IDENTIFICATION

PRODUCT CODE: AC-F858B-MC

PRODUCT NAME: CXDMRB0 DMR-11 MODULE

PRODUCT DATE: JUNE 1980

MAINTAINER: DEC/X11 SUPPORT GROUP

DEC/X11-1 SYSTEM EXERCISER MACRO DEFINITION MODULE

1. ABSTRACT

DMR IS AN IOMOD THAT EXERCISES UP TO AND INCLUDING TWO CONSECUTIVELY ADDRESSED AND CONSECUTIVELY VECTORED DMR11 SYNCHRONOUS INTERFACES. THIS MODULE EXERCISE MAY BE RUN WITH LINE UNIT LOOPBACK (TTL LOOPBACK). ADDITIONALLY IF TWO DMRS ARE TO BE EXERCISED, THE OPERATOR MAY SELECT TO PERFORM AN ACTUAL LINK EXERCISE WITH ACTUAL CABLE CONNECTION.
A SET DATA PATTERN IS USED FOR DATA TRANSMISSION. THE RECEIVER AND TRANSMITTER ISR ARE ALWAYS AT PRIORITY FIVE (BR1,BR2). DATA CHECKING IS PERFORMED AT LEVEL 0. THIS CHECKING IS DONE OUTSIDE THE ISRS AND ONLY ON THE LAST BUFFER RECEIVED BECAUSE A COMMON IS USED FOR ALL RECEIVED BUFFERS.

CXDMRB IS A REVISION THAT SOLVES 4 PROBLEMS IN REV A (CXDMRA) THESE PROBLEMS WERE
A. A PASS IN INTERNAL LOOPBACK TOOK MORE THAN 1 MINUTE
B. AN ILLEGAL INSTRUCTION FOR AN 11/20 (SOB) WAS USED.
C. WHEN 2 DMRS WERE EXERCISED IN DMR MODE, A FLAG CONTENTION COULD OCCUR.
D. A HUNG CONDITION COULD OCCUR UNDER CERTAIN CONDITIONS IF THE MODULE WERE RELOCATED BETWEEN ITERATIONS.

2. REQUIREMENTS

HARDWARE:
1. M8207-RA (MICROPROCESSOR)
2. M8203 (LINE UNIT)

STORAGE:: DMR REQUIRES:
1. DECIMAL WORDS: 2190.
2. OCTAL WORDS: 4216
3. OCTAL BYTES: 10434

3. PASS DEFINITION

ONE PASS OF THE DMR MODULE CONSISTS OF TRANSMITTING AND RECEIVING ALL BUFFERS 25 ITERATIONS FOR EACH SELECTED DEVICE. THE BUFFER NUMBER IS 7 WHEN IN DMC MODE AND 64. WHEN IN DMR MODE. BUFFER SIZE IS VARIED FROM 1 TO 512. BYTES BY THE PROGRAM.

4. EXECUTION TIME

RUNNING ALONE ON AN 11/40, ONE PASS TAKES APPROXIMATELY ONE MINUTE IN INTERNAL LOOPBACK

5. CONFIGURATION PARAMETERS.

DEFAULT PARAMETERS:
ADDR: 1, VECTOR: 1, BR1: 5, BR2: 5, DVID1: 1, SR1:0
DMR WILL RUN UP TO TWO CONSECUTIVELY ADDRESSED AND CONSECUTIVELY VECTORED DMR11'S.

```
SR1:
        BIT 0 = 1 - TURN AROUND CONNECTOR
              = 0 - LINE UNIT LOOPBACK (TTL LOOPBACK)

        BIT 1 = 1 - DMR MODE
              = 0 - DMC MODE

        BIT 2 = 1 - LINK MODE (BIT 0 IGNORED)
              = 0 - NON-LINK

SR4:            (VALID ONLY IN DMR MODE)
        BIT 0 = 1 - MODEM WRITE
        BIT 1 = 1 - ENABLE EXTENDED ERROR
        BIT 2 = 1 - DISABLE EXTENDED ERROR
        BIT 3 = 1 - DESELECT DMC LINE
        BIT 4 = 1 - REQUEST BASE TABLE UPDATE
        BIT 5 = 1 - SET REP/SEL TIMER
        BIT 6 = 1 - SET THRESHOLD VALUES
        BIT 7 = 1 - READ M8207 RAM

                    **********************
                    CAUTION - UNDERSTAND THE EFFECT OF THIS
                              COMMAND BEFORE USING IT.
        BIT 8 = 1 - WRITE AX3-15 INTERFACE
                    **********************

        BIT 9 = 1 - NOP1
        BIT 10 = 1 - READ MODEM (NOP)

IF ANY OF THE FOLLOWING DMR COMMANDS ARE USED COMMAND VALUES ARE
GIVEN AT LOCATION 300
```

6. DEVICE/OPTION SETUP

CONFIGURE MODULE WITH CORRECT PARAMETERS.

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

THE INTERRELATION OF CERTAIN PARAMETERS SHOULD BE UNDERSTOOD.

1. THERE MUST BE 2 DEVICES FOR A LINK EXERCISE
2. IN A LINK EXERCISE BIT 0 OF SR1 (TURN AROUND) IS IGNORED (AN ACTUAL CONNECTION IS ASSUMED).
3. DMR COMMANDS SPECIFIED IN SR4 CAN ONLY BE PERFORMED WHEN THE DMR11 IS IN DMR MODE.
4. VALUES USED FOR CERTAIN DMR COMMANDS ARE LOCATED AT LOCATION 300. BEFORE CHANGING THOSE VALUES REALIZE THE EFFECT IT MAY HAVE ON THE EXERCISE. NOTE: THAT TO ACTUALLY USE THE DMR COMMAND TO WRITE THE AX3-15 INTERFACE, UNDERSTAND THE EFFECT OF THE AX3 VARIABLE BEFORE USING THE COMMAND.

7. MODULE OPERATION

1. LOAD SOFTWARE POINTERS IN LINK TABLE.
2. LOAD VECTORS AND PRIORITIES IN TABLE
3. ENABLE SELECTED DEVICES.
4. IF FIRST PASS, PERFORM A BASE IN AND CONTROL IN. ALSO IF IN DMR MODE, PERFORM ANY REQUESTED DMR COMMANDS.
5. COMMENCE BUFFER TRANSACTIONS.
6. SCAN FOR ALL DEVICES TO FINISH BUFFER TRANSACTIONS.
7. IF NOT DONE GO TO 5.
   IF HUNG REPORT SO AND DROP HUNG DEVICE.
8. IF NOT LAST ITERATION, GO TO 10
9. AFTER LAST ITERATION, HALT THE DMR AND WAIT FOR THE HALT TO BE COMPLETED.
10. CHECK DATA FOR ALL DEVICES SELECTED.
11. DECREMENT ITERATION COUNT
12. IF NOT = 0 GO TO 1
13. SIGNAL ENDPASS.

IISR: INPUT INTERRUPT SERVICE ROUTINE.
I1. GET INTERRUPTING DMR CSR.
I2. IF BASE IN WAS REQUESTED, LOAD BASE ADDRESS. IF NOT FIRST ITERATION OF THE PASS SET THE BASE IN RESUME BIT.
I3. IF IN DMR MODE AND IF DMR COMMANDS REQUESTED, PERFORM ANY AND ALL DMR COMMANDS.
I4. IF RECEIVE BA/CC WAS REQUESTED, LOAD REC BA/CC.
I5. IF XMIT BA/CC WAS REQUESTED, LOAD XMIT BA/CC.
I6. IF LAST RCV OR XMIT BA/CC IN SET THE APPROPRIATE BIT IN THE END OF PASS FLAG.
I7. RTI

OISR: OUTPUT INTERRUPT SERVICE ROUTINE.
O1. GET INTERRUPTING DMR CSR
O2. IF ERROR, REPORT IT AND EXIT.
O3. IF XMIT DONE OR REC DONE, ISSUE A HALT (PROCEDURE ERROR).
O4. AFTER RECEIVING THE CONTROL OUT FROM THE HALT (PROCEDURE ERROR), SET THE BIT FOR THE ENDPASS FLAG
O5. 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: 4660
XX21: 4722

ONLY BITS 0 THRU 4 ARE USED AND ARE DEFINED AS FOLLOWS:
BIT0 = 1 THE BASE ADDRESS WAS LOADED.
BIT1 = 1 ALL BA/CC IN RECEIVE BUFFERS LOADED
BIT2 = 1 ALL BA/CC IN TRANSMIT BUFFERS LOADED.
BIT3 = 1 ALL BA/CC OUT TRANSMITS WERE RECEIVED.
BIT4 = 1 ALL BA/CC OUT RECEIVES WERE RECEIVED.
BIT5 = 1 HALT DONE (OR END OF ITERATION)

A CORRECT END PASS FLAG = 77, WHEN THE ENDPASS FLAGS = 77 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 77 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 DMR'S PROTOCOL CHECKERS DETECT AN ERROR IN THE TRANSMISSION OF A MESSAGE, IT WILL RETRANSMIT THE ENTIRE MESSAGE, UPDATING AN ERROR COUNTER IN ITS 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. NOTE: THIS THRESHOLD OF 7 MAY BE CHANGED IN DMR MODE BY CHANGING THE NAKS THRESHOLD. FOR DEC/X11 PURPOSES, HOWEVER, THE DMR 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, DMR WILL DECLARE A SOFT ERROR.

THE SOFT ERROR MESSAGE MAY INDICATE AN INTERMITTANT DEVICE 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 DMR-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 DMR-11'S NPR'S ARE NOT BEING HONORED QUICKLY ENOUGH TO PREVENT BIT-DROPPING.

TO VERIFY WHETHER THIS IS THE CONDITION, RUN A SINGLE DMR MODULE, WITH A SINGLE DMR-11 DEVICE SELECTED. THE SOFT ERROR MESSAGE SHOULD NOT OCCUR UNDER THESE CONDITIONS. IF IT DOES OCCUR, THE PROBLEM IS PROBABLY IN DMR-11 HARDWARE OR A

