IDENTIFICATION

PRODUCT CODE: AC-F858B-MC

PRODUCT NAME: CXDMRB0 DMR-11 MODULE

PRODUCT DATE: JUNE 1980

MAINTAINER: DEC/X11 SUPPORT GROUP

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.

```
                                        .TOMOD <DMRB >,1,1,5,5,0,25.,127
                                        MODULE  140000,DMRB ,1,1,5,5,0,25.,127
                                        .TITLE  DMRB DEC/X11 SYSTEM EXERCISER MODULE
                                ;       DDXCOM  VERSION  6      23-MAY-78
                                        .LIST   MIN
                                ;*****************************************************************************
000000                          BEGIN:
000000  104     115     122     MODNAM: .ASCII  /DMRB / ;MODULE NAME.
000003  102     040
000005  000                     XFLAG:  .BYTE   0PEN            ;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:    0PEN                    ;SWITCH REGISTER 1
000020  000000                  SR2:    0PEN                    ;SWITCH REGISTER 2
000022  000000                  SR3:    0PEN                    ;SWITCH REGISTER 3
000024  000000                  SR4:    0PEN                    ;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:   0PEN                    ;LOC TO SAVE R0.
000064  000000                  SVR1:   0PEN                    ;LOC TO SAVE R1.
000066  000000                  SVR2:   0PEN                    ;LOC TO SAVE R2.
000070  000000                  SVR3:   0PEN                    ;LOC TO SAVE R3.
000072  000000                  SVR4:   0PEN                    ;LOC TO SAVE R4.
000074  000000                  SVR5:   0PEN                    ;LOC TO SAVE R5.
000076  000000                  SVR6:   0PEN                    ;LOC TO SAVE R6.
000100  000000                  CSRA:   0PEN                    ;ADDR OF CURRENT CSR.
000102                          SBADR:                          ;ADDR OF GOOD DATA, OR
000102  000000                  ACSR:   0PEN                    ;CONTENTS OF CSR.
000104                          WASADR:                         ;ADDR OF BAD DATA, OR
000104  000000                  ASTAT:  0PEN                    ;STATUS REG CONTENTS.
000106                          ERRTYP:                         ;TYPE OF ERROR
000106  000000                  ASB:    0PEN                    ;EXPECTED DATA.
000110  000000                  AWAS:   0PEN                    ;ACTUAL DATA.
000112  000640'                 RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114  000000                  WDTO:   0PEN                    ;WORDS TO MEMORY PER ITERATION
000116  000000                  WDFR:   0PEN                    ;WORDS FROM MEMORY PER ITERATION
000120  000000                  INTR:   0PEN                    ;# 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                  SELEC1: 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                  LOLOOP: 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 = DIR
                                                                ;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 LO 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),ASIAT     ;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                      11SR:
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 RTI
                                    ;-----------------------------------------------------------
002760                      2$:
002760  017705  003370              MOV     @INQOUT,R5      ;GET LINK POINTER IN R5
002764  062767  000002  003362      ADD     #2,INQOUT       ;UPDATE QUEUE
002772  022767  006152' 003354      CMP     #PIRING+200,INQOUT ;END OF QUEUE?
003000  001003                      BNE     3$              ;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$:
004210  105711                          TSTB    (R1)            ;IS RQI GONE?
004212  100776                          BMI     1$              ;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      LAST    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      EXIT$ = 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*
BASEI     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#
BREAKS    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
```

.REM _

IDENTIFICATION

PRODUCT CODE: AC-FB18A-MC

PRODUCT NAME: CXDRJA0 DRV11-J MODULE

PRODUCT DATE: JANUARY 1981

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

DRJA IS AN IOMOD THAT EXERCISES ONE DRV11-J. THE MODULE USES A LOOP BACK CABLE TO CHECK DATA TRANSFERS TO AND FROM THE DRV11-J. IT TRANSMITS AND RECEIVES A SERIES OF WORST-CASE BUS PATTERNS AND ALSO TESTS THE ABILITY OF THE DRV11J TO GENERATE BOTH SEND AND RECEIVE INTERRUPTS.

2. REQUIREMENTS

HARDWARE: ONE DRV11-J WITH A BC05W-02 CABLE WHICH MUST BE INSTALLED WITH 1/2 TWIST BETWEEN THE 50 PIN CONNECTORS TO LOOPBACK OUTPUTS TO INPUTS.

STORAGE:: DRJA REQUIRES:
1. DECIMAL WORDS: 458
2. OCTAL WORDS: 712
3. OCTAL BYTES: 1624

3. PASS DEFINITION

ONE ITERATION OF THE DRJA MODULE CONSISTS OF SENDING AND RECEIVING 128 WORDS AND GENERATING 2 SEND AND 2 RECEIVE INTERRUPTS.

