000                 SAR1:   0

        000000                 TERM    =0
```

```


                                ;       BEGIN THE TESTS FOR THE DMC11

000256' 012767  000010  177634  START:   MOV     #8.,INTR        ;8 INTERRUPTS/ITERATION
000264' 012767  000700  177622           MOV     #700,WDTO       ;700 WORDS TO MEM/ITERATION
000272' 012767  000700  177616           MOV     #700,WDFR       ;700 WORDS FRON MEM/ITERATION
000300' 032767  177774  177506           BIT     #^C<3>,DVID1    ;DROP MODULE IF DEVICES OTHER
000306' 001004                           BNE     DROP            ;THAN FIRST 4 ARE SELECTED
000310' 016767  177500  177712           MOV     DVID1,SELECT    ;SELECT=ACTIVE DEVICES
000316' 001002                           BNE     RESTRT          ;DROP MODULE IF NO ACTIVE DEVICES
000320'                         DROP:
000320' 104410  000000'                  END$,BEGIN              ;NO DMC'S OR ILLEGAL DMC'S SELECTED
000324' 005067  177704          RESTRT:  CLR     FIRST           ;CLEAR FIRST TIME FLAG
000330' 012700  003324'         LOOP:    MOV     #RBUF11,R0      ;GET SET TO CLEAR BUFFERS
000334' 005020                  1$:      CLR     (R0)+           ;CLEAR BUFFER
000336' 022700  005124'                  CMP     #BASE1,R0       ;END OF BUFFERS?
000342' 001374                           BNE     1$              ;BR IF NO
000344' 016700  177660                   MOV     SELECT,R0       ;R0=ACTIVE BITS
000350' 001763                           BEQ     DROP            ;DROP MODULE IF NO DEVICES ARE SELECTED
000352' 016701  177430                   MOV     ADDR,R1         ;R1=DEVICE CSR
000356' 016702  177426                   MOV     VECTOR,R2       ;R2=VECTOR
000362' 012703  002776'                  MOV     #INTLNK,R3      ;R3=POINTER TO INTERRUPT LINKAGE
000366' 016767  177640  002414           MOV     FLAGB,XX11      ;SET END PASS FLAG FOR DEVICE #1
000374' 016767  177632  002442           MOV     FLAGB,XX21      ;SET END PASS FLAG FOR DEVICE #2
000402' 012767  003170' 002640           MOV     #PIRINQ,INQIN   ;SET UP ALL QUEUES
000410' 012767  003170' 002634           MOV     #PIRINQ,INQOUT
000416' 012767  003210' 002630           MOV     #PIROUTQ,OUTQIN
000424' 012767  003210' 002624           MOV     #PIROUTQ,OUTQOUT
000432' 012767  003230' 002620           MOV     #REGQ,REGQI
000440' 012767  003230' 002614           MOV     #REGQ,REGQO
000446' 006200                  2$:      ASR     R0              ;ACTIVE?
000450' 103410                           BCS     4$              ;BR IF ACTIVE
000452' 001452                           BEQ     SETUP2          ;BR IF DONE
000454' 062701  000010          3$:      ADD     #10,R1          ;UPDATE CSR
000460' 062702  000010                   ADD     #10,R2          ;UPDATE VECTOR
000464' 062703  000034                   ADD     #34,R3          ;UPDATE LINK
000470' 000766                           BR      2$              ;CONTINUE
000472' 010312                  4$:      MOV     R3,(R2)         ;LOAD VECTOR
000474' 116762  177312  000002           MOVB    BR1,2(R2)       ;LOAD INTERRUPT LEVEL
000502' 010163  000010                   MOV     R1,10(R3)       ;LOAD CSR TO LINKAGE
000506' 010362  000004                   MOV     R3,4(R2)        ;LOAD LINKAGE ADDRESS IN VECTOR
000512' 062762  000004  000004           ADD     #4,4(R2)        ;ADJUST IT
000520' 116762  177266  000006           MOVB    BR1,6(R2)       ;LOAD INTERRUPT LEVEL
000526' 005063  000020                   CLR     20(R3)          ;CLEAR INPUT OFFSET LOCATION
000532' 005063  000022                   CLR     22(R3)          ;CLEAR OUTPUT COUNT LOCATION
000536' 005063  000012                   CLR     12(R3)          ;CLEAR END PASS FLAG
000542' 005767  177466                   TST     FIRST           ;BEGINNING OF A PASS?
000546' 001011                           BNE     5$              ;BR IF NOT
000550' 005063  000024                   CLR     24(R3)          ;CLEAR AREA TO SAVE
000554' 005063  000026                   CLR     26(R3)          ;BASE TABLE ERROR
000560' 005063  000030                   CLR     30(R3)          ;COUNTS FOR COMPARISION
000564' 005063  000032                   CLR     32(R3)          ;
000570' 000731                           BR      3$              ;CONTINUE
000572' 005263  000012          5$:      INC     12(R3)          ;SET BITO OF ENDPASS FLAG
000576' 000726                           BR      3$              ;CONTINUE
000600' 016701  177202          SETUP2:  MOV     ADDR,R1         ;R1=DEVICE CSR
```

```
000604' 016700  177420                   MOV     SELECT,R0       ;R0=ACTIVE BITS
000610' 006200                  1$:      ASR     R0              ;ACTIVE?
000612' 103404                           BCS     3$              ;BR IF YES
000614' 001454                           BEQ     SCAN            ;BR IF DONE
000616' 062701  000010          2$:      ADD     #10,R1          ;UPDATE CSR
000622' 000772                           BR      1$              ;CONTINUE
000624' 005767  177404          3$:      TST     FIRST           ;FIRST PASS?
000630' 001002                           BNE     4$              ;MASTER CLEAR FIRST TIME ONLY
000632' 012711  040000                   MOV     #BIT14,(R1)     ;MASTER CLEAR
000636' 005711                  4$:      TST     (R1)            ;RUN SET?
000640' 100415                           BMI     5$              ;BR IF YES
000642' 010067  177404                   MOV     R0,SAR0         ;SAVE R0
000646' 010167  177402                   MOV     R1,SAR1         ;SAVE R1
000652' 104407  000000'                  BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
000656' 104407  000000'                  BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
000662' 016700  177364                   MOV     SAR0,R0         ;RESTORE R0
000666' 016701  177362                   MOV     SAR1,R1         ;RESTORE R1
000672' 000761                           BR      4$              ;WAIT FOR RUN
000674' 052761  000100  000002  5$:      BIS     #100,2(R1)      ;SET IEO
000702' 032767  000001  177106           BIT     #BIT0,SR1       ;IS SR1 ZERO?
000710' 001003                           BNE     6$              ;BR IF NO (TURNAROUND CONNECTOR)
000712' 052711  004000                   BIS     #4000,(R1)      ;OTHERWISE SET LU LOOP
000716' 000402                           BR      7$              ;CONTINUE
000720' 042711  004000          6$:      BIC     #4000,(R1)      ;IF SR1 BITO IS SET,CLEAR LU LOOP
000724' 005767  177304          7$:      TST     FIRST           ;FIRST TIME HERE?
000730' 001003                           BNE     8$              ;BR IF NO
000732' 052711  000143                   BIS     #143,(R1)       ;SET LU LOOP, IEI,RQI,BASEI
000736' 000727                           BR      2$              ;CONTINUE NEXT DEVICE
000740' 052711  000144          8$:      BIS     #144,(R1)       ;SET LU LOOP, IEI, REC BA/CC
000744' 000724                           BR      2$              ;CONTINUE NEXT DEVICE
```