CABLE FAULT.

```
      1 000000                                IOMOD <DMRB >,1,1,5,5,0,25.,127
        000000                                MODULE  140000,DMRB ,1,1,5,5,0,25.,127
                                              .TITLE  DMRB DEC/X11 SYSTEM EXERCISER MODULE
                                      ;       DDXCOM  VERSION  6       23-MAY-78
                                              .LIST   BIN
                                      ;**********************************************************************
        000000                        BEGIN:
        000000     104     115     122  MODNAM: .ASCII  /DMRB / ;MODULE NAME.
        000003     102     040
        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  000346'               INIT:   START                    ;MODULE START ADDR.
        000032  000224'               SPOINT: MODSP                    ;MODULE STACK POINTER.
        000034  000000                PASCNT: 0                        ;PASS COUNTER.
        000036  000031                ICONT:  25.                      ;# OF ITERATIONS PER PASS=25.
        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  000640'               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
                000040                        .REPT   SPSIZ            ;MODULE STACK STARTS HERE.
        000224                        MODSP:
```



```
                                      ;**********************************************************************
      2
                                      ;VARIABLES FOR DMR11

        000224  000000                DLY1:   0                        ;DELAY COUNTER 1 USED IN DELAY LOOP.
        000226  000000                DLY2:   0                        ;DELAY COUNTER 2 USED IN DELAY LOOP.
        000230  000000                SELECT: 0                        ;LOCATION TO SAVE THE # OF ACTIVE DMRS.
        000232  000037                FLAGA:  37                       ;MASK FOR ALL BUT HALT COMPLETE
        000234  000077                FLAGB:  77                       ;MASK FOR END OF PASS FLAGS (XX11 AND XX21).
                                                                       ;A BIT IN THE FLAG IS SET WHEN THE DMR IS
                                                                       ;COMPLETED A COMMAND(S) I.E. ALL BA/CC IN
                                                                       ;RECEIVE ACTIONS.  37 = END OF PASS.
        000236  000000                FIRST:  0                        ;FLAG USED TO MARK FIRST ITERATION OF A PASS.
        000240  000000                LAST:   0                        ;FLAG TO MARK LAST ITERATION.
        000242  000000                COUNT:  0                        ;WORD USED TO DETERMINE THE LAST ITERATION.
        000244  000000                MASK:   0                        ;MASK USED IN "SCAN" LOOP TO MARK THE
                                                                       ;NUMBER OF DMRS UNDER TEST.
        000246  000000                BUFNUM: 0                        ;NUMBER OF RECV AND TRANSMIT BUFFERS
        000250  000000                DMREXT: 0                        ;DMR EXTENDED CODE FLAG
                                                                       ;       BIT 0 = MODEM WRITE
                                                                       ;       BIT 1 = ENABLE EXT ERROR
                                                                       ;       BIT 2 = DISABLE EXT ERROR
                                                                       ;       BIT 3 = DESELECT DMC LINE
                                                                       ;       BIT 4 = REQUEST BASE T. UPDATE
                                                                       ;       BIT 5 = SET REP/SELECT TIMER
                                                                       ;       BIT 6 = SET THRESHOLD VALUES.
                                                                       ;       BIT 7 = READ M8207 RAM
                                                                       ;       BIT 8 = WRITE AX3-15  ****CAUTION****
                                                                       ;       BIT 9 = NOP
                                                                       ;       BIT 10= READ MODEM STATUS
        000252  000000                MODE:   0                        ;MODE FLAG
                                                                       ;MODE =  0 FOR DMC MODE
                                                                       ;     = -1 FOR DMR MODE

        000254  000000                LINK:   0                        ;FLAG FOR LINK TEST.
        000256  000000                LULOOP: 0                        ;FLAG USED TO MARK WHEN LINE UNIT
                                                                       ;LOOPBACK IS DESIRED
        000260  000000                VA:     0                        ;VIRTUAL ADDRESS
        000262  000000                PA:     0                        ;PHYSICAL ADDRESS
        000264  000000                EA:     0                        ;EXTENDED ADDRESS
        000266  000000                SAR0:   0                        ;SAVE R0
        000270  000000                SAR1:   0                        ;SAVE R1
                                              .BLKW   3
                                      ;
                                      ; PATCH AREA TO CHANGE ANY OF THE FOLLOWING COMMAND VALUES
                                      ;
                                                                       ;MODEM STATUS BITS IN BSEL6
                                                                       ;BIT0 = RTS HOLD
                                                                       ;BIT1 = SELECT STANDBY
                                                                       ;BIT2 = MAINTENANCE 1 (REMOTE MODEM LOOPBACK)
                                                                       ;BIT3 = MAINTENANCE 2 (LOCAL MODEM LOOPBACK)
                                                                       ;BIT4 = HALF DUPLEX
                                                                       ;BIT5 = SELECT FREQUENCY
                                                                       ;BIT6 = DTR
                                                                       ;BIT7 = POLL
        000300  000000                MODEM1: 0                        ;BITS TO CLEAR IN MODEM REGISTER
        000302  000000                MODEM2: 0                        ;BITS TO SET IN MODEM REGISTER
```

```
000304  000036                  TIME:   30.                     ;REP/SELECT TIMER VALUE IN SEL6
                                                                ;IN 100 MSEC INCREMENTS (I.E.
                                                                ;VALUE OF 5 = 500 MSECS)
                                                                ;THRESHOLD VALUES:
000306  003006                  THRSH1: 3006                    ;LO BYTE (BSEL4) = NAKS REC.
                                                                ;HI BYTE (BSEL5) = NAKS XMIT.
000310  003006                  THRSH2: 3006                    ;LO BYTE (BSEL6) = REP/SEL SENT
                                                                ;HI BYTE (BSEL7) = NO BUFFER
000312  000000                  MRAM:   0                       ;MICRO RAM ADDRESS (LOADED IN SEL6)
                                                                ;CONTENTS RETURNED IN BSEL3

                                                                ;****** WARNING *********
                                                                ;THE CODE TO WRITE AX3-15
                                                                ;MAY CAUSE PROBLEMS IF YOU
                                                                ;ATTEMPT TO CHANGE TO AN INTERFACE
                                                                ;THAT YOU DON'T HAVE A TURNAROUND
                                                                ;OR IMPROPER CONFIGURATION.
                                                                ;**************************

000314  000133                  AX3:    133                     ;DEFAULT RS422
                                                                ;THE BIT CLEARED WILL DETERMINE THE
                                                                ;PATTERN WRITTEN INTO AX3-15. THIS
                                                                ;VALUE IS LOADED INTO BSEL7
                                                                ;I.E.
                                                                ;BIT3 (10)  AX3=323   INTEGRAL  - H3254
                                                                ;BIT4 (20)  AX3=313   V.35      - H3254
                                                                ;BIT6 (100) AX3=233   E1A (XYZ) - H3255
                                                                ;BIT7 (200) AX3=133   RS422     - H3255

                                        .BLKW   4.              ;

000326  000001  000077  000010  BUFFER: 1,63.,8.,128.,32.,257.,512.,17. ;BUFFER SIZE TABLE
000334  000200  000040  000401
000342  001000  000021
                                ;
                                ;
                                ;
                                ; ********    DMR COMMANDS        ***********

                000000          BACC1I  =0                      ;BA/CC IN TRANSMIT COMMAND
                000001          CNTIN   =1                      ;CONTROL IN COMMAND
                000002          FIN1    =2                      ;HALT COMMAND
                000003          BASE1   =3                      ;BASE IN COMMAND
                000004          BACCIR  =4                      ;BA/CC IN RECEIVE COMMAND
                000005          MW      =5                      ;MODEM WRITE COMMAND
                000006          EXER    =6                      ;ENABLE EXTENDED ERROR COMMAND
                000007          DXER    =7                      ;DISABLE EXT. ERROR COMMAND
                000010          DDMC    =10                     ;DESELECT DMC LINE MODE
                000011          UPDATE  =11                     ;REQUEST BASE TABLE UPDATE COMMADN
                000012          TIMER   =12                     ;SET REP/SELECT TIMER COMMAND
                000013          THRESH  =13                     ;SET THRESHOLD COMMANDS
                000014          RRAM    =14                     ;READ M8207 RAM
                000015          INTER   =15                     ;WRITE AX3-15 INTERFACE
                000016          NOP1    =16                     ;NOP
                000017          RMODEM  =17                     ;READ MODEM STATUS COMMAND (NOP).
                000522          DMR     =522                    ;DMR MODE PASSWORD
                001000          STOP    =1000                   ;HALT (OR PROCEDURE ERROR) BIT
```



```
                010000          RESUME  =10000                 ;RESUME BIT

                020040          OK      =20040                  ;MASK FOR NON ERROR CONTROL OUT IN DMR MODE:
                                                                ;BIT 5  = REPORT DMR START UP.
                                                                ;BIT 13 = REPORT BASE TABLE UPDATE.
                                                                ;DMR RUN AND BASE TABLE UPDATE
                001000          HALTC   =1000                   ;CONTROL OUT - BIT SET FOR HALT.
                000000          TERM    =0
```

```
                                ;       BEGIN THE TESTS FOR THE DMC11/DMR11

                                START:
000346  032767  000004  177442          BIT     #BIT2,SR1       ;IS THIS A LINK TEST
000354  001406                          BEQ     1$              ;IF NOT PROCEED
000356  012767  177777  177670          MOV     #-1,LINK        ;SET THE LINK TEST FLAG
000364  005067  177666                  CLR     LULOOP          ;CLEAR THE LINE UNIT LOOP FLAG.
000370  000402                          BR      2$
000372                          1$:
000372  005067  177656                  CLR     LINK            ;NO LINK TEST - CLEAR FLAG
000376                          2$:
000376  032767  000002  177412          BIT     #BIT1,SR1       ;WHAT MODE IS DESIRED?
000404  001412                          BEQ     3$              ;BR FOR DMC MODE
000406  012767  000100  177632          MOV     #64.,BUFNUM     ;# OF BUFFERS IN DMR MODE.
000414  012767  177777  177630          MOV     #-1,MODE        ;SET THE DMR MODE FLAG
000422  016767  177376  177620          MOV     SR4,DMREXT      ;SET DMR COMMAND FLAG.
000430  000407                          BR      4$
000432                          3$:
000432  012767  000007  177606          MOV     #7,BUFNUM       ;# OF BUFFERS IN DMC MODE
000440  005067  177606                  CLR     MODE            ;CLEAR FLAG = DMC MODE
000444  005067  177600                  CLR     DMREXT          ;CLEAR DMR COMMAND FLAG.

000450                          4$:
000450  005767  177600                  TST     LINK            ;IS THIS A LINK TEST?
000454  001010                          BNE     5$              ;IF YES - DON'T ALLOW SELECTION OF LU LOOP
000456  032767  000001  177332          BIT     #BIT0,SR1       ;TURN AROUND CONNECTOR
000464  001004                          BNE     5$              ;YES - BR
000466  012767  177777  177562          MOV     #-1,LULOOP      ;SET FLAG TO INDICATE LINE UNIT
                                                                ;LOOP NEEDED
000474  000402                          BR      6$
000476                          5$:
000476  005067  177554                  CLR     LULOOP          ;CLEAR FLAG - USING TURN AROUND
000502                          6$:
000502  016767  177540  177410          MOV     BUFNUM,INTR     ;INTERRUPTS/ITERATION
000510  012767  001000  177376          MOV     #512.,WDTO      ;MAX. WORDS TO MEM/ITERATION
000516  012767  001000  177372          MOV     #512.,WDFR      ;MAX. WORDS FROM MEM/ITERATION
000524  032767  177774  177262          BIT     #^C<3>,DVID1    ;DROP MODULE IF DEVICES OTHER
000532  001040                          BNE     DROP            ;THAN FIRST 4 ARE SELECTED
000534  016767  177254  177466          MOV     DVID1,SELECT    ;SELECT=ACTIVE DEVICES
000542  001434                          BEQ     DROP            ;NO DEVICES - DROP.
000544  005767  177504                  TST     LINK            ;IS THIS A LINK TEST?
000550  001433                          BEQ     RESTRT          ;IF NOT - START (SKIP DEVICE CHECK)
000552  022767  000003  177450          CMP     #3,SELECT       ;ARE THERE TWO DEVICES?
000560  001427                          BEQ     RESTRT          ;IF YES - PROCEED
                                ;-------------------------------------------------------
                                ; ERROR - MUST HAVE 2 DEVICES FOR A LINK TEST.
                                ;-------------------------------------------------------
000562  104401  000000' 002641'          MSG$,BEGIN,NEED2        ;ASCII MESSAGE CALL
000570  016767  177212  177302          MOV     ADDR,CSRA       ;SAVE CSR FOR DEVICE 1
000576  016767  177276  177276          MOV     CSRA,ACSR       ;CALCULATE CSR FOR DEVICE 2
000604  012767  000010  177270          MOV     #10,ACSR        ;SECOND DEVICE SHOULD BE AT THIS CSR.
000612  016767  177412  177264          MOV     SELECT,ASTAT    ;SAVE THE SELECT FLAG
000620  012767  000006  177260          MOV     #6,ERRTYP       ;DEVICE OFF LINE, NON EX OR NOT READY ERROR.
                                        ;*************************************************************
000626  104405  000000' 000000          HRDERS,BEGIN,NULL       ;MUST HAVE 2 DEVICES FOR A LINK TEST
                                        ;*************************************************************
```



```
000634                          DROP:
000634  104410  000000'              ENDS,BEGIN              ;NO DMR'S OR ILLEGAL DMR'S SELECTED
000640                          RESTRT:
000640  005067  177372                  CLR     FIRST           ;FLAG FOR 1ST PASS
000644  005067  177370                  CLR     LAST            ;FLAG FOR LAST PASS
000650  016767  177162  177364          MOV     ICONT,COUNT     ;# OF ITERATIONS
000656  005367  177360                  DEC     COUNT           ;COUNT IS 1 LESS
000662  001002                          BNE     LOOP            ;AS LONG AS IT ISN'T 0, OK
000664  005267  177352                  INC     COUNT           ;IF 0, CORRECT (ONLY WHEN ICONT=1)
000670                          LOOP:
000670  012700  006362'                MOV     #RCVBUF,R0      ;GET SET TO CLEAR BUFFERS
000674                          3$:
000674  005020                          CLR     (R0)+           ;CLEAR BUFFER
000676  022700  007362'                CMP     #BASE1,R0       ;END OF BUFFERS?
000702  001374                          BNE     3$              ;BR IF NO
000704  016700  177320                  MOV     SELECT,R0       ;R0=ACTIVE BITS
000710  001751                          BEQ     DROP            ;DROP MODULE IF NO DEVICES ARE SELECTED
000712  016701  177070                  MOV     ADDR,R1         ;R1=DEVICE CSR
000716  016702  177066                  MOV     VECTOR,R2       ;R2=VECTOR
000722  012703  004646'                MOV     #INTLNK,R3      ;R3=POINTER TO INTERRUPT LINKAGE
000726  016767  177302  003724          MOV     FLAGB,XX11      ;SET END PASS FLAG FOR DEVICE #1
000734  016767  177274  003760          MOV     FLAGB,XX21      ;SET END PASS FLAG FOR DEVICE #2
000742  012767  005752' 005402          MOV     #PIRING,INQIN   ;SET UP ALL QUEUES
000750  012767  005752' 005376          MOV     #PIRING,INQOUT
000756  012767  006152' 005372          MOV     #PIROUTQ,OUTQIN
000764  012767  006152' 005366          MOV     #PIROUTQ,OUTQOUT
000772                          4$:
000772  006200                          ASR     R0              ;ACTIVE?
000774  103410                          BCS     6$              ;BR IF ACTIVE
000776  001455                          BEQ     SETUP2          ;BR IF DONE
001000                          5$:
001000  062701  000010                  ADD     #10,R1          ;UPDATE CSR