ONE PASS IS EQUAL TO 900. ITERATIONS.

4. EXECUTION TIME

ONE PASS OF DRJA RUNNING ALONE ON A PDP11/03 PROCESSOR TAKES APPROXIMATELY 25 SECONDS.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 164160, VECTOR: 300, BRI: 4, DEVCNI: 1

REQUIRED PARAMETERS:

AT CONFIGURATION TIME MODIFY "VECTOR" IF SYSTEM SPECIFIES OTHER THAN 300

THE DRV11-J HAS PROGRAMABLE VECTORS AND AT RUN TIME THE DRJA MODULE WILL SELECT THE CONFIGURED VECTOR AND SETUP 4 VECTORS IN INCREMENTS ABOVE THE SELECTED VECTOR.
EX: 300,304,310,314

6. DEVICE/OPTION SET-UP

CONNECT THE BC05W-02 CABLE WITH 1/2 TWIST BETWEEN THE 50 PIN CONNECTORS TO TIE OUTPUT BACK TO INPUT.

7. MODULE OPERATION

TEST SEQUENCE:

A. SET UP VECTORS AND ADDRESS POINTER
B. OUTPUT TEST DATA TO OUTPUT BUFFER
C. COMPARE OUTPUT BUFFER WITH TEST DATA-REPORT ANY DATA ERROR
D. COMPARE INPUT BUFFER WITH TEST DATA-REPORT ANY DATA ERROR
E. IF NOT 128 TRANSFERS, NEXT TEST DATUM--REPEAT B-D
F. IF FINISHED GENERATE AND TEST INPUT/OUTPUT INTERRUPTS.
G. IF NO INTERRUPT - DO NOT REPORT END PASS
H. IF INTERRUPT - REPORT END PASS RESTART AT A