```
                                  ;----------------------------------------
                                  ;SCAN ALL ENDPASS FLAGS UNTIL
                                  ;ALL ACTIVE DMC11S ARE FINISHED
                                  ;     (ENDPASS FLAG = 37)
                                  ;----------------------------------------
000746' 012767  000003   177262   SCAN:    MOV     #3,MASK             ;SET BIT FOR ALL DEVICES
000754' 012767  000010   177242            MOV     #10,DLY1            ;DELAY COUNT
000762' 005067  177240                     CLR     DLY2                ;DELAY
000766' 026767  177240   002014   1$:      CMP     FLAGB,XX11          ;DEVICE 1 DONE?
000774' 001003                             BNE     2$                  ;BR IF NO
000776' 042767  000001   177232            BIC     #1,MASK             ;DEVICE 1 IS DONE SO CLEAR BIT0
001004' 026767  177222   002032   2$:      CMP     FLAGB,XX21          ;DEVICE 2 DONE?
001012' 001003                             BNE     3$                  ;BR IF NO
001014' 042767  000002   177214            BIC     #2,MASK             ;DEVICE 2 IS DONE SO CLEAR BIT1
001022' 005767  177210            3$:      TST     MASK                ;ARE ALL DEVICES FINISHED?
001026' 001146                             BNE     16$                 ;BR IF NO
001030' 012701  003006'                    MOV     #INTLNK+10,R1       ;R1 IS POINTER TO DEVICE CSR
001034' 016700  177170                     MOV     SELECT,R0           ;R0 CONTAINS BITS FOR ACTIVE DEVICES
001040' 012703  003264'                    MOV     #BUFTAB,R3          ;R3 IS POINTER TO DEVICE RECEIVER BUFFER
001044' 006200                    4$:      ASR     R0                  ;ACTIVE?
001046' 103414                             BCS     8$                  ;BR IF YES
001050' 001404                             BEQ     6$                  ;BR IF DONE
001052' 062701  000034            5$:      ADD     #34,R1              ;UPDATE R1 TO NEXT DEVICE CSR
001056' 005723                             TST     (R3)+               ;UPDATE R3 TO NEXT BUFFER
001060' 000771                             BR      4$                  ;CONTINUE
001062' 012767  177777   177144   6$:      MOV     #-1,FIRST           ;SET FIRST FLAG TO -1
001070' 104413  000000'                    ENDIT$,BEGIN                ;SIGNAL END OF ITERATION.
                                                                       ;MONITOR SHALL TEST END OF PASS
001074' 000167  177230            7$:      JMP     LOOP                ;LOOP MODULE
001100' 012767  000007   177130   8$:      MOV     #7,MASK             ;MASK = BUFFER COUNT
001106' 011302                             MOV     (R3),R2             ;R2 POINTS TO FIRST REC BUFFER
001110' 012204                    9$:      MOV     (R2)+,R4            ;R4 POINTS TO RECEIVER DATA
001112' 012705  003066'                    MOV     #XBUF,R5            ;R5 POINTS TO GOOD DATA
001116' 121514                    10$:     CMPB    (R5),(R4)           ;COMPARE DATA
001120' 001414                             BEQ     11$                 ;BR IF GOOD
001122' 011167  176752                     MOV     (R1),CSRA           ;LOAD CSR
001126' 010567  176750                     MOV     R5,SBADR            ;LOAD GOOD ADDRESS
001132' 010467  176746                     MOV     R4,WASADR           ;LOAD BAD ADDRESS
001136' 111567  176744                     MOVB    (R5),ASB            ;LOAD GOOD DATA
001142' 111467  176742                     MOVB    (R4),AWAS           ;LOAD BAD DATA
                                           ;************************************************************
001146' 104404  000000'                    DATER$,BEGIN                ;DATA ERROR!!!
                                           ;************************************************************
001152' 122524                    11$:     CMPB    (R5)+,(R4)+         ;POP DATA POINTERS
001154' 122715  000000                     CMPB    #TERM,(R5)          ;DONE YET?
001160' 001356                             BNE     10$                 ;BR IF NO
001162' 005367  177050                     DEC     MASK                ;DEC BUFFER COUNT
001166' 001350                             BNE     9$                  ;BR IF NOT ALL 7 BUFFERS CHECKED
001170' 012705  000003                     MOV     #3,R5               ;NOW LETS CHECK BASE TABLE ERROR COUNTS
001174' 016104  000006                     MOV     6(R1),R4            ;GET BASE TABLE ADDRESS
001200' 060504                             ADD     R5,R4               ;ADD OFFSET TO ERROR COUNTS
001202' 010102                             MOV     R1,R2               ;GET POINTER TO CSR
001204' 062702  000014                     ADD     #14,R2              ;MAKE IT POINT TO SAVED ERROR COUNTS
001210' 122224                    12$:     CMPB    (R2)+,(R4)+         ;COMPARE BASE TABLE ERROR COUNTS
001212' 001005                             BNE     13$                 ;TO SAVED ERROR COUNTS BR IF NOT SAME
001214' 005205                             INC     R5                  ;BUMP TO NEXT ERROR COUNT
```