001004  062702  000010                  ADD     #10,R2          ;UPDATE VECTOR
001010  062703  000042                  ADD     #42,R3          ;UPDATE LINK
001014  000766                          BR      4$              ;CONTINUE
001016                          6$:
001016  010312                          MOV     R3,(R2)         ;LOAD VECTOR
001020  116762  176766  000002          MOVB    BR1,2(R2)       ;LOAD INTERRUPT LEVEL
001026  010163  000010                  MOV     R1,10(R3)       ;LOAD CSR TO LINKAGE
001032  010362  000004                  MOV     R3,4(R2)        ;LOAD LINKAGE ADDRESS IN VECTOR
001036  062762  000004  000004          ADD     #4,4(R2)        ;ADJUST IT
001044  116762  176742  000006          MOVB    BR1,6(R2)       ;LOAD INTERRUPT LEVEL
001052  042763  177776  000012          BIC     #<^C1>,12(R3)   ;CLEAR ALL BUT BIT0 IN THE END OF PASS FLAG.
                                                                ;THIS BIT WILL ONLY BE SET ON THE FIRST
                                                                ;ITERATION AFTER A BASE IN.
001060  005063  000020                  CLR     20(R3)          ;CLEAR INPUT COUNT LOCATION
001064  005063  000022                  CLR     22(R3)          ;CLEAR OUTPUT COUNT LOCATION
001070  005767  177142                  TST     FIRST           ;FIRST PASS?
001074  001341                          BNE     5$              ;BR IF NOT
001076  005063  000012                  CLR     12(R3)          ;CLEAR ENTIRE END OF PASS FLAG ON FIRST ITER.
001102  005063  000024                  CLR     24(R3)          ;CLEAR AREA TO SAVE
001106  005063  000026                  CLR     26(R3)          ;BASE TABLE ERROR
001112  005063  000030                  CLR     30(R3)          ;COUNTS FOR COMPARISION
001116  005063  000032                  CLR     32(R3)          ;
001122  016763  177122  000034          MOV     DMREXT,34(R3)   ;INITIALIZE DMR EXTENDED COMMANDS
001130  000723                          BR      5$              ;CONTINUE
```

```
001132                          SETUP2:
001132  016701  176650                  MOV     ADDR,R1         ;R1=DEVICE CSR
001136  016700  177066                  MOV     SELECT,R0       ;R0=ACTIVE BITS
001142                          1$:
001142  006200                          ASR     R0              ;ACTIVE?
001144  103404                          BCS     3$              ;BR IF YES
001146  001523                          BEQ     SCAN            ;BR IF DONE
001150                          2$:
001150  062701  000010                  ADD     #10,R1          ;UPDATE CSR
001154  000772                          BR      1$              ;CONTINUE
001156                          3$:
001156  005767  177054                  TST     FIRST           ;FIRST PASS?
001162  001012                          BNE     4$              ;MASTER CLEAR FIRST TIME ONLY
001164  105061  000003                  CLRB    3(R1)           ;CLEAR BSEL3 - VALUE RETURNED AFTER
                                                                ;INIT, IF DEVICE IS A DMR.
001170  012711  040000                  MOV     #BIT14,(R1)     ;MASTER CLEAR
001174  000240                          NOP                     ;PATCH AREA FOR M8206
001176  000240                          NOP
001200  000240                          NOP
001202  000240                          NOP
001204  000240                          NOP
001206  000402                          BR      5$
001210                          4$:
001210  052711  100000                  BIS     #BIT15,(R1)     ;SET THE RUN BIT.
001214                          5$:
001214  005711                          TST     (R1)            ;RUN SET?
001216  100415                          BMI     6$              ;BR IF YES
001220  010067  177042                  MOV     R0,SAR0         ;SAVE R0
001224  010167  177040                  MOV     R1,SAR1         ;SAVE R1
001230  104407  000000'                  BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
001234  104407  000000'                  BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001240  016700  177022                  MOV     SAR0,R0         ;RESTORE R0
001244  016701  177020                  MOV     SAR1,R1         ;RESTORE R1
001250  000761                          BR      5$              ;WAIT FOR RUN
001252                          6$:
001252  005767  176760                  TST     FIRST           ;IS THIS THE FIRST PASS?
001256  001033                          BNE     15$             ;IF NOT, MODE SHOULD BE DETERMINED.
001260  005767  176766                  TST     MODE            ;MODE?
001264  001430                          BEQ     15$             ;BR IF DMC MODE.
001266  105761  000003                  TSTB    3(R1)           ;IS THIS A DMR?
001272  001025                          BNE     15$             ;IF DMR - OK (NOTE: NOTHING
                                                                ;RETURNED IN BSEL3 IF DEVICE IS DMC)

                                        ;-------------------------------------------------------
                                        ; ERROR - CAN'T BE IN DMR MODE UNLESS THE DEVICE IS ACTUALLY A DMR.
                                        ;         (NOTE: DMR CAN BE IN DMC MODE; HOWEVER DMC CAN ONLY BE IN
                                        ;         DMC MODE).
                                        ;-------------------------------------------------------
001274  010146                          MOV     R1,-(SP)        ;SAVE R1 - DURING THE CALL TO THE MSG
                                                                ;MACRO, INTERRUPTS MAY OCCUR IF ANOTHER
                                                                ;DEVICE HAS BEEN SETUP. IF THIS OCCURS
                                                                ;R1 MAY RETURN FROM THE INTERRUPT WITH
                                                                ;THE ADDRESS OF THE 1ST DEVICE.
001276  104401  000000' 002556'          MSG$,BEGIN,MIXED        ;ASCII MESSAGE CALL
001304  012601                          MOV     (SP)+,R1        ;RESTORE THE ADDRESS IN R1
001306  010167  176566                  MOV     R1,CSRA         ;SAVE CSR
001312  016167  000002  176562          MOV     2(R1),ACSR      ;SAVE CONTENTS OF SEL2
```



```
001320  016767  176726  176556          MOV     MODE,ASTAT      ;SAVE THE MODE FLAG
001326  012767  000006  176552          MOV     #6,ERRTYP       ;DEVICE NOT READY.
                                        ;*****************************************************************
001334  104405  000000' 000000          HRDER$,BEGIN,NULL       ;DMR MODE ILLEGAL WHEN DEVICE IS A DMC
                                        ;*****************************************************************
001342  104410  000000'                  END$,BEGIN              ;DROP IT
001346                          15$:
001346  005767  176704                  TST     LOLOOP          ;IS LINE UNIT REQUESTED?
001352  001403                          BEQ     16$             ;BR IF NO (CONNECTOR USED)
001354  052711  004000                  BIS     #4000,(R1)      ;OTHERWISE SET LU LOOP
001360  000402                          BR      17$             ;CONTINUE
001362                          16$:
001362  042711  004000                  BIC     #4000,(R1)      ;TURNAROUND OR LINK CONNECTOR
001366                          17$:
001366  052761  000100  000002          BIS     #100,2(R1)      ;SET IEO
001374  005767  176636                  TST     FIRST           ;IS THIS THE FIRST ITERATION
001400  001003                          BNE     18$             ;IF NOT - SKIP BASE IN
001402  052711  000143                  BIS     #143,(R1)       ;SET IEI,RQI,BASEI
001406  000660                          BR      2$              ;CONTINUE NEXT DEVICE
001410                          18$:
001410  052711  000144                  BIS     #144,(R1)       ;SET IEI,RQI,BA/CC IN
001414  000655                          BR      2$              ;CONTINUE NEXT DEVICE
```

```
      0                                        ;----------------------------------------
                                               ;SCAN ALL ENDPASS FLAGS UNTIL
                                               ;ALL ACTIVE DMR11S ARE FINISHED
                                               ;    (ENDPASS FLAG = 37)
                                               ;----------------------------------------
         001416                                SCAN:
         001416  012767  000003  176620                MOV     #3,MASK              ;SET BIT FOR ALL DEVICES
         001424  012767  000010  176572                MOV     #10,DLY1             ;DELAY COUNT
         001432  005067  176570                        CLR     DLY2                 ;DELAY
         001436                                1$:
         001436  026767  176572  003214                CMP     FLAGB,XX11           ;DEVICE 1 DONE?
         001444  001003                                BNE     2$                   ;BR IF NO
         001446  042767  000001  176570                BIC     #1,MASK              ;DEVICE 1 IS DONE SO CLEAR BIT0
         001454                                2$:
         001454  026767  176554  003240                CMP     FLAGB,XX21           ;DEVICE 2 DONE?
         001462  001003                                BNE     3$                   ;BR IF NO
         001464  042767  000002  176552                BIC     #2,MASK              ;DEVICE 2 IS DONE SO CLEAR BIT1
         001472                                3$:
         001472  005767  176546                        TST     MASK                 ;ARE ALL DEVICES FINISHED?
         001476  001440                                BEQ     CHECK                ;IF YES - CHECK DATA

         001500                                20$:
                                               ;------------------------------------------------------------
                                               ; DELAY LOOP
                                               ;------------------------------------------------------------
         001500  104407  000000'                       BREAK$,BEGIN                 ;TEMPORARY RETURN TO MONITOR....
         001504  104407  000000'                       BREAK$,BEGIN                 ;THEN CONTINUE AT NEXT INSTRUCTION.
         001510  005367  176512                        DEC     DLY2                 ;DEC DELAY COUNT
         001514  001402                                BEQ     25$                  ;
         001516  000167  177714                        JMP     1$                   ;BR IF NOT DONE
         001522                                25$:
         001522  005367  176476                        DEC     DLY1                 ;DEC DELAY COUNT
         001526  001402                                BEQ     30$                  ;
         001530  000167  177702                        JMP     1$                   ;BR IF NOT DONE
         001534                                30$:
         001534  016700  176504                        MOV     MASK,R0              ;PUT BITS OF HUNG DEVICES IN R0
         001540  040067  176464                        BIC     R0,SELECT            ;DROP ANY HUNG DEVICES
         001544  006000                                ROR     R0                   ;WAS DEVICE 1 HUNG?
         001546  103004                                BCC     35$                  ;BR IF NOT
         001550  004367  000446                        JSR     R3,XERR              ;TYPE ERROR MESSAGE
         001554  004656'                               CSR1                         ;POINTER TO DEVICE #1 CSR
         001556  000001                                1                            ;DEVICE NUMBER FOR TYPEOUT
         001560                                35$:
         001560  006000                                ROR     R0                   ;WAS DEVICE 2 HUNG?
         001562  103004                                BCC     40$                  ;BR IF NOT
         001564  004367  000432                        JSR     R3,XERR              ;TYPE ERROR MESSAGE
         001570  004720'                               CSR2                         ;POINTER TO DEVICE #2 CSR
         001572  000002                                2                            ;DEVICE NUMBER FOR TYPEOUT
         001574                                40$:
         001574  000167  177070                        JMP     LOOP                 ;RESTART MODULE

                                               ;------------------------------------------------------------
                                               ;CHECK EACH CHARACTER OF THE LAST RECEIVE BUFFER, AND THE BASE
                                               ;TABLE ERROR COUNTS.
                                               ;------------------------------------------------------------
         001600                                CHECK:
         001600  012701  004656'                       MOV     #INTLNK+10,R1              ;R1 IS POINTER TO CSR
```



```
         001604  016700  176420                        MOV     SELECT,R0            ;R0 CONTAINS BITS FOR ACTIVE DEVICES
         001610                                4$:
         001610  006200                                ASR     R0                   ;ACTIVE?
         001612  103462                                BCS     12$                  ;BR IF YES
         001614  001403                                BEQ     6$                   ;BR IF DONE
         001616                                5$:
         001616  062701  000042                        ADD     #42,R1               ;UPDATE R1 TO NEXT DEVICE CSR
         001622  000772                                BR      4$                   ;CONTINUE
         001624                                6$:
         001624  012767  177777  176404                MOV     #-1,FIRST            ;SET FIRST FLAG TO -1
         001632  005367  176404                        DEC     COUNT                ;DECREMENT LAST ITERATION COUNT
         001636  001003                                BNE     7$                   ;IF NOT 0, NOT LAST ITERATION.
         001640  012767  177777  176372                MOV     #-1,LAST             ;SET LAST PASS FLAG
         001646                                7$:

         001646  104413  000000'                       ENDIT$,BEGIN                 ;SIGNAL END OF ITERATION.
                                                                                    ;MONITOR SHALL TEST END OF PASS
                                                                                    ;THIS CODE WAS ADDED TO SEE IF RELOCATION
                                                                                    ;TOOK PLACE. IF IT DID, MAKE SURE THE DMR
                                                                                    ;IS RESET (BY USING THE FIRST FLAG).
         001652  032767  000001  176350                BIT     #BIT0,SELECT         ;WAS THERE A DEVICE 1?
         001660  001416                                BEQ     8$                   ;IF NOT CHECK BASE2 ADDRESS
         001662  012767  007362' 176370                MOV     #BASE1,VA            ;GET BASE TABLE 1 ADDRESS.
         001670  004767  002322                        JSR     PC,EABITS            ;GET THE PHYSICAL ADDRESS OF THE TABLE
         001674  026767  003004  176360                CMP     XX15,PA              ;IS THE ADDRESS THE SAME AS ON LAST BASE IN?
         001702  001022                                BNE     9$                   ;IF NOT, CLEAR THE FIRST FLAG
         001704  026767  002776  176352                CMP     XX16,EA              ;CHECK THE EXTENDED ADDRESS
         001712  001016                                BNE     9$                   ;IF NOT, CLEAR THE FIRST FLAG
         001714  000417                                BR      10$
         001716                                8$:
         001716  012767  007762' 176334                MOV     #BASE2,VA            ;GET BASE TABLE 2 ADDRESS.
         001724  004767  002266                        JSR     PC,EABITS            ;GET THE PHYSICAL ADDRESS OF THE TABLE
         001730  026767  003012  176324                CMP     XX25,PA              ;IS THE ADDRESS THE SAME AS ON LAST BASE IN?
         001736  001004                                BNE     9$                   ;IF NOT, CLEAR THE FIRST FLAG
         001740  026767  003004  176316                CMP     XX26,EA              ;CHECK THE EXTENDED ADDRESS
         001746  001402                                BEQ     10$                  ;IF YES, PROCEED.
         001750                                9$:
         001750  005067  176262                        CLR     FIRST                ;CLEAR FIRST FLAG TO FORCE A MASTER CLEAR.
         001754                                10$:
         001754  000167  176710                        JMP     LOOP                 ;LOOP MODULE
         001760                                12$:
                                                                                    ;ONLY THE LAST BUFFER IS CHECKED BECAUSE
                                                                                    ;WE OVERLAY THE BUFFER AREA.
         001760  012703  001000                        MOV     #512.,R3             ;R3 IS THE MAX. BUFFER SIZE
         001764  012704  006362'                       MOV     #RCVBUF,R4           ;R4 POINTS TO FIRST REC BUFFER
         001770  012705  004752'                       MOV     #XBUF,R5             ;R5 POINTS TO GOOD DATA
         001774                                14$:
         001774  121514                                CMPB    (R5),(R4)            ;COMPARE DATA
         001776  001414                                BEQ     15$                  ;BR IF GOOD
         002000  011167  176074                        MOV     (R1),CSRA            ;LOAD CSR
         002004  010567  176072                        MOV     R5,SBADR             ;LOAD GOOD ADDRESS
         002010  010467  176070                        MOV     R4,WASADR            ;LOAD BAD ADDRESS
         002014  111567  176066                        MOVB    (R5),ASB             ;LOAD GOOD DATA
         002020  111467  176064                        MOVB    (R4),AWAS            ;LOAD BAD DATA