```
001216' 022705  000013                     CMP     #13,R5              ;ALL DONE YET?
001222' 001372                             BNE     12$                 ;BR IF NO
001224' 000712                             BR      5$                  ;CONTINUE NEXT DEVICE
001226'                           13$:
001226' 104403  000000'  001576'           MSGN$,BEGIN,SOFT            ;ASCII MESSAGE CALL WITH COMMON HEADER
001234' 012705  000003                     MOV     #3,R5               ;BASE OFFSET TO ERROR COUNTS
001240' 012702  001602'                    MOV     #ESAV1,R2           ;LOAD COUNTS FOR TYPEOUT
001244' 016104  000006                     MOV     6(R1),R4            ;GET BASE ADDRESS
001250' 060504                             ADD     R5,R4               ;ADD IN OFFSET
001252' 112422                    14$:     MOVB    (R4)+,(R2)+         ;OK LOAD TABLE FOR TYPEOUT
001254' 005205                             INC     R5                  ;INCREMENT COUNTER
001256' 022705  000013                     CMP     #13,R5              ;DONE YET?
001262' 001373                             BNE     14$                 ;BR IF NOT
                                           ;SAVE THESE BASE TABLE ERROR COUNTS
001264' 010102                             MOV     R1,R2               ;GET POINTER TO CSR
001266' 062702  000014                     ADD     #14,R2              ;MAKE IT POINT TO SAVED ERROR COUNTS
001272' 012704  001602'                    MOV     #ESAV1,R4           ;R4 POINTS TO NEW COUNT VALUES
001276' 012422                    15$:     MOV     (R4)+,(R2)+         ;STORE BASE TABLE COUNTS
001300' 022704  001612'                    CMP     #ESAV4+2,R4         ;DONE?
001304' 001374                             BNE     15$                 ;BR IF NOT
001306' 011167  176566                     MOV     (R1),CSRA           ;LOAD CSR
001312' 011105                             MOV     (R1),R5
001314' 011567  176562                     MOV     (R5),ACSR           ;SAV CONTENTS OF SEL0
001320' 016167  000006   176556            MOV     6(R1),ASTAT         ;SAVE BASE ADDRESS
001326' 012767  000001   176552            MOV     #1,ERRTYP           ;DATA ERROR
                                           ;************************************************************
001334' 104406  000000'  001626'           SOFER$,BEGIN,FTABLE         ;BASE TABLE DDCMP ERROR COUNTERS
                                           ;************************************************************
001342' 000643                             BR      5$                  ;CONTINUE NEXT DEVICE
001344'                           16$:
001344' 104407  000000'                    BREAK$,BEGIN                ;TEMPORARY RETURN TO MONITOR....
001350' 104407  000000'                    BREAK$,BEGIN                ;THEN CONTINUE AT NEXT INSTRUCTION.
001354' 005367  176646                     DEC     DLY2                ;DEC DELAY COUNT
001360' 001402                             BEQ     .+6                 ;
001362' 000167  177400                     JMP     1$                  ;BR IF NOT DONE
001366' 005367  176632                     DEC     DLY1                ;DEC DELAY COUNT
001372' 001402                             BEQ     .+6                 ;
001374' 000167  177366                     JMP     1$                  ;BR IF NOT DONE
001400' 016700  176632                     MOV     MASK,R0             ;PUT BITS OF HUNG DEVICES IN R0
001404' 040067  176620                     BIC     R0,SELECT           ;DROP ANY HUNG DEVICES
001410' 006000                             ROR     R0                  ;WAS DEVICE 1 HUNG?
001412' 103004                             BCC     17$                 ;BR IF NOT
001414' 004367  000024                     JSR     R3,XERR             ;TYPE ERROR MESSAGE
001420' 003006'                            CSR1                        ;POINTER TO DEVICE #1 CSR
001422' 000001                             1                           ;DEVICE NUMBER FOR TYPEOUT
001424' 006000                    17$:     ROR     R0                  ;WAS DEVICE 2 HUNG?
001426' 103004                             BCC     18$                 ;BR IF NOT
001430' 004367  000010                     JSR     R3,XERR             ;TYPE ERROR MESSAGE
001434' 003042'                            CSR2                        ;POINTER TO DEVICE #2 CSR
001436' 000002                             2                           ;DEVICE NUMBER FOR TYPEOUT
001440' 000167  176664            18$:     JMP     LOOP                ;RESTART MODULE

001444' 012302                    XERR:    MOV     (R3)+,R2            ;GET POINTER TO CSR
001446' 012367  000112                     MOV     (R3)+,DEV           ;GET DEVICE NUMBER
001452' 052767  000060   000104            BIS     #60,DEV             ;MAKE IT ASCII
001460' 104403  000000'  001566'           MSGN$,BEGIN,DROP1           ;ASCII MESSAGE CALL WITH COMMON HEADER
```

```
001466' 011201                                  MOV     (R2),R1         ;GET CSR ADDRESS
001470' 010167  176404                          MOV     R1,CSRA         ;SAVE CSR
001474' 011167  176402                          MOV     (R1),ACSR       ;SAVE CONTENTS OF SEL0
001500' 016167  000002  176376                  MOV     2(R1),ASTAT     ;SAVE CONTENTS OF SEL2
001506' 016167  000004  176510                  MOV     4(R1),DLY1      ;SAVE CONTENTS OF SEL4
001514' 016167  000006  176504                  MOV     6(R1),DLY2      ;SAVE CONTENTS OF SEL6
001522' 016267  000002  000052                  MOV     2(R2),ESAV1     ;END PASS FLAG
001530' 016267  000010  000046                  MOV     10(R2),ESAV2    ;RECEIVE BUFFER OFFSET
001536' 016267  000012  000042                  MOV     12(R2),ESAV3    ;REC/XMIT COUNTERS
001544' 012767  000023  176334                  MOV     #23,ERRTYP      ;NO INTERRUPT
                                                ;*********************************************************************
001552' 104405  000000' 001612'                 HRDERS,BEGIN,ETABLE     ;DUMP DMC CSR'S AND STATUS FLAGS
                                                ;*********************************************************************
001560' 005011                                  CLR     (R1)            ;SHUT OFF HUNG DMC11
001562' 000203                                  RTS     R3              ;RETURN

001564' 000000                          DEV:    0

001566' 001640'                         DROP1:  XDROP1
001570' 001564'                                 DEV
001572' 001661'                                 XDROP2
001574' 177777                                  -1

001576' 001717'                         SOFT:   SOFT1
001600' 177777                                  -1

001602' 000000                          ESAV1:  0                       ;EXTENDED ERROR PRINTOUT LOCATIONS
001604' 000000                          ESAV2:  0
001606' 000000                          ESAV3:  0
001610' 000000                          ESAV4:  0

001612' 000224'                         ETABLE: DLY1                    ;TABLE OF ADDRESSES FOR EXTENDED ERROR PRINTOUT
001614' 000226'                                 DLY2
001616' 001602'                                 ESAV1
001620' 001604'                                 ESAV2
001622' 001606'                                 ESAV3
001624' 177777                                  -1

001626' 001602'                         FTABLE: ESAV1
001630' 001604'                                 ESAV2
001632' 001606'                                 ESAV3
001634' 001610'                                 ESAV4
001636' 177777                                  -1

001640' 042045  041515  030461          XDROP1: .ASCIZ  /%DMC11 DEVICE # /
001646' 042040  053105  041511
001654' 020105  020043     000
001661'    040  051511  044040          XDROP2: .ASCIZ  / IS HUNG AND HAS BEEN DROPPED/
001666' 047125  020107  047101
001674' 020104  040510  020123
001702' 042502  047105  042040
001710' 047522  050120  042105
001716'    000
001717'    045  047523  052106          SOFT1:  .ASCIZ  /%SOFT ERROR - DDCMP ERROR COUNTERS ARE NON ZERO%/
001724' 042440  051122  051117
001732' 026440  042040  041504
```

```
001740' 050115  042440  051122
001746' 051117  041440  052517
001754' 052116  051105  020123
001762' 051101  020105  047516
001770' 020116  042532  047522
001776' 000045
                                        .EVEN
```

```
002000' 010577  001244          IISR:    MOV     R5,@INQIN          ;STORE LINK POINTER IN QUEUE
002004' 062767  000002  001236           ADD     #2,INQIN           ;UPDATE QUEUE
002012' 022767  003210' 001230           CMP     #PIRINQ+20,INQIN;END OF QUEUE?
002020' 001003                           BNE     1$                 ;BR IF NO
002022' 012767  003170' 001220           MOV     #PIRINQ,INQIN      ;RESET QUEUE POINTER
002030' 012605                  1$:      MOV     (SP)+,R5           ;RESTORE R5
                                         ;-------------------------------------------------------------
002032' 000004  000000' 002040'          PIRQ$,BEGIN,2$             ; QUEUE UP TO CONTINUE AT 2$ AND RTI
                                         ;-------------------------------------------------------------
002040' 017705  001206          2$:      MOV     @INQOUT,R5         ;GET LINK POINTER IN R5
002044' 062767  000002  001200           ADD     #2,INQOUT          ;UPDATE QUEUE
002052' 022767  003210' 001172           CMP     #PIRINQ+20,INQOUT;END OF QUEUE?
002060' 001003                           BNE     3$                 ;BR IF NO
002062' 012767  003170' 001162           MOV     #PIRINQ,INQOUT     ;RESET QUEUE POINTER
002070' 016501  000004          3$:      MOV     4(R5),R1           ;LOAD CSR ADDRESS
002074' 032711  000007                   BIT     #7,(R1)            ;XMIT BA/CC?
002100' 001511                           BEQ     XMIT               ;BR IF YES
002102' 032711  000004                   BIT     #4,(R1)            ;REC BA/CC?
002106' 001037                           BNE     REC                ;BR IF YES
002110' 032711  000002                   BIT     #2,(R1)            ;CNTL I?
002114' 001424                           BEQ     4$                 ;BR IF YES
002116' 016567  000012  176120           MOV     12(R5),VA          ;LOAD VA WITH VIRTUAL ADDRESS
002124' 004767  000312                   JSR     PC,EABITS          ;GET PHYSICAL ADDRESS
002130' 016761  176112  000004           MOV     PA,4(R1)           ;LOAD PHYSICAL BASE ADDRESS
002136' 016761  176106  000006           MOV     EA,6(R1)           ;LOAD EA BITS FOR BASE ADDRESS
002144' 004767  000260                   JSR     PC,ENDCLR          ;CLEAR RQI
002150' 052765  000001  000006           BIS     #1,6(R5)           ;SET BIT0 IN END PASS FLAG TO
                                                                    ;SHOW THAT A BASE ADRESS WAS LOADED
002156' 152711  000041                   BISB    #41,(R1)           ;ASK FOR CNTL I
002162' 104400  000000'                  EXIT$,BEGIN                      ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
002166' 005061  000006          4$:      CLR     6(R1)              ;SET FULL DUPLEX
002172' 004767  000232                   JSR     PC,ENDCLR          ;CLEAR RQI
002176' 152711  000044                   BISB    #44,(R1)           ;ASK FOR REC BA/CC I
002202' 104400  000000'                  EXIT$,BEGIN                      ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
002206' 016504  000010          REC:     MOV     10(R5),R4          ;GET REC BUFFER POINTER
002212' 066504  000014                   ADD     14(R5),R4          ;ADD OFFSET
002216' 011467  176022                   MOV     (R4),VA            ;VA GETS REC BUFFER VIRTUAL ADDRESS
002222' 004767  000214                   JSR     PC,EABITS          ;GET PHYSICAL ADDRESS
002226' 016761  176014  000004           MOV     PA,4(R1)           ;LOAD PHYSICAL RBUF
002234' 016761  176010  000006           MOV     EA,6(R1)           ;LOAD RBUF EA BITS
002242' 056761  175772  000006           BIS     RCOUNT,6(R1)       ;LOAD RECEIVER COUNT
002250' 004767  000154                   JSR     PC,ENDCLR          ;CLEAR RQI
002254' 062765  000002  000014           ADD     #2,14(R5)          ;UPDATE OFFSET
002262' 022765  000016  000014           CMP     #16,14(R5)         ;LOADED 7 BUFFERS YET?
002270' 001404                           BEQ     1$                 ;BR IF YES
002272' 152711  000044                   BISB    #44,(R1)           ;REQUEST REC BA/CC
002276' 104400  000000'                  EXIT$,BEGIN                      ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
002302' 052765  000002  000006  1$:      BIS     #2,6(R5)           ;SET BIT1 IN END PASS FLAG TO
                                                                    ;SHOW THAT ALL 7 REC BA/CC WERE LOADED
002310' 005065  000014                   CLR     14(R5)             ;CLEAR OFFSET
002314' 152711  000040                   BISB    #40,(R1)           ;ASK FOR XMIT BA/CC I
002320' 104400  000000'                  EXIT$,BEGIN                      ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
002324' 012767  003066' 175712  XMIT:    MOV     #XBUF,VA           ;LOAD XBUF VIRTUAL ADDRESS
002332' 004767  000104                   JSR     PC,EABITS          ;GET PHYSICAL ADDRESS
002336' 016761  175704  000004           MOV     PA,4(R1)           ;LOAD PHYSICAL XBUF
002344' 016761  175700  000006           MOV     EA,6(R1)           ;LOAD EA BITS
```