                                                       ;***********************************************************
         002024  104404  000000'                       DATER$,BEGIN                 ;DATA ERROR!!!
                                                       ;***********************************************************
```

```
002030                          15$:
002030  122524                          CMPB    (R5)+,(R4)+     ;POP DATA POINTERS
002032  005303                          DEC     R3              ;CONTINUE UNTIL DONE
002034  001357                          BNE     14$             ;
002036  005767  176176                  TST     LAST            ;IS THIS THE LAST PASS?
002042  001665                          BEQ     5$              ;IF NOT - SKIP BASE TABLE ERROR CHECK.
002044  012705  000003                  MOV     #3,R5           ;NOW LETS CHECK BASE TABLE ERROR COUNTS
002050  016104  000006                  MOV     6(R1),R4        ;GET BASE TABLE ADDRESS
002054  060504                          ADD     R5,R4           ;ADD OFFSET TO ERROR COUNTS
002056  010102                          MOV     R1,R2           ;GET POINTER TO CSR
002060  062702  000014                  ADD     #14,R2          ;MAKE IT POINT TO SAVED ERROR COUNTS
002064                          16$:
002064  122224                          CMPB    (R2)+,(R4)+     ;COMPARE BASE TABLE ERROR COUNTS
002066  001005                          BNE     17$             ;TO SAVED ERROR COUNTS BR IF NOT SAME
002070  005205                          INC     R5              ;BUMP TO NEXT ERROR COUNT
002072  022705  000013                  CMP     #13,R5          ;ALL DONE YET?
002076  001372                          BNE     16$             ;BR IF NO
002100  000646                          BR      5$              ;CONTINUE NEXT DEVICE
002102                          17$:
002102  104403  000000' 002354'        MSGN$,BEGIN,SOFT        ;ASCII MESSAGE CALL WITH COMMON HEADER
002110  012705  000003                  MOV     #3,R5           ;BASE OFFSET TO ERROR COUNTS
002114  012702  002360'                MOV     #ESAV1,R2       ;LOAD COUNTS FOR TYPEOUT
002120  016104  000006                  MOV     6(R1),R4        ;GET BASE ADDRESS
002124  060504                          ADD     R5,R4           ;ADD IN OFFSET
002126                          18$:
002126  112422                          MOVB    (R4)+,(R2)+     ;OK LOAD TABLE FOR TYPEOUT
002130  005205                          INC     R5              ;INCREMENT COUNTER
002132  022705  000013                  CMP     #13,R5          ;DONE YET?
002136  001373                          BNE     18$             ;BR IF NOT
                                        ;SAVE THESE BASE TABLE ERROR COUNTS
002140  010102                          MOV     R1,R2           ;GET POINTER TO CSR
002142  062702  000014                  ADD     #14,R2          ;MAKE IT POINT TO SAVED ERROR COUNTS
002146  012704  002360'                MOV     #ESAV1,R4       ;R4 POINTS TO NEW COUNT VALUES
002152                          19$:
002152  012422                          MOV     (R4)+,(R2)+     ;STORE BASE TABLE COUNTS
002154  022704  002370'                CMP     #ESAV4+2,R4     ;DONE?
002160  001374                          BNE     19$             ;BR IF NOT
002162  011167  175712                  MOV     (R1),CSRA       ;LOAD CSR
002166  011105                          MOV     (R1),R5
002170  011567  175706                  MOV     (R5),ACSR       ;SAV CONTENTS OF SEL0
002174  016167  000006  175702          MOV     6(R1),AS1A1     ;SAVE BASE ADDRESS
002202  012767  000001  175676          MOV     #1,ERRTYP       ;DATA ERROR
                                        ;****************************************************************
002210  104406  000000' 002404'        SUPER$,BEGIN,FTABLE     ;BASE TABLE DDCMP ERROR COUNTERS
                                        ;****************************************************************
002216  000167  177374                  JMP     5$              ;CONTINUE NEXT DEVICE


002222  012302                  XERR:   MOV     (R3)+,R2        ;GET POINTER TO CSR
002224  012367  000112                  MOV     (R3)+,DEV       ;GET DEVICE NUMBER
002230  052767  000060  000104          BIS     #60,DEV         ;MAKE IT ASCII
002236  104403  000000' 002344'        MSGN$,BEGIN,DROP1       ;ASCII MESSAGE CALL WITH COMMON HEADER
002244  011201                          MOV     (R2),R1         ;GET CSR ADDRESS
002246  010167  175626                  MOV     R1,CSRA         ;SAVE CSR
002252  011167  175624                  MOV     (R1),ACSR       ;SAVE CONTENTS OF SEL0
002256  016167  000002  175620          MOV     2(R1),ASTAT     ;SAVE CONTENTS OF SEL2
002264  016167  000004  175732          MOV     4(R1),DLY1      ;SAVE CONTENTS OF SEL4
```



```
002272  016167  000006  175726          MOV     6(R1),DLY2      ;SAVE CONTENTS OF SEL6
002300  016267  000002  000052          MOV     2(R2),ESAV1     ;END PASS FLAG
002306  016267  000010  000046          MOV     10(R2),ESAV2    ;RECEIVE BUFFER OFFSET
002314  016267  000012  000042          MOV     12(R2),ESAV3    ;REC/XMIT COUNTERS
002322  012767  000023  175556          MOV     #23,ERRTYP      ;NO INTERRUPT
                                        ;****************************************************************
002330  104405  000000' 002370'        HRDER$,BEGIN,ETABLE     ;DUMP DMC CSR'S AND STATUS FLAGS
                                        ;****************************************************************
002336  005011                          CLR     (R1)            ;SHUT OFF HUNG DMC11
002340  000203                          RTS     R3              ;RETURN

002342  000000                  DEV:    0


002344                          DROP1:
002344  002416'                         XDROP1
002346  002342'                         DEV
002350  002437'                         XDROP2
002352  177777                         -1

002354                          SOFT:
002354  002475'                         SOFT1
002356  177777                         -1

002360  000000                  ESAV1:  0                       ;EXTENDED ERROR PRINTOUT LOCATIONS
002362  000000                  ESAV2:  0
002364  000000                  ESAV3:  0
002366  000000                  ESAV4:  0

002370  000224'                 ETABLE: DLY1                    ;TABLE OF ADDRESSES FOR EXTENDED ERROR PRINTOUT
002372  000226'                         DLY2
002374  002360'                         ESAV1
002376  002362'                         ESAV2
002400  002364'                         ESAV3
002402  177777                         -1

002404  002360'                 FTABLE: ESAV1
002406  002362'                         ESAV2
002410  002364'                         ESAV3
002412  002366'                         ESAV4
002414  177777                         -1

002416      045     104     115 XDROP1: .ASCIZ  /%DMR11 DEVICE # /
002421      122     061     061
002424      040     104     105
002427      126     111     103
002432      105     040     043
002435      040     000
002437      040     111     123 XDROP2: .ASCIZ  / IS HUNG AND HAS BEEN DROPPED/
002442      040     110     125
002445      116     107     040
002450      101     116     104
002453      040     110     101
002456      123     040     102
002461      105     105     116
002464      040     104     122
002467      117     120     120
```

```
002472  105  104  000
002475  045  123  117  SOFT1:  .ASCIZ  /%SOFT ERROR - DDCMP ERROR COUNTERS ARE NON ZERO%/
002500  106  124  040
002503  105  124  122
002506  117  122  040
002511  055  040  104
002514  104  103  115
002517  120  040  105
002522  122  122  117
002525  122  040  103
002530  117  125  116
002533  124  105  122
002536  123  040  101
002541  122  105  040
002544  116  117  116
002547  040  132  105
002552  122  117  045
002555  000
002556  045  111  114  MIXED:  .ASCIZ  /%ILLEGAL TO BE IN DMR MODE IF THE DEVICE IS A DMC%/
002561  114  105  107
002564  101  114  040
002567  124  117  040
002572  102  105  040
002575  111  116  040
002600  104  115  122
002603  040  115  117
002606  104  105  040
002611  111  106  040
002614  124  110  105
002617  040  104  105
002622  126  111  103
002625  105  040  111
002630  123  040  101
002633  040  104  115
002636  103  045  000
002641  045  115  125  NEED2:  .ASCIZ  /%MUST HAVE TWO DEVICES SELECTED FOR LINK TEST%/
002644  123  124  040
002647  110  101  126
002652  105  040  124
002655  127  117  040
002660  104  105  126
002663  111  103  105
002666  123  040  123
002671  105  114  105
002674  103  124  105
002677  104  040  106
002702  117  122  040
002705  114  111  116
002710  113  040  124
002713  105  123  124
002716  045  000
                      .EVEN
```



```
      0 002720                        1ISR:
002720  010577  003426                        MOV     R5,@INQIN       ;STORE LINK POINTER IN QUEUE
002724  062767  000002  003420                ADD     #2,INQIN        ;UPDATE QUEUE
002732  022767  006152' 003412                CMP     #PIRING+200,INQIN ;END OF QUEUE?
002740  001003                                BNE     1$              ;BR IF NO
002742  012767  005752' 003402                MOV     #PIRING,INQIN   ;RESET QUEUE POINTER
002750                                  1$:
002750  012605                                MOV     (SP)+,R5        ;RESTORE R5
                                              ;-----------------------------------------------------------------
002752  000004  000000' 002760'               PIRQ$,BEGIN,2$          ; QUEUE UP TO CONTINUE AT 2$ AND RT1
                                              ;-----------------------------------------------------------------
002760                                  2$:
002760  017705  003370                        MOV     @1NQOUT,R5      ;GET LINK POINTER IN R5
002764  062767  000002  003362                ADD     #2,INQOUT       ;UPDATE QUEUE
002772  022767  006152' 003354                CMP     #PIRING+200,INQOUT ;END OF QUEUE?
003000  001003                                BNE     3$              ;BR IF NO
003002  012767  005752' 003344                MOV     #PIRING,INQOUT  ;RESET QUEUE POINTER
003010                                  3$:
003010  016501  000004                        MOV     4(R5),R1        ;LOAD CSR ADDRESS
003014  011100                                MOV     @R1,R0          ;SAVE SEL0
003016  042700  177760                        BIC     #177760,R0      ;KEEP ONLY THE COMMAND BITS (0-3)
003022  020027  000000                        CMP     R0,#BACCIT      ;IS THE COMMAND BA/CC XMIT ?
003026  001503                                BEQ     XMIT            ;BR IF YES
003030  020027  000004                        CMP     R0,#BACCIR      ;IS THE COMMAND BA/CC RECV ?
003034  001501                                BEQ     REC             ;BRANCH IF YES
003036  020027  000001                        CMP     R0,#CNTIN       ;IS THE COMMAND A CONTROL IN?
003042  001466                                BEQ     CNTRL           ;BRANCH IF YES
003044  020027  000003                        CMP     R0,#BASEI       ;IS THE COMMAND A BASE IN?
003050  001413                                BEQ     BASEIN          ;BR IF YES
003052  020027  000002                        CMP     R0,#FINI        ;IS THIS THE HALT COMMAND
003056  001402                                BEQ     DONE            ;BR IF YES.
003060  000167  000422                        JMP     DMRIN           ;IF NOT ONE OF THE ABOVE MUST BE
                                                                      ;AN EXTENDED DMR COMMAND.
                                                                      ;JUMP TO PROCESS EXTENDED COMMAND.
003064                                  DONE:
003064  052711  000100                        BIS     #100,(R1)       ;CLEAR INTERRUPT ENABLE
003070  004767  001110                        JSR     PC,ENDCLR       ;CLEAR RQI
003074  104400  000000'                       EXIT$,BEGIN                     ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.

003100                                  BASEIN:                         ;BASE IN
003100  016567  000012  175152                MOV     12(R5),VA       ;LOAD VA WITH VIRTUAL ADDRESS
003106  004767  001104                        JSR     PC,EABITS       ;GET PHYSICAL ADDRESS
003112  016761  175144  000004                MOV     PA,4(R1)        ;LOAD PHYSICAL BASE ADDRESS
003120  016761  175140  000006                MOV     EA,6(R1)        ;LOAD EA BITS FOR BASE ADDRESS
003126  016765  175130  000032                MOV     PA,32(R5)       ;SAVE THE PHYSICAL ADDRESS.
003134  016765  175124  000034                MOV     EA,34(R5)       ;SAVE THE EXTENDED ADDRESS.
003142  005767  175104                        TST     MODE            ;IS THIS DMC MODE?
003146  001403                                BEQ     6$              ;IF NOT, NO NEED TO SET MODE.
003150  052761  000522  000006                BIS     #DMR,6(R1)      ;SET DMR MODE.
003156                                  6$:
003156  004767  001022                        JSR     PC,ENDCLR       ;CLEAR RQI
003162  052765  000001  000006                BIS     #1,6(R5)        ;SET BIT0 IN END PASS FLAG TO
                                                                      ;SHOW THAT A BASE ADRESS WAS LOADED
003170  005767  175056                        TST     MODE            ;WHAT IS THE MODE
003174  001405                                BEQ     7$              ;BR IF IN DMC MODE
003176  005765  000030                        TST     30(R5)          ;ARE ANY DMR MODE EXTENDED COMMANDS
                                                                      ;REQUESTED ?
```

```
003202  001402                          BEQ     7$              ;BR IF NONE
003204  000167  000422                  JMP     DMRNXT          ;ISSUE THE DMR COMMAND
003210                          7$:
003210  152711  000041                  BISB    #41,(R1)        ;ASK FOR CNTL I
003214  104400  000000'                  EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
003220                          CNTRL:
003220  005061  000006                  CLR     6(R1)           ;SET FULL DUPLEX
003224  004767  000754                  JSR     PC,ENDCLR       ;CLEAR RQI
003230  152711  000044                  BISB    #44,(R1)        ;ASK FOR REC BA/CC I
003234  104400  000000'                  EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
003240                          REC:
003240  012767  006362' 175012          MOV     #RCVBUF,VA      ;VA GETS REC BUFFER VIRTUAL ADDRESS
003246  004767  000744                  JSR     PC,EABITS       ;GET PHYSICAL ADDRESS
003252  016761  175004  000004          MOV     PA,4(R1)        ;LOAD PHYSICAL RBUF
003260  016761  175000  000006          MOV     EA,6(R1)        ;LOAD RBUF EA BITS
003266  052761  001000  000006          BIS     #512.,6(R1)     ;MAX BUFFER SIZE POSSIBLE
003274  004767  000704                  JSR     PC,ENDCLR       ;CLEAR RQI
003300  105265  000014                  INCB    14(R5)          ;UPDATE COUNTER.
003304  126765  174736  000014          CMPB    BUFNUM,14(R5)   ;LOADED ALL BUFFERS YET?
003312  001404                          BEQ     1$              ;BR IF YES
003314  152711  000044                  BISB    #44,(R1)        ;REQUEST REC BA/CC
003320  104400  000000'                  EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
003324                          1$:
003324  052765  000002  000006          BIS     #2,6(R5)        ;SET BIT1 IN END PASS FLAG TO
                                                                ;SHOW THAT ALL 7 REC BA/CC WERE LOADED
003332  005767  174716                  TST     LINK            ;IS THIS A LINK TEST?
003336  001003                          BNE     2$              ;IF YES - BR
003340  152711  000040                  BISB    #40,(R1)        ;ASK FOR XMIT BA/CC I
003344  000412                          BR      3$              ;EXIT
003346                          2$:
003346  032767  000002  001346          BIT     #2,XX21         ;IS THE 2ND DEVICE RCV. DONE?
003354  001406                          BEQ     3$              ;IF NOT, DON'T ASK FOR XMIT.
003356  152777  000040  001272          BISB    #40,@CSR1       ;BA/CC IN XMIT FOR DEVICE 1.
003364  152777  000040  001326          BISB    #40,@CSR2       ;BA/CC IN XMIT FOR DEVICE 2.
003372                          3$:
003372  104400  000000'                  EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
003376                          XMIT:
003376  012767  004752' 174654          MOV     #XBUF,VA        ;LOAD XBUF VIRTUAL ADDRESS
003404  004767  000606                  JSR     PC,EABITS       ;GET PHYSICAL ADDRESS
003410  016761  174646  000004          MOV     PA,4(R1)        ;LOAD PHYSICAL XBUF
003416  016761  174642  000006          MOV     EA,6(R1)        ;LOAD EA BITS
003424  116500  000015                  MOVB    15(R5),R0       ;SEE WHAT BUFFER NUMBER THIS WILL BE
003430  042700  177770                  BIC     #177770,R0      ;SAVE ONLY THE LAST DIGIT (0-7)
003434  006300                          ASL     R0              ;GET LOCATION IN BUFFER TABLE
003436  056061  000326' 000006          BIS     BUFFER(R0),6(R1);LOAD TRANSMIT COUNT
                                                                ;ACCORDING TO THE BUFFER TABLE
                                                                ;+OFFSET IN ORDER TO VARY BUFFER SIZE.
003444  004767  000534                  JSR     PC,ENDCLR       ;CLEAR RQI
003450  105265  000015                  INCB    15(R5)          ;COUNT HOW MANY XMIT BA/CC ARE DONE.
003454  126765  174566  000015          CMPB    BUFNUM,15(R5)   ;DO WE HAVE ALL BUFFERS?
003462  001404                          BEQ     1$              ;BR IF YES
003464  152711  000040                  BISB    #40,(R1)        ;ASK FOR XMIT BA/CC
003470  104400  000000'                  EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
003474                          1$:
003474  052765  000004  000006          BIS     #4,6(R5)        ;SET BIT2 IN END PASS FLAG TO
                                                                ;SHOW THAT ALL 7 XMIT BA/CC WERE LOADED
003502  104400  000000'                  EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```