```
002352' 056761  175664  000006           BIS     TCOUNT,6(R1)       ;LOAD XMIT COUNT
002360' 004767  000044                   JSR     PC,ENDCLR          ;CLEAR RQI
002364' 005265  000014                   INC     14(R5)             ;COUNT HOW MANT XMIT BA/CC HAVE BEEN LOADED
002370' 022765  000007  000014           CMP     #7,14(R5)          ;DO WE HAVE ALL 7?
002376' 001404                           BEQ     1$                 ;BR IF YES
002400' 152711  000040                   BISB    #40,(R1)           ;ASK FOR XMIT BA/CC
002404' 104400  000000'                  EXIT$,BEGIN                      ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
002410' 052765  000004  000006  1$:      BIS     #4,6(R5)           ;SET BIT2 IN END PASS FLAG TO
                                                                    ;SHOW THAT ALL 7 XMIT BA/CC WERE LOADED
002416' 005065  000014                   CLR     14(R5)             ;CLEAR OFFSET
002422' 105011                           CLRB    (R1)               ;CLEAR IEI
002424' 104400  000000'                  EXIT$,BEGIN                      ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
                                                                    ;
002430' 142711  000040          ENDCLR:  BICB    #40,(R1)           ;CLEAR RQI
002434' 105711                  1$:      TSTB    (R1)               ;IS RDI GONE?
002436' 100776                           BMI     1$                 ;BR IF NO
002440' 000207                           RTS     PC                 ;RETURN

002442'                         EABITS:
002442' 104415  000000' 000244'          GETPA$,BEGIN, VA                 ;GET PHYSICAL ADDRESS FROM 16-BIT VA
002450' 000367  175574                   SWAB    EA                 ;BITS 4+5 TO 13+12
002454' 006167  175570                   ROL     EA                 ;NOW 14+13
002460' 006167  175564                   ROL     EA                 ;NOW 15+14
002464' 042767  037776  175556           BIC     #37776,EA          ;CLEAR ALL BUT 14 & 15
002472' 000207                           RTS     PC                 ;RETURN
```

```
002474' 010577  000554          OISR:   MOV     R5,@OUTQIN          ;MOVE LINK POINTER TO QUEUE
002500' 062767  000002  000546          ADD     #2,OUTQIN           ;UPDATE QUEUE
002506' 022767  003230' 000540          CMP     #PIROUTQ+20,OUTQIN;END OF QUEUE?
002514' 001003                          BNE     1$                  ;BR IF NO
002516' 012767  003210' 000530          MOV     #PIROUTQ,OUTQIN ;RESET QUEUE POINTER
002524' 012605                  1$:     MOV     (SP)+,R5            ;RESTORE R5
                                        ;------------------------------------------------------------
002526' 000004  000000' 002534'         PIRQ$,BEGIN,2$              ; QUEUE UP TO CONTINUE AT 2$ AND RTI
                                        ;------------------------------------------------------------
002534' 017705  000516          2$:     MOV     @OUTQOUT,R5         ;GET LINK POINTER FROM QUEUE
002540' 062767  000002  000510          ADD     #2,OUTQOUT          ;UPDATE QUEUE
002546' 022767  003230' 000502          CMP     #PIROUTQ+20,OUTQOUT;END OF QUEUE?
002554' 001003                          BNE     3$                  ;BR IF NO
002556' 012767  003210' 000472          MOV     #PIROUTQ,OUTQOUT;RESET QUEUE POINTER
002564' 011501                  3$:     MOV     (R5),R1             ;LOAD CSR ADDRESS
002566' 032761  000001  000002          BIT     #BIT0,2(R1)         ;ERROR?
002574' 001422                          BEQ     4$                  ;BR IF NO
002576' 010167  175276                  MOV     R1,CSRA             ;LOAD DEVICE CSR
002602' 016167  000004  175272          MOV     4(R1),ACSR          ;LOAD CONTENTS OF DEVICE CSR
002610' 016167  000006  175266          MOV     6(R1),ASTAT         ;LOAD ERROR BITS
002616' 005067  175264                  CLR     ERRTYP              ;UNKNOWN ERROR
                                        ;************************************************************
002622' 104405  000000' 000000          HRDER$,BEGIN,NULL           ;A CNTL O WAS RECEIVED, ASTAT=ERROR BITS
                                        ;************************************************************
002630' 142761  000207  000002          BICB    #207,2(R1)          ;CLEAR RDO
002636' 104400  000000'                 EXIT$,BEGIN                         ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
002642' 032761  000004  000002  4$:     BIT     #BIT2,2(R1)         ;RECEIVER DONE?
002650' 001025                          BNE     6$                  ;BR IF YES
002652' 142761  000207  000002          BICB    #207,2(R1)          ;CLEAR RDO
002660' 105265  000012                  INCB    12(R5)              ;LO BYTE IS XMIT DONE COUNT
002664' 122765  000007  000012          CMPB    #7,12(R5)           ;DO WE HAVE 7 XMIT DONES YET?
002672' 001012                          BNE     5$                  ;BR IF NO
002674' 052765  000010  000002          BIS     #10,2(R5)           ;SET BIT3 IN ENDPASS FLAG TO
                                                                    ;SHOW THAT WE GOT 7 XMIT DONES
002702' 026765  175324  000002          CMP     FLAGB,2(R5)         ;ALL DONE?
002710' 001003                          BNE     5$                  ;BR IF NO
002712' 042761  000100  000002          BIC     #100,2(R1)          ;CLEAR OIE IF YES
002720'                         5$:
002720' 104400  000000'                 EXIT$,BEGIN                         ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
002724' 142761  000207  000002  6$:     BICB    #207,2(R1)          ;CLEAR RDO
002732' 105265  000013                  INCB    13(R5)              ;HI BYTE IS REC DONE COUNT
002736' 122765  000007  000013          CMPB    #7,13(R5)           ;DO WE HAVE 7 REC DONES YET?
002744' 001012                          BNE     7$                  ;BR IF NO
002746' 052765  000020  000002          BIS     #20,2(R5)           ;SET BIT4 IN ENDPASS FLAG TO
                                                                    ;SHOW THAT WE GOT ALL 7 REC DONES
002754' 026765  175252  000002          CMP     FLAGB,2(R5)         ;ALL DONE?
002762' 001003                          BNE     7$                  ;BR IF NO
002764' 042761  000100  000002          BIC     #100,2(R1)          ;CLEAR OIE IF YES
002772'                         7$:
002772' 104400  000000'                 EXIT$,BEGIN                         ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```

```
                                ;LINK TABLE TO INTERRUPT SERVICE ROUTINES
                                ;----------------------------------------

002776'                         INTLNK:
002776' 004567  176776                  JSR     R5,IISR
003002' 004567  177466                  JSR     R5,OISR
003006' 000000                  CSR1:   .WORD                       ;DMC CSR FOR DEVICE 1
003010' 000000                  XX11:   .WORD                       ;END PASS FLAG FOR DEVICE 1
003012' 003270'                         RBUF1                       ;RECEIVE BUFFER POINTER FOR DEVICE 1
003014' 005124'                         BASE1                       ;BASE ADDRESS FOR DEVICE 1
003016' 000000                  XX12:   .WORD                       ;RECEIVE BUFFER OFFSET FOR DEVICE 1
003020' 000000                  XX13:   .WORD                       ;REC/XMIT COUNTERS
003022' 000000                          .WORD                       ;THESE NEXT 8 BYTES ARE FOR
003024' 000000                          .WORD                       ;THE DDCMP ERROR COUNTS
003026' 000000                          .WORD                       ;IN THE BASE TABLE TO BE
003030' 000000                          .WORD                       ;SAVED FOR COMPARISION.

003032' 004567  176742                  JSR     R5,IISR
003036' 004567  177432                  JSR     R5,OISR
003042' 000000                  CSR2:   .WORD                       ;DMC CSR FOR DEVICE 2
003044' 000000                  XX21:   .WORD                       ;END PASS FLAG FOR DEVICE 2
003046' 003306'                         RBUF2                       ;RECEIVE BUFFER POINTER FOR DEVICE 2
003050' 005524'                         BASE2                       ;BASE ADDRESS FOR DEVICE 2
003052' 000000                  XX22:   .WORD                       ;RECEIVE BUFFER OFFSET FOR DEVICE 2
003054' 000000                  XX23:   .WORD                       ;REC/XMIT COUNTERS
003056' 000000                          .WORD                       ;THESE NEXT 8 BYTES ARE FOR
003060' 000000                          .WORD                       ;THE DDCMP ERROR COUNTS
003062' 000000                          .WORD                       ;IN THE BASE TABLE TO BE
003064' 000000                          .WORD                       ;SAVED FOR COMPARISION.
```

```
                                    ;BUFFERS & QUEUES
                                    ;----------------

003066' 001001  004004  020020     XBUF:   .ASCII  <001><002><004><010><020><040><100><200><377><376>
003074' 100100  177377
003100' 175775  167767  137737             .ASCII  <375><373><367><357><337><277><177><037><076><174>
003106' 017577  076076
003112' 170370  160001  140003             .ASCII  <370><360><001><340><003><300><007>/ABCDEFGHIJKLM/
003120' 040407  041502  042504
003126' 043506  044510  045512
003134' 046514
003136' 047516  050520  051522             .ASCIZ  /NOPQRSTUVWXYZ01234567890/
003144' 052524  053526  054530
003152' 030132  031061  032063
003160' 033065  034067  030071
003166'    000
        003170'                     .EVEN

003170' 000010                      PIRINQ: .BLKW 10
003210' 000010                      PIROUTQ:.BLKW 10
003230' 000010                      REGQ:   .BLKW 10
003250' 000000                      INQIN:  0
003252' 000000                      INQOUT: 0
003254' 000000                      OUTQIN: 0
003256' 000000                      OUTQOUT:0
003260' 000000                      REGQI:  0
003262' 000000                      REGQO:  0

                                            ;TABLE OF RECEIVE BUFFER POINTERS

003264' 003270'                     BUFTAB: RBUF1                   ;BUFFER POINTER FOR DEVICE 1
003266' 003306'                             RBUF2                   ;BUFFER POINTER FOR DEVICE 2

                                            ;TABLE OF RECEIVE BUFFERS

003270'                             RBUF1:                          ;RECEIVE BUFFERS FOR DEVICE 1
003270' 003324'                             RBUF11
003272' 003424'                             RBUF12
003274' 003524'                             RBUF13
003276' 003624'                             RBUF14
003300' 003724'                             RBUF15
003302' 004024'                             RBUF16
003304' 004124'                             RBUF17
003306'                             RBUF2:                          ;RECEIVE BUFFERS FOR DEVICE 2
003306' 004224'                             RBUF21
003310' 004324'                             RBUF22
003312' 004424'                             RBUF23
003314' 004524'                             RBUF24
003316' 004624'                             RBUF25
003320' 004724'                             RBUF26
003322' 005024'                             RBUF27



                                            ;RECEIVE BUFFERS FOR DEVICE 1