```
003506                          DMRIN:
003506  020027  000005                  CMP     R0,#MW          ;IS THIS A MODEM WRITE?
003512  001007                          BNE     1$              ;BR IF NOT
003514  046761  174560  000006          BIC     MODEM1,6(R1)    ;CLEAR NECESSARY BITS
003522  056761  174554  000006          BIS     MODEM2,6(R1)    ;SET NECESSARY BITS
003530  000436                          BR      10$
003532                          1$:
003532  020027  000012                  CMP     R0,#TIMER       ;IS THIS TO SET REP/SELECT TIMER VALUE ?
003536  001004                          BNE     2$              ;BR IF NOT
003540  016761  174540  000006          MOV     TIME,6(R1)      ;LOAD VALUE
003546  000427                          BR      10$
003550                          2$:
003550  020027  000013                  CMP     R0,#THRESH      ;IS THIS TO SET THE THRESHOLD VALUES?
003554  001007                          BNE     3$              ;BR IF NOT
003556  016761  174524  000004          MOV     THRSH1,4(R1)    ;LOAD VALUE FOR NAKS
                                                                ;BSEL4 = NAKS REC  BSEL5 = NAKS SENT
003564  016761  174520  000006          MOV     THRSH2,6(R1)    ;LOAD VALUE FOR REP/SEL AND NO BUFFER
                                                                ;BSEL6 = REP/SEL   BSEL7 = NO BUFFER
003572  000415                          BR      10$
003574                          3$:
003574  020027  000014                  CMP     R0,#RRAM        ;IS THIS THE READ M8207 RAM?
003600  001004                          BNE     5$              ;BR IF NOT.
003602  116761  174504  000006          MOVB    MRAM,6(R1)      ;ADDRESS OF RAM LOADED IN SEL6
                                                                ;CONTENTS RETURNED IN BSEL3.
003610  000406                          BR      10$
003612                          5$:
003612  020027  000015                  CMP     R0,#INTER       ;IS THIS THE AX3-15 WRITE?
003616  001003                          BNE     10$             ;BR IF NOT
                                                                ;****** WARNING ********
                                                                ;MAKE SURE THAT AX3 HAS THE CORRECT
                                                                ;VALUE FOR THE DESIRED INTERFACE. SEE
                                                                ;THE COMMENTS ON AX3 IN THE VARIABLE
                                                                ;LISTINGS.
                                                                ;************************
003620  116761  174470  000007          MOVB    AX3,7(R1)       ;LOAD AX3-15 VALUE
003626                          10$:
003626  004767  000352                  JSR     PC,ENDCLR       ;CLEAR RQI.
003632                          DMRNXT:
003632  005765  000030                  TST     30(R5)          ;SEE IF THERE ARE ANY DMR MODE COMMANDS
003636  001556                          BEQ     20$
                                ;
                                ;   DMRXT IS A FLAG THAT ALLOWS THE USER TO SELECT ANY COMBINATION OF THE
                                ;   DMR COMMANDS TO BE PERFORMED (IF SET)
                                ;       BIT 0 = MODEM WRITE             BIT 1 = ENABLE EXT ERROR
                                ;       BIT 2 = DISABLE EXT ERROR       BIT 3 = DESELECT DMC LINE
                                ;       BIT 4 = REQUEST BASE T. UPDATE  BIT 5 = SET REP/SELECT TIMER
                                ;       BIT 6 = SET THRESHOLD VALUES    BIT 7 = READ M8207 RAM
                                ;       BIT 8 = WRITE AX3-15                    BIT 9 = NOP
                                ;       BIT 10= READ MODEM
                                ;
003640  032765  000001  000030          BIT     #BIT0,30(R5)    ;MODEM WRITE?
003646  001406                          BEQ     1$              ;BR IF NOT
003650  042765  000001  000030          BIC     #BIT0,30(R5)    ;CLEAR THE REQUEST
003656  152711  000045                  BISB    #401MW,(R1)     ;ASK FOR A MODEM WRITE
003662  000546                          BR      25$
003664                          1$:
```

```
003664  032765  000002  000030          BIT     #BIT1,30(R5)    ;ENABLE EXT ERROR?
003672  001406                          BEQ     2$              ;BR IF NOT
003674  042765  000002  000030          BIC     #BIT1,30(R5)    ;CLEAR THE REQUEST
003702  152711  000046                  BISB    #40!EXER,(R1)   ;ASK FOR ENABLE EXT ERROR.
003706  000534                          BR      25$
003710                          2$:
003710  032765  000004  000030          BIT     #BIT2,30(R5)    ;DISABLE EXT ERROR?
003716  001406                          BEQ     3$              ;BR IF NOT
003720  042765  000004  000030          BIC     #BIT2,30(R5)    ;CLEAR THE REQUEST
003726  152711  000047                  BISB    #40!DXER,(R1)   ;ASK FOR DISABLE EXT ERROR.
003732  000522                          BR      25$
003734                          3$:
003734  032765  000010  000030          BIT     #BIT3,30(R5)    ;DESELECT DMC LINE MODE?
003742  001406                          BEQ     4$              ;BR IF NOT
003744  042765  000010  000030          BIC     #BIT3,30(R5)    ;CLEAR THE REQUEST
003752  152711  000050                  BISB    #40!DDMC,(R1)   ;ASK FOR DESELT DMC.
003756  000510                          BR      25$
003760                          4$:
003760  032765  000020  000030          BIT     #BIT4,30(R5)    ;REQUEST BASE TABLE UPDATE?
003766  001406                          BEQ     5$              ;BR IF NOT
003770  042765  000020  000030          BIC     #BIT4,30(R5)    ;CLEAR THE REQUEST
003776  152711  000051                  BISB    #40!UPDATE,(R1) ;ASK FOR UPDATE.
004002  000476                          BR      25$
004004                          5$:
004004  032765  000040  000030          BIT     #BIT5,30(R5)    ;SET REP/SELECT TIMER VALUE?
004012  001406                          BEQ     6$              ;BR IF NOT
004014  042765  000040  000030          BIC     #BIT5,30(R5)    ;CLEAR THE REQUEST
004022  152711  000052                  BISB    #40!TIMER,(R1)  ;SET TIMER
004026  000464                          BR      25$
004030                          6$:
004030  032765  000100  000030          BIT     #BIT6,30(R5)    ;SET THRESHOLD VALUES?
004036  001406                          BEQ     7$              ;BR IF NOT
004040  042765  000100  000030          BIC     #BIT6,30(R5)    ;CLEAR THE REQUEST
004046  152711  000053                  BISB    #40!THRESH,(R1) ;SET THRESHOLD.
004052  000452                          BR      25$
004054                          7$:
004054  032765  000200  000030          BIT     #BIT7,30(R5)    ;READ M8207 RAM?
004062  001406                          BEQ     8$              ;BR IF NOT
004064  042765  000200  000030          BIC     #BIT7,30(R5)    ;CLEAR THE REQUEST
004072  152711  000054                  BISB    #40!RRAM,(R1)   ;READ M8207 RAM.
004076  000440                          BR      25$
004100                          8$:
004100  032765  000400  000030          BIT     #BIT8,30(R5)    ;WRITE AX3-15?
004106  001406                          BEQ     9$              ;BR IF NOT
004110  042765  000400  000030          BIC     #BIT8,30(R5)    ;CLEAR THE REQUEST
004116  152711  000055                  BISB    #40!INTER,(R1)  ;WRITE AX3-15
004122  000426                          BR      25$
004124                          9$:
004124  032765  001000  000030          BIT     #BIT9,30(R5)    ;NOP?
004132  001406                          BEQ     10$             ;BR IF NOT
004134  042765  001000  000030          BIC     #BIT9,30(R5)    ;CLEAR THE REQUEST
004142  152711  000056                  BISB    #40!NOP1,(R1)   ;NOP
004146  000414                          BR      25$
004150                          10$:
004150  032765  002000  000030          BIT     #BIT10,30(R5)   ;READ MODEM?
004156  001406                          BEQ     20$             ;IF NOT, ISSUE A CONTROL IN
004160  042765  002000  000030          BIC     #BIT10,30(R5)   ;CLEAR THE REQUEST
```