```

```
003324' 000100                      RBUF11: .BLKB   100     ;RECEIVE BUFFER 11
003424' 000100                      RBUF12: .BLKB   100     ;RECEIVE BUFFER 12
003524' 000100                      RBUF13: .BLKB   100     ;RECEIVE BUFFER 13
003624' 000100                      RBUF14: .BLKB   100     ;RECEIVE BUFFER 14
003724' 000100                      RBUF15: .BLKB   100     ;RECEIVE BUFFER 15
004024' 000100                      RBUF16: .BLKB   100     ;RECEIVE BUFFER 16
004124' 000100                      RBUF17: .BLKB   100     ;RECEIVE BUFFER 17

                                            ;RECEIVE BUFFERS FOR DEVICE 2

004224' 000100                      RBUF21: .BLKB   100     ;RECEIVE BUFFER 21
004324' 000100                      RBUF22: .BLKB   100     ;RECEIVE BUFFER 22
004424' 000100                      RBUF23: .BLKB   100     ;RECEIVE BUFFER 23
004524' 000100                      RBUF24: .BLKB   100     ;RECEIVE BUFFER 24
004624' 000100                      RBUF25: .BLKB   100     ;RECEIVE BUFFER 25
004724' 000100                      RBUF26: .BLKB   100     ;RECEIVE BUFFER 26
005024' 000100                      RBUF27: .BLKB   100     ;RECEIVE BUFFER 27



005124' 000400                      BASE1:  .BLKB   256.    ;BASE TABLE FOR DEVICE 1

005524' 000400                      BASE2:  .BLKB   256.    ;BASE TABLE FOR DEVICE 2


        000001                      .END
```

```
DMCB DEC/X11 SYSTEM EXERCISER MODULE    MACY11 30A(1052)  12-OCT-78  16:29  PAGE 23
XDMCB0.P11    12-OCT-78 12:02                CROSS REFERENCE TABLE -- USER SYMBOLS                     SFQ 0021

ACSR      000102R         247#    442*    478*    639*
ADDR      000006R         213#    297     333
ADDR22=   001000          260#
ASB       000106R         251#    402*
ASTAT     000104R         249#    443*    479*    640*
AWAS      000110R         252#    403*
BASE1     005124R         293     680     777#
BASE2     005524R         693     779#
BEGIN     000000R         210#    289     347     348     390     405     424     446     450     451     475     497     546
                          568     572     585     590     601     606     614     628     643     646     659     670
BIT0  =   000001          260#    353     636
BIT1  =   000002          260#
BIT10 =   002000          260#
BIT11 =   004000          260#
BIT12 =   010000          260#
BIT13 =   020000          260#
BIT14 =   040000          260#    342
BIT15 =   100000          260#
BIT2  =   000004          260#    647
BIT3  =   000010          260#
BIT4  =   000020          260#
BIT5  =   000040          260#
BIT6  =   000100          260#
BIT7  =   000200          260#
BIT8  =   000400          260#
BIT9  =   001000          260#
BREAK$=   104407          260#    347     348     450     451
BR1       000012R         215#    316     320
BR2       000013R         216#
BTOD$ =   104421          260#
BUFTAB    003264R         382     731#
CDATA$=   104412          260#
CONFIG    000056R         235#
CSRA      000100R         245#    399*    440*    477*    638*
CSR1      003006R         463     677#
CSR2      003042R         468     690#
DATCK$=   104411          260#
DATER$=   104404          260#    405
DEV       001564R         473*    474*    492#    495
DLY1      000224R         263#    370*    455*    480*    507
DLY2      000226R         264#    371*    452*    481*    508
DROP      000320R         285     288#    296
DROP1     001566R         475     494#
DVID1     000014R         217#    284     286
EA        000250R         273#    563     578     594     615*    616*    617*    618*
EABITS    002442R         561     576     592     613#
ENDCLR    002430R         564     570     580     596     608#
ENDIT$=   104413          260#    390
END$  =   104410          260#    289
ERRTYP    000106R         250#    444*    485*    641*
ESAV1     001602R         426     436     482*    502#    509     514
ESAV2     001604R         483*    503#    510     515
ESAV3     001606R         484*    504#    511     516
ESAV4     001610R         438     505#    517
ETABLE    001612R         487     507#
EXIT$ =   104400          260#    568     572     585     590     601     606     646     659     670




DMCB DEC/X11 SYSTEM EXERCISER MODULE    MACY11 30A(1052)  12-OCT-78  16:29  PAGE 24
XDMCB0.P11    12-OCT-78 12:02                CROSS REFERENCE TABLE -- USER SYMBOLS                     SFQ 0022

FIRST     000234R         267#    290*    324     340     358     389*
FLAGB     000232R         266#    300     301     372     375     655     666
FTABLE    001626R         446     514#
GETPA$=   104415          260#    614
GWBUF$=   104414          260#
HRDCNT    000044R         230#
HRDER$=   104405          260#    487     643
HRDPAS    000050R         232#
ICONT     000036R         227#
ICOUNT    000040R         228#
IDNUM     000122R         257#
IISR      002000R         539#    675     688
INIT      000030R         224#
INQIN     003250R         302*    539*    540*    541     543*    722#
INQOUT    003252R         303*    548     549*    550     552*    723#
INTLNK    002776R         299     380     674#
INTR      000120R         256#    281*
LOOP      000330R         291#    392     470
MAP22$=   104416          260#
MASK      000236R         268#    369*    374*    377*    378     393*    410*    458
MODNAM    000000R         211#
MODSP     000224R         225     258#
MSGN$ =   104403          260#    424     475
MSGS$ =   104402          260#
MSG$  =   104401          260#
NULL  =   000000          260#    643
OISR      002474R         621#    676     689
OPEN  =   000000          212     218     219     220     221     238     239     240     241     242     243     244     245
                          247     249     251     252     254     255     256     260#
OTOA$ =   104420          260#
OUTQIN    003254R         304*    621*    622*    623     625*    724#
OUTQOU    003256R         305*    630     631*    632     634*    725#
PA        000246R         272#    562     577     593
PASCNT    000034R         226#
PIRINQ    003170R         302     303     541     543     550     552     719#
PIROUT    003210R         304     305     623     625     632     634     720#
PIRQ$ =   000004          260#    546     628
POPSP =   005726          260#
POPSP2=   022626          260#
PRTY  =   000000          260#
PRTY0 =   000000          260#
PRTY1 =   000040          260#
PRTY2 =   000100          260#
PRTY3 =   000140          260#
PRTY4 =   000200          260#
PRTY5 =   000240          215     216     260#
PRTY6 =   000300          260#
PRTY7 =   000340          260#
PS    =   177776          260#
PSW   =   177776          260#
PUSH  =   005746          260#
PUSH2 =   024646          260#
RAND$ =   104417          260#
RANNUM    000054R         234#
RBUF1     003270R         679     731     736#
RBUF11    003324R         291     737     757#
```

```
DMCB DEC/X11 SYSTEM EXERCISER MODULE    MACY11 30A(1052)  12-OCT-78  16:29  PAGE 25
XDMCB0.P11    12-OCT-78 12:02           CROSS REFERENCE TABLE -- USER SYMBOLS                                               SEQ 0023

RBUF12  003424R           738     758#
RBUF13  003524R           739     759#
RBUF14  003624R           740     760#
RBUF15  003724R           741     761#
RBUF16  004024R           742     762#
RBUF17  004124R           743     763#
RBUF2   003306R           692     732     744#
RBUF21  004224R           745     767#
RBUF22  004324R           746     768#
RBUF23  004424R           747     769#
RBUF24  004524R           748     770#
RBUF25  004624R           749     771#
RBUF26  004724R           750     772#
RBUF27  005024R           751     773#
RCOUNT  000240R           269#    579
REC     002206R           557     573#
REGQ    003230R           306     307     721#
REGQI   003260R           306*    726#
REGQO   003262R           307*    727#
RESTRT  000324R           253     287     290#
RES1    000056R           236#
RES2    000060R           237#
RSTRT   000112R           253#
SAR0    000252R           274#    345*    349
SAR1    000254R           275#    346*    350
SBADR   000102R           246#    400*
SCAN    000746R           337     369#
SELECT  000230R           265#    286*    295     334     381     459*
SETUP2  000600R           310     333#
SOFCNT  000042R           229#
SOFER$= 104406            260#    446
SOFPAS  000046R           231#
SOFT    001576R           424     499#
SOFT1   001717R           499     529#
SPOINT  000032R           225#
SPSIZ = 000040              1#    258
SR1     000016R           218#    353
SR2     000020R           219#
SR3     000022R           220#
SR4     000024R           221#
START   000256R           224     281#
STAT    000026R           223#
SVR0    000062R           238#
SVR1    000064R           239#
SVR2    000066R           240#
SVR3    000070R           241#
SVR4    000072R           242#
SVR5    000074R           243#
SVR6    000076R           244#
SYSCNT  000052R           233#
TCOUNT  000242R           270#    595
TERM  = 000000            277#    408
TRPDFD= 000022            260#
VA      000244R           271#    560*    575*    591*    614
VECTOR  000010R           214#    298
WASADR  000104R           248#    401*
```

```
DMCB DEC/X11 SYSTEM EXERCISER MODULE    MACY11 30A(1052)  12-OCT-78  16:29  PAGE 26
XDMCB0.P11    12-OCT-78 12:02           CROSS REFERENCE TABLE -- USER SYMBOLS                                               SEQ 0024

WDFR    000116R           255#    283*
WDTO    000114R           254#    282*
XBUF    003066R           396     591     704#
XDROP1  001640R           494     520#
XDROP2  001661R           496     523#
XERR    001444R           462     467     472#
XFLAG   000005R           212#
XMIT    002324R           555     591#
XX11    003010R           300*    372     678#
XX12    003016R           681#
XX13    003020R           682#
XX21    003044R           301*    375     691#
XX22    003052R           694#
XX23    003054R           695#
.     = 006124R           453     456     717#    719#    720#    721#    757#    758#    759#    760#    761#    762#    763#
                          767#    768#    769#    770#    771#    772#    773#    777#    779#

. ABS.  000000     000
        006124     001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XDMCB0,XDMCB0/SOL/CRF:SYM=DDXCOM,XDMCB0
 RUN-TIME: 2 3 .4 SECONDS
 RUN-TIME RATIO: 14/6=2.2
 CORE USED:  11K  (21 PAGES)
```

DIAGNOSTIC ENGINEERING

digital **DECO □ DEPO ☒ SUBMISSION □** □ NEW

FOR RELEASE ENG. USE □ CHANGE □ DELETE

PRODUCT IDENTIFICATION

	LIBRARY	PRODUCT NUMBER	REV	PATCH	ECO TALLY	PRODUCT DATE DD	MMM	YY	STATUS	DISTRIBUTION		1ST COPY-RIGHT YEAR	LAST COPY-RIGHT YEAR
MD	ZZ	CXDMC	B	1	Ø1	24	APR	79	□ OBSOLETE	X G	□ R	1976	1979

TITLE CXDMCB1 DMC-11 MODULE

AUTHOR D. BUTENHOF MAINTAINING GROUP DEC/X11 SUPT GRP MAINTAINER D. BUTENHOF SUBMITTING ENGINEER D. BUTENHO

PRODUCT COMPONENTS

CK	DESCRIPTION	PRODUCT NO.	REV	CK	DESCRIPTION	PRODUCT NO.	REV
	DOCUMENT				INDEX		
	LISTING				SOURCE MEDIA		
	OBJECT MEDIA				TEST MEDIA		
		AF-E953B-M1					

PRODUCTS OBSOLETED (other than previous version)

	LIBRARY	PRODUCT NUMBER	REV		LIBRARY	PRODUCT NUMBER	REV		LIBRARY	PRODUCT NUMBER	REV
MD				M D				M D			

PRODUCT CHARACTERISTICS

PROCESSORS PRODUCT OPERATES WITH (Enter all applicable 2-digit codes representing the Processor the product operates with. See separate instructions.)

OPERATIONAL CODES (Enter all applicable 2-digit codes that describe the product. See separate instructions.)

ACT/APT/XXDP INFORMATION FIELD

EXT

ACT SEQ NUMBER

ACT/XXDP COMPATIBLE? □ Y □ N

APT COMPATIBLE? □ Y □ N

1ST PASS RUN TIME SECONDS

SUBSEQUENT PASS RUN TIME SECONDS

DECO/DEPO INFORMATION

PROBLEM REPORTS CLOSED:

DEVICE AFFECTED

MULTIMEDIA AFFECTED? ☒ YES □ NO

KIT NUMBERS	ZJ13Ø-RB	
	ZJ129-RZ,	FR

PROBLEM:

THE "DDCMP ERROR COUNTERS ARE NON-ZERO" SOFT ERROR IS MISLEADING SINCE IT REFERS TO A RECOVERABLE AND SOFTWARE-TRANSPARENT SITUATION. NEVERTHELESS, IF 4Ø SUCH OCCUR , THE MODULE WILL BE DROPPED BY THE MONITOR.

SOLUTION:

THE SOFT ERROR CALL AND MESSAGE ARE DELETED FROM THE MODULE BY THIS PATCH, AND THE HARD ERROR CALL IS EXTENDED FOR CLARIFICATION OF THE SITUATION, IF THE DMC-11 SHOULD BE UNABLE TO RECOVER.

DEPO PATCH AREA

CHANGE LOC	FROM	TO	CHANGE LOC	FROM	TO
1226	1Ø44Ø3	4Ø2	1726	51122	51Ø4Ø
1334	1Ø44Ø6	4Ø2	173Ø	51117	521Ø5
1716	224ØØ	2ØØ72	1732	2644Ø	44522
172Ø	47523	53117	1734	42Ø4Ø	515Ø5
1722	521Ø6	511Ø5	1736	415Ø4	Ø
1724	4244Ø	3344Ø			

SUBMITTING ENGINEER	MANUFACTURING ENGINEER J. E. CASELLA	SUPPORT ENGINEER	CHARGE DECO/DEPO TO DISCRETE PROJECT NUMBER
DATE: 24 APR 79	DATE: 10-MAY-79	DATE:	Ø98-Ø546Ø
MAINTAINER	FIELD SERVICE	WAIVERING MANAGER	COORDINATION NO. 313Ø
	DATE:	DATE:	