```
004166  152711  000057                  BISB    #40!RMODEM,(R1) ;READ MODEM.
004172  000402                          BR      25$
004174                          20$:
004174  152711  000041                  BISB    #41,(R1)        ;CONTROL IN
004200                          25$:
004200  104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

                                                                ;
004204                          ENDCLR:
004204  142711  000040                  BICB    #40,(R1)        ;CLEAR RQI
004210                          1$:
004210  105711                          TSTB    (R1)            ;IS RDI GONE?
004212  100776                          BMI     1$              ;BR IF NO
004214  000207                          RTS     PC              ;RETURN

004216                          EABITS:
004216  104415  000000' 000260'         GETPAS,BEGIN, VA                ;GET PHYSICAL ADDRESS FROM 16-BIT VA
004224  000367  174034                  SWAB    EA              ;BITS 4+5 TO 13+12
004230  006167  174030                  ROL     EA              ;NOW 14+13
004234  006167  174024                  ROL     EA              ;NOW 15+14
004240  042767  037776  174016         BIC     #37776,EA       ;CLEAR ALL BUT 14 & 15
004246  000207                          RTS     PC              ;RETURN
```

```
      0 004250                          UISR:
        004250  010577  002102                   MOV     R5,@OUTQIN      ;MOVE LINK POINTER TO QUEUE
        004254  062767  000002  002074           ADD     #2,OUTQIN       ;UPDATE QUEUE
        004262  022767  006352' 002066           CMP     #PIROUTQ+200,OUTQIN ;END OF QUEUE?
        004270  001003                           BNE     1$              ;BR IF NO
        004272  012767  006152' 002056           MOV     #PIROUTQ,OUTQIN ;RESET QUEUE POINTER
        004300                          1$:
        004300  012605                           MOV     (SP)+,R5        ;RESTORE R5
                                                ;---------------------------------------------------------------
        004302  000004  000000' 004310'          PIRQS,BEGIN,2$          ; QUEUE UP TO CONTINUE AT 2$ AND RTI
                                                ;---------------------------------------------------------------
        004310                          2$:
        004310  017705  002044                   MOV     @OUTQOUT,R5     ;GET LINK POINTER FROM QUEUE
        004314  062767  000002  002036           ADD     #2,OUTQOUT      ;UPDATE QUEUE
        004322  022767  006352' 002030           CMP     #PIROUTQ+200,OUTQOUT ;END OF QUEUE?
        004330  001003                           BNE     3$              ;BR IF NO
        004332  012767  006152' 002020           MOV     #PIROUTQ,OUTQOUT;RESET QUEUE POINTER
        004340                          3$:
        004340  011501                           MOV     (R5),R1         ;LOAD CSR ADDRESS
        004342  032761  000001  000002           BIT     #BIT0,2(R1)     ;CONTROL OUT?
        004350  001451                           BEQ     15$             ;BR IF NO
        004352  032761  001000  000006           BIT     #HALTC,6(R1)    ;HALT?
        004360  001407                           BEQ     5$              ;BR IF NOT.
        004362  052765  000040  000002           BIS     #40,2(R5)       ;SET THE END OF PASS FLAG.
        004370  042761  000100  000002           BIC     #100,2(R1)      ;CLEAR DIE IF YES
        004376  000407                           BR      8$
        004400                          5$:
        004400  005767  173646                   TST     MODE            ;ARE WE IN DMC MODE?
        004404  001411                           BEQ     10$             ;IF YES, CAN ONLY BE AN ERROR
        004406  032761  020040  000006           BIT     #DR,6(R1)       ;IF DMR, IS THIS DMR RUN OR
                                                                        ;BASE TABLE UPDATE
        004414  001405                           BEQ     10$             ;IF NOT, MUST BE AN ERROR.
        004416                          8$:
        004416  142761  000207  000002           BICB    #207,2(R1)      ;CLEAR RDO (NOT AN ERROR)
        004424  104400  000000'                  EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
        004430                          10$:
        004430  010167  173444                   MOV     R1,CSRA         ;LOAD DEVICE CSR
        004434  016167  000004  173440           MOV     4(R1),ACSR      ;LOAD CONTENTS OF DEVICE CSR
        004442  016167  000006  173434           MOV     6(R1),ASTAT     ;LOAD ERROR BITS
        004450  005067  173432                   CLR     ERRTYP          ;UNKNOWN ERROR
                                                ;***************************************************************
        004454  104405  000000' 000000          HRDER$,BEGIN,NULL       ;A CNTL O WAS RECEIVED, ASTAT=ERROR BITS
                                                ;***************************************************************
        004462  142761  000207  000002           BICB    #207,2(R1)      ;CLEAR RDO
        004470  104400  000000'                  EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
        004474                          15$:
        004474  032761  000004  000002           BIT     #BIT2,2(R1)     ;RECEIVER DONE?
        004502  001017                           BNE     17$             ;BR IF YES
        004504  142761  000207  000002           BICB    #207,2(R1)      ;CLEAR RDO
        004512  105265  000013                   INCB    13(R5)          ;HI BYTE IS XMIT DONE COUNT
        004516  126765  173524  000013           CMPB    BUFNUM,13(R5)   ;DO WE HAVE ALL XMIT DONES YET?
        004524  001004                           BNE     16$             ;BR IF NO
        004526  052765  000010  000002           BIS     #10,2(R5)       ;SET BIT3 IN ENDPASS FLAG TO
                                                                        ;SHOW THAT WE GOT 7 XMIT DONES
        004534  000416                           BR      18$             ;BR TO COMMON EXIT.
        004536                          16$:
        004536  104400  000000'                  EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```



```
        004542                          17$:
        004542  142761  000207  000002           BICB    #207,2(R1)      ;CLEAR RDO
        004550  105265  000012                   INCB    12(R5)          ;LO BYTE IS REC DONE COUNT
        004554  126765  173466  000012           CMPB    BUFNUM,12(R5)   ;DO WE HAVE ALL REC DONES YET?
        004562  001027                           BNE     20$             ;BR IF NO
        004564  052765  000020  000002           BIS     #20,2(R5)       ;SET BIT4 IN ENDPASS FLAG TO
                                                                        ;SHOW THAT WE GOT ALL 7 REC DONES
        004572                          18$:
        004572  026765  173434  000002           CMP     FLAGA,2(R5)     ;ALL BUT HALT DONE?
        004600  001020                           BNE     20$             ;BR IF NOT
        004602  005767  173432                   TST     LAST            ;IS THIS THE LAST PASS?
        004606  001403                           BEQ     19$             ;IF NOT - DON'T HALT
        004610  052711  000042                   BIS     #42,(R1)        ;ISSUE A HALT.
        004614  000412                           BR      20$
        004616                          19$:
        004616  052765  000040  000002           BIS     #40,2(R5)       ;SET THE END OF PASS FLAG.
        004624  042711  000100                   BIC     #100,(R1)       ;CLEAR IIE IF YES
        004630  042761  000100  000002           BIC     #100,2(R1)      ;CLEAR DIE IF YES
        004636  042711  100000                   BIC     #BIT15,(R1)     ;TURN OFF THE RUN BIT.
        004642                          20$:
        004642  104400  000000'                  EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```

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

004646                                  INTLNK:
004646  004567  176046                          JSR     R5,1ISR
004652  004567  177372                          JSR     R5,0ISR
004656  000000                          CSR1:   .WORD                   ;DMR CSR FOR DEVICE 1
004660  000000                          XX11:   .WORD                   ;END PASS FLAG FOR DEVICE 1
004662  006362'                                 RCVBUF                  ;RECEIVE BUFFER 1
004664  007362'                                 BASE1                   ;BASE ADDRESS FOR DEVICE 1
004666  000000                          XX12:   .WORD                   ;BACCI XMIT/REC COUNTERS
004670  000000                          XX13:   .WORD                   ;BACCO XMIT/REC COUNTERS
004672  000000                                  .WORD                   ;THESE NEXT 8 BYTES ARE FOR
004674  000000                                  .WORD                   ;THE DDCMP ERROR COUNTS
004676  000000                                  .WORD                   ;IN THE BASE TABLE TO BE
004700  000000                                  .WORD                   ;SAVED FOR COMPARISION.
004702  000000                          XX14:   .WORD                   ;DMREXT FLAG FOR THE DEVICE 1
                                                                        ;THIS FLAG CONTROLS DMR COMMANDS.
004704  000000                          XX15:   .WORD                   ;PHYSICAL BASE ADDR FOR DEVICE 1
004706  000000                          XX16:   .WORD                   ;EXTENDED BASE ADDR FOR DEVICE 1

004710  004567  176004                          JSR     R5,1ISR
004714  004567  177330                          JSR     R5,0ISR
004720  000000                          CSR2:   .WORD                   ;DMR CSR FOR DEVICE 2
004722  000000                          XX21:   .WORD                   ;END PASS FLAG FOR DEVICE 2
004724  006362'                                 RCVBUF                  ;RECEIVE BUFFER 2
004726  007762'                                 BASE2                   ;BASE ADDRESS FOR DEVICE 2
004730  000000                          XX22:   .WORD                   ;BACCI XMIT/REC COUNTERS
004732  000000                          XX23:   .WORD                   ;BACCO XMIT/REC COUNTERS
004734  000000                                  .WORD                   ;THESE NEXT 8 BYTES ARE FOR
004736  000000                                  .WORD                   ;THE DDCMP ERROR COUNTS
004740  000000                                  .WORD                   ;IN THE BASE TABLE TO BE
004742  000000                                  .WORD                   ;SAVED FOR COMPARISION.
004744  000000                          XX24:   .WORD                   ;DMREXT FLAG FOR THE DEVICE 2
                                                                        ;THIS FLAG CONTROLS DMR COMMANDS.
004746  000000                          XX25:   .WORD                   ;PHYSICAL BASE ADDR FOR DEVICE 2
004750  000000                          XX26:   .WORD                   ;EXTENDED BASE ADDR FOR DEVICE 2
```



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

004752                                  XBUF:           ;TRANSMIT BUFFER (512. BYTES)
                                                        ; ** CCITT PSUEDO-RANDOM TEST PATTERN **
                                                        ; THE FOLLOWING 32 WORDS TRANSLATE INTO A 512.
                                                        ; BIT PATTERN THAT WAS GENERATED ACCORDING TO CCITT
                                                        ; RECOMMENDATION V.52.  NOTE: THE RECOMMENDED PATTERN
                                                        ; IS 511 BITS  - THIS WAS EXTENDED BY 1 BIT TO END
                                                        ; ON AN EVEN WORD BOUNDARY.

        000010                          .REPT   8.
004752  177603  157427  031011                .WORD   177603,157427,031011
004760  047321  163715  105221                .WORD   047321,163715,105221
004766  143325  142304  040041                .WORD   143325,142304,040041
004774  014116  052606  172334                .WORD   014116,052606,172334
005002  105025  123754  111337                .WORD   105025,123754,111337
005010  111523  030030  145064                .WORD   111523,030030,145064
005016  137642  143531  063617                .WORD   137642,143531,063617
005024  135015  066730  026575                .WORD   135015,066730,026575
005032  052012  053627  070071                .WORD   052012,053627,070071
005040  151172  165044  031605                .WORD   151172,165044,031605
005046  166632  016741                        .WORD   166632,016741
                                        .LIST   ME

005752                                  PIRING: .BLKW 100       ;QUEUE FOR 64. IN INTERRUPTS
006152                                  PIROUTQ:.BLKW 100       ;QUEUE FOR 64. OUT INTERRUPTS
006352  000000                          INQIN:  0
006354  000000                          INQOUT: 0
006356  000000                          OUTQIN: 0
006360  000000                          OUTQOUT:0
006362                                  RCVBUF: .BLKB   512.    ;RECEIVE BUFFER IS 512. BYTES


007362                                  BASE1:  .BLKB   256.    ;BASE TABLE FOR DEVICE 1


007762                                  BASE2:  .BLKB   256.    ;BASE TABLE FOR DEVICE 2




                                        ;PATCH AREA
                                        ;----------------
010362  000240                          PATCH:  NOP
                                                .BLKW   20.             ;PATCH AREA


1       000001                          .END
```

SYMBOL TABLE

```
ACSR    000102R     CSR1    004656R     IDNUM   000122R     PRTY0 = 000000     START   000346R
ADDR    000006R     CSR2    004720R     IISR    002720R     PRTY1 = 000040     STAT    000026R
ADDR22= 001000      DATCK$= 104411      INIT    000030R     PRTY2 = 000100     STOP  = 001000
ASB     000106R     DATER$= 104404      INQIN   006352R     PRTY3 = 000140     SVR0    000062R
ASTAT   000104R     DDMC  = 000010      INQOUT  006354R     PRTY4 = 000200     SVR1    000064R
AWAS    000110R     DEV     002342R     INTER = 000015      PRTY5 = 000240     SVR2    000066R
AX3     000314R     DLY1    000224R     INTLNK  004646R     PRTY6 = 000300     SVR3    000070R
BACC1R= 000004      DLY2    000226R     INTR    000120R     PRTY7 = 000340     SVR4    000072R
BACCIT= 000000      DMR   = 000522      LAS1    000240R     PS    = 177776     SVR5    000074R
BASE1 = 000003      DMREXT  000250R     LINK    000254R     PSW   = 177776     SVR6    000076R
BASEIN  003100R     DMRIN   003506R     LOOP    000670R     PUSH  = 005746     SYSCNT  000052R
BASE1   007362R     DMRNXT  003632R     LULOOP  000256R     PUSH2 = 024646     TERM  = 000000
BASE2   007762R     DONE    003064R     MAP22$= 104416      PWRFLG= 000002     THRESH= 000013
BEGIN   000000R     DROP    000634R     MASK    000244R     RAND$ = 104417     THRSH1  000306R
BIT0  = 000001      DROP1   002344R     MIXED   002556R     RANNUM  000054R     THRSH2  000310R
BIT1  = 000002      DVID1   000014R     MODE    000252R     RCVBUF  006362R     TIME    000304R
BIT10 = 002000      DXER  = 000007      MODEM1  000300R     REC     003240R     TIMER = 000012
BIT11 = 004000      EA      000264R     MODEM2  000302R     RESTRT  000640R     TRPDFD= 000022
BIT12 = 010000      EABITS  004216R     MODNAM  000000R     RESUME= 010000     UPDATE= 000011
BIT13 = 020000      ENDCLR  004204R     MODSP   000224R     RES1    000056R     VA      000260R
BIT14 = 040000      ENDITS= 104413      MRAM    000312R     RES2    000060R     VECTOR  000010R
BIT15 = 100000      ENDS  = 104410      MSGN$ = 104403      RMODEM= 000017     WASADR  000104R
BIT2  = 000004      ERRTYP  000106R     MSGS$ = 104402      RRAM  = 000014     WDFR    000116R
BIT3  = 000010      ESAV1   002360R     MSG$  = 104401      RSTRT   000112R     WDTO    000114R
BIT4  = 000020      ESAV2   002362R     MW    = 000005      R6    =%000006     XBUF    004752R
BIT5  = 000040      ESAV3   002364R     NEED2   002641R     R7    =%000007     XDROP1  002416R
BIT6  = 000100      ESAV4   002366R     NOP1  = 000016      SAR0    000266R     XDROP2  002437R
BIT7  = 000200      ETABLE  002370R     NULL  = 000000      SAR1    000270R     XERR    002222R
BIT8  = 000400      EXER  = 000006      OISR    004250R     SBADR   000102R     XFLAG   000005R
BIT9  = 001000      EXITS = 104400      OK    = 020040      SCAN    001416R     XMIT    003376R
BREAK$= 104407      FIN1  = 000002      OPEN  = 000000      SELECT  000230R     XX11    004660R
BR1     000012R     FIRST   000236R     OTOA$ = 104420      SETUP2  001132R     XX12    004666R
BR2     000013R     FLAGA   000232R     OUTQIN  006356R     SOFCNT  000042R     XX13    004670R
BTOD$ = 104421      FLAGB   000234R     OUTQOU  006360R     SOFER$= 104406     XX14    004702R
BUFFER  000326R     FTABLE  002404R     PA      000262R     SOFPAS  000046R     XX15    004704R
BUFNUM  000246R     GETPA$= 104415      PASCN1  000034R     SOFT    002354R     XX16    004706R
CDATA$= 104412      GWBUF$= 104414      PATCH   010362R     SUF11   002475R     XX21    004722R
CHECK   001600R     HALTC = 001000      PIRING  005752R     SPOINT  000032R     XX22    004730R
CNTIN = 000001      HRDCNT  000044R     PIROUT  006152R     SPSIZ = 000040     XX23    004732R
CNTRL   003220R     HRDER$= 104405      PIRQ$ = 000004      SR1     000016R     XX24    004744R
CONFIG  000056R     HRDPAS  000050R     POPSP = 005726      SR2     000020R     XX25    004746R
COUNT   000242R     ICONT   000036R     POPSP2= 022626      SR3     000022R     XX26    004750R
CSRA    000100R     ICOUNT  000040R     PRTY  = 000000      SR4     000024R
```

```
. ABS.  000000      000
        010434      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  19848 WORDS  ( 78 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  69 PAGES
XDMRB0,CXDMRB.SEQ/C=#DDXCOM.P11,CXDMRB.P11
```


CROSS REFERENCE TABLE (CREF V01-05)

```
ACSR      6-1#    7-4*    7-4*    7-4*    8-0*    8-0*    10-0*
ADDR      6-1#    7-4     7-4     7-4
ADDR22    6-1#
ASB       6-1#    8-0*
ASTAT     6-1#    7-4*    7-4*    8-0*    8-0*    10-0*
AWAS      6-1#    8-0*
AX3       6-2#    9-0
BACC1R    6-2#    9-0
BACCIT    6-2#    9-0
BASE1     7-4     8-0     11-0    12-0*
BASE2     8-0     11-0    12-0*
BASE1     6-2#    9-0
BASEIN    9-0     9-0*
BEGIN     6-1#    7-4     7-4     7-4     7-4     7-4     7-4     7-4     7-4     8-0     8-0     8-0     8-0     8-0
          8-0     8-0     8-0     9-0     9-0     9-0     9-0     9-0     9-0     9-0     9-0     9-0     9-0     10-0
          10-0    10-0    10-0    10-0    10-0
BIT0      6-1#    7-4     8-0     9-0     9-0     10-0
BIT1      6-1#    7-4     9-0     9-0
BIT10     6-1#    9-0     9-0
BIT11     6-1#
BIT12     6-1#
BIT13     6-1#
BIT14     6-1#    7-4
BIT15     6-1#    7-4     10-0
BIT2      6-1#    7-4     9-0     9-0     10-0
BIT3      6-1#    9-0     9-0
BIT4      6-1#    9-0     9-0
BIT5      6-1#    9-0     9-0
BIT6      6-1#    9-0     9-0
BIT7      6-1#    9-0     9-0
BIT8      6-1#    9-0     9-0
BIT9      6-1#    9-0     9-0
BR1       6-1#    7-4     7-4
BR2       6-1#
BREAK$    6-1#    7-4     7-4     8-0     8-0
BTOD$     6-1#
BUFFER    6-2#    9-0
BUFNUM    6-2#    7-4     7-4*    7-4*    9-0     9-0     10-0    10-0
CDATA$    6-1#
CHECK     8-0     8-0#
CNTIN     6-2#    9-0
CNTRL     9-0     9-0#
CONFIG    6-1#
COUNT     6-2#    7-4*    7-4*    7-4*    8-0*
CSR1      8-0     9-0*    11-0*
CSR2      8-0     9-0*    11-0*
CSRA      6-1#    7-4     7-4*    7-4*    8-0*    8-0*    8-0*    10-0*
DATCK$    6-1#
DATER$    6-1#    8-0
DDMC      6-2#    9-0
DEV       8-0     8-0*    8-0*    8-0*
DLY1      6-2#    8-0     8-0*    8-0*    8-0*
DLY2      6-2#    8-0     8-0*    8-0*    8-0*
DMR       6-2#    9-0
DMREXT    6-2#    7-4     7-4*    7-4*
DMRIN     9-0     9-0#
```

CROSS REFERENCE TABLE (CREF V01-05)

```
DMRNXT   9-0     9-0#
DONE     9-0     9-0#
DROP     7-4     7-4      7-4      7-4#
DROP1    8-0     8-0#
DVID1    6-1#    7-4      7-4
DXER     6-2#    9-0
EA       6-2#    8-0      8-0      9-0      9-0      9-0      9-0      9-0*     9-0*     9-0*     9-0*
EABITS   8-0     8-0      9-0      9-0      9-0      9-0#
ENDS     6-1#    7-4      7-4
ENDCLR   9-0     9-0      9-0      9-0      9-0      9-0      9-0#
ENDITS   6-1#    8-0
ERRTYP   6-1#    7-4*     7-4*     8-0*     8-0*     10-0*
ESAV1    8-0     8-0      8-0      8-0      8-0#     8-0*
ESAV2    8-0     8-0      8-0#     8-0*
ESAV3    8-0     8-0      8-0#     8-0*
ESAV4    8-0     8-0      8-0#
ETABLE   8-0     8-0#
EXER     6-2#    9-0
EXITS    6-1#    9-0      9-0      9-0      9-0      9-0      9-0      9-0      9-0      10-0     10-0     10-0     10-0
FIN1     6-2#    9-0
FIRST    6-2#    7-4      7-4      7-4      7-4      7-4*     8-0*     8-0*
FLAGA    6-2#    10-0
FLAGB    6-2#    7-4      7-4      8-0      8-0
FTABLE   8-0     8-0#
GETPAS   6-1#    9-0
GWBUFS   6-1#
HALTC    6-2#    10-0
HRDCNT   6-1#
HRDERS   6-1#    7-4      7-4      8-0      10-0
HRDPAS   6-1#
ICONT    6-1#    7-4
ICOUNT   6-1#
IDNUM    6-1#
IISR     9-0#    11-0     11-0
INIT     6-1#
INQIN    7-4*    9-0      9-0*     9-0*     9-0*     12-0#
INQOUT   7-4*    9-0      9-0      9-0*     9-0*     12-0*
INTER    6-2#    9-0      9-0
INTLNK   7-4     8-0      11-0#
INTR     6-1#    7-4*
LAST     6-2#    7-4*     8-0      8-0*     10-0
LINK     6-2#    7-4      7-4      7-4*     7-4*     9-0
LOOP     7-4     7-4#     8-0      8-0
LULOOP   6-2#    7-4      7-4*     7-4*     7-4*
MAP22$   6-1#
MASK     6-2#    8-0      8-0      8-0*     8-0*     8-0*
MIXED    7-4     8-0#
MODE     6-2#    7-4      7-4      7-4*     7-4*     9-0      9-0      10-0
MODEM1   6-2#    9-0
MODEM2   6-2#    9-0
MODNAM   6-1#
MODSP    6-1     6-1#
MRAM     6-2#    9-0
MSG$     6-1#    7-4      7-4
MSGN$    6-1#    8-0      8-0
MSGS$    6-1#
MW       6-2#    9-0      9-0
```


CROSS REFERENCE TABLE (CREF V01-05)

```
NEED2    7-4     8-0#
NOP1     6-2#    9-0
NULL     6-1#    7-4      7-4      10-0
OISR     10-0#   11-0     11-0
OK       6-2#    10-0
OPEN     6-1     6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1
         6-1     6-1      6-1      6-1      6-1      6-1      6-1#
OTUA$    6-1#
OUTQIN   7-4*    10-0     10-0*    10-0*    10-0*    12-0#
OUTQOU   7-4*    10-0     10-0     10-0*    10-0*    12-0#
PA       6-2#    8-0      8-0      9-0      9-0      9-0      9-0
PASCNT   6-1#
PATCH    12-0#
PIRING   7-4     7-4      9-0      9-0      9-0      9-0      12-0#
PIROUT   7-4     7-4      10-0     10-0     10-0     10-0     12-0#
PIRQ$    6-1#    9-0      10-0
POPSP    6-1#
POPSP2   6-1#
PRTY     6-1#
PRTY0    6-1#
PRTY1    6-1#
PRTY2    6-1#
PRTY3    6-1#
PRTY4    6-1#
PRTY5    6-1     6-1      6-1#
PRTY6    6-1#
PRTY7    6-1#
PS       6-1#
PSW      6-1#
PUSH     6-1#
PUSH2    6-1#
PWRFLG   6-1#
R6       6-1#
R7       6-1#
RAND$    6-1#
RANNUM   6-1#
RCVBUF   7-4     8-0      9-0      11-0     11-0     12-0#
REC      9-0     9-0#
RES1     6-1#
RES2     6-1#
RESTRT   6-1     7-4      7-4      7-4#
RESUME   6-2#
RMODEM   6-2#    9-0
RRAM     6-2#    9-0      9-0
RSTRT    6-1#
SAR0     6-2#    7-4      7-4*
SAR1     6-2#    7-4      7-4*
SBADR    6-1#    8-0*
SCAN     7-4     8-0#
SELECT   6-2#    7-4      7-4      7-4      7-4      7-4*     8-0      8-0      8-0*
SETUP2   7-4     7-4#
SOFCNT   6-1#
SOFER$   6-1#    8-0
SOFPAS   6-1#
SOFT     8-0     8-0#
SOFT1    8-0     8-0#
SPOINT   6-1#
```

```
SPSIZ    1-23#    6-1
SR1      6-1#     7-4      7-4      7-4
SR2      6-1#
SR3      6-1#
SR4      6-1#     7-4
START    6-1      7-4#
STAT     6-1#
STOP     6-2#
SVR0     6-1#
SVR1     6-1#
SVR2     6-1#
SVR3     6-1#
SVR4     6-1#
SVR5     6-1#
SVR6     6-1#
SYSCNT   6-1#
TERM     6-2#
THRESH   6-2#     9-0      9-0
THRSH1   6-2#     9-0
THRSH2   6-2#     9-0
TIME     6-2#     9-0
TIMER    6-2#     9-0      9-0
TRPDFD   6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1
         6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1
         6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1#     6-1#     6-1#     6-1#     6-1#     6-1#
         6-1#     6-1#     6-1#     6-1#     6-1#     6-1#     6-1#     6-1#     6-1#     6-1#     6-1#     6-1#     6-1#
UPDATE   6-2#     9-0
VA       6-2#     8-0*     8-0*     9-0      9-0*     9-0*     9-0*
VECTOR   6-1#     7-4
WASADR   6-1#     8-0*
WDFR     6-1#     7-4*
WDTO     6-1#     7-4*
XBUF     8-0      9-0      12-0*
XDROP1   8-0      8-0*
XDROP2   8-0      8-0*
XERR     8-0      8-0      8-0*
XFLAG    6-1#
XMIT     9-0      9-0*
XX11     7-4*     8-0      11-0*
XX12    11-0*
XX13    11-0*
XX14    11-0*
XX15     8-0      11-0*
XX16     8-0      11-0*
XX21     7-4*     8-0      9-0      11-0*
XX22    11-0*
XX23    11-0*
XX24    11-0*
XX25     8-0      11-0*
XX26     8-0      11-0*
```



```
$BUFF    5-866#   11-1
$CKDAT   5-125#    8-0
$HUNG    5-920#    8-0
$IISR    5-510#    8-1
$LINK    5-840#   10-1
$MISC    5-4#      6-2
$OISR    5-762#    9-1
$SETUP   5-231#    7-4
BKMOD    1-111#
BREAK    1-201#    7-4      8-0
BTOD     1-220#
CKDATA   1-256#
DATACK   1-265#
DATERR   1-154#    8-0
DFSEVN   1-288#    6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1      6-1
DSEVNT   1-298#    6-1      6-1      6-1      6-1      6-1
END      1-191#    7-4      7-4
ENDIT    1-182#    8-0
ENDMOD   1-187#
EQUATS   1-306#    6-1
EXIT     1-136#    9-0      9-0      9-0      9-0      9-0      9-0      9-0      9-0      10-0     10-0     10-0     10-0
GETPA    1-247#    9-0
GWBUFF   1-235#
HRDER    1-144#    7-4      7-4      8-0      10-0
IOMOD    1-107#    6-1
IOMODP   1-131#
IOMODR   1-127#
IOMODX   1-123#
MAP22    1-251#
MODULE   1-24#     6-1
MSG      1-170#    7-4      7-4
MSGN     1-174#    8-0      8-0
MSGS     1-178#
NBKMOD   1-119#
OTOA     1-206#
PIRQ     1-195#    9-0      10-0
RAND     1-140#
SBKMOD   1-115#
SOFER    1-160#    8-0
```