8. OPERATION OPTIONS

NONE

9. NON-STANDARD PRINTOUTS

NONE: ALL PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN THE DEC/X11 DOCUMENT

```
                ;DRV11-J DEC/X11 EXERCISER MODULE
000000'                 IOMOD   <DRJA >,164160,300,4,,,900.,0
000000'                 MODULE  140000,DRJA ,164160,300,4,,,900.,0
                        .TITLE  DRJA DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6      23-MAY-78
                                .LIST   BIN
                                ;*****************************************************************************
000000'                         BEGIN:
000000' 051104  040512      040 MODNAM: .ASCII  /DRJA / ;MODULE NAME.
000005'    000                  XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 164160                  ADDR:   164160+0                ;1ST DEVICE ADDR.
000010' 000300                  VECTOR: 300+0                   ;1ST DEVICE VECTOR.
000012'    200                  BR1:    .BYTE   PRTY4+0         ;1ST BR LEVEL.
000013'    000                  BR2:    .BYTE   PRTY+0          ;2ND BR LEVEL.
000014' 000001                  DVID1:  +1                      ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN                    ;SWITCH REGISTER 4
                                ;*****************************************************************************
000026' 140000                  STAT:   140000                  ;STATUS WORD.
000030' 000262'                 INIT:   START                   ;MODULE START ADDR.
000032' 000224'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                       ;PASS COUNTER.
000036' 001604                  ICONT:  900.                    ;# OF ITERATIONS PER PASS=900.
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                  ASR:    OPEN                    ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN                    ;ACTUAL DATA.
000112' 000304'                 RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122' 000000                  IDNUM:  0                       ;MODULE IDENTIFICATION NUMBER=0
```

```
                000040                          .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                                .NLIST
                                                .WORD   0
                                                .LIST
                                                .ENDR
        000224'                         MODSP:
                                        ;*****************************************************************************
    513                                 ;
    514                                 ;
    515                                 ;
    516                                 ;THIS MODULE TESTS THE DRV11-J  GENERAL DEVICE INTERFACE
    517                                 ;BC05W-02 CABLE MUST BE INSTALLED WITH A 1/2 TWIST BETWEEN THE
    518                                 ;50 PIN CONNECTORS TO LOOPBACK OUTPUTS TO INPUTS.
    519                                 ;
    520
    521                                 ;DRV11-J BUS REGISTER ADDRESS POINTERS
    522 000224' 164160                  DRVJCA: 164160
    523 000226' 164162                  DRVJBA: 164162
    524
    525 000230' 164164                  DRVJCB: 164164
    526 000232' 164166                  DRVJBB: 164166
    527
    528 000234' 164170                  DRVJCC: 164170
    529 000236' 164172                  DRVJBC: 164172
    530
    531 000240' 164174                  DRVJCD: 164174
    532 000242' 164176                  DRVJBD: 164176
    533
    534
    535                                 ;DRV11-J VECTOR ADDRESS POINTERS
    536 000244' 000300                  DRVECA: 300
    537 000246' 000304                  DRVECB: 304
    538 000250' 000310                  DRVECC: 310
    539 000252' 000314                  DRVECD: 314
    540
    541                                 ;LOCAL VARIABLES
    542 000254' 177777                  ALLON:  -1
    543 000256' 000000                  ALLOFF: 0
    544 000260' 177776                  BUBBLE: 177776
    545                                 ;
    546                                 ;
    547                                 ;
    548                                 ;
    549                                 ;
    550                                 ;INITIALIZATION FOR GENERAL DEVICE INTERFACE
    551
    552
    553 000262' 012767  000004  177630  START:  MOV     #4.,INTR            ;4 INTERRUPTS PER ITERATION
    554 000270' 012767  000200  177616          MOV     #128.,WDTO          ;128. WDS TO MEM PER ITERATION
    555 000276' 012767  000200  177612          MOV     #128.,WDFR          ;128. WDS FROM MEM PER ITERATION
    556
    557 000304' 012700  000224'         RESTRT: MOV     #DRVJCA,R0          ;SET UP REGS ADDR POINTER
    558 000310' 016701  177472                  MOV     ADDR,R1             ;GET BASE ADDRESS
    559 000314' 010120                  SETUP2: MOV     R1,(R0)+            ;LOAD THEM
    560 000316' 062701  000002                  ADD     #2,R1               ;GET NEXT READY
    561 000322' 022700  000244'                 CMP     #DRVJBD+2,R0        ;DONE?
```

```
                                        BNE     SETUP2                  ;BR IF NOT
                                        MOV     #DRVECA,R0              ;SET UP VECTOR ADDRESS POINTER
                                        MOV     VECTOR,R1               ;GET BASE VECTOR ADDRESS
                                SETUP3: MOV     R1,(R0)+
                                        ADD     #4,R1                   ;POINT TO NEXT
                                        CMP     #DRVECD+2,R0            ;ALL DONE?
                                        BNE     SETUP3                  ;BR IF NOT
                                        MOV     DRVJCA,R5               ;STORE CSRA IN R5
                                        MOV     R5,CSRA
                                        MOV     DRVECA,R0               ;SETUP INTERRUPT SERVICE
                                        MOV     #DRASRV,(R0)+           ;RPLY A SERVICE ROUTINE
                                        MOV     BR1,(R0)+               ;STORE PRIORITY
                                        MOV     #DRBSRV,(R0)+           ;RPLY B SERVICE ROUTINE
                                        MOV     BR1,(R0)+               ;STORE PRIORITY
                                        MOV     #DRCSRV,(R0)+           ;RPLY C SERVICE ROUTINE
                                        MOV     BR1,(R0)+               ;STORE PRIORITY
                                        MOV     #DRDSRV,(R0)+           ;RPLY D SERVICE ROUTINE
                                        MOV     BR1,(R0)+               ;STORE PRIORITY
                                        CLR     (R5)                    ;CLEAR CSRA AND INT. ENABLE
                                        CLR     10(R5)                  ;CLEAR CSRC,SET INPUT MODE
                                        CLRB    15(R5)                  ;CLEAR CSRD,SET INPUT MODE
000442' 112765  000001  000001          MOVB    #BIT0,1(R5)             ;SET CSRA INTO OUTPUT MODE
000450' 112765  000001  000005          MOVB    #BIT0,5(R5)             ;SET CSRB INTO OUTPUT MODE
000456' 016701  177572                  MOV     ALLON,R1                ;SET UP INITIAL TEST PATTERNS; 177777
000462' 016702  177572                  MOV     BUBBLE,R2               ;  AND 177776 (0 WILL BUBBLE TO LEFT)
000466' 010103                          MOV     R1,R3                   ;R3 REMEMBERS ALLON/ALLOFF WHEN BUBBLE IS IN USE
000470' 010104                          MOV     R1,R4                   ;C-BIT SET WORD FOR BUBBLE ROL


                                        ;CHECK DATA TRANSFER ON DRV11-J
                                        ;       TRANSMIT, RECIEVE AND CHECK (VIA BC05W-02 CABLE) A SERIES OF
                                        ;       64 WORD PATTERNS FORCING 'WORST-CASE' TRANSITIONS.  THIS
                                        ;       CONSISTS OF 177777 FOLLOWED BY 177776, 177777, 177775, 177777,
                                        ;       177773, 177777, 177767, ETC (ALL ONES ALTERNATING WITH A
                                        ;       0 BUBBLING LEFT THROUGH A WORD OF ONES), THEN ALL ZEROS ALTERN-
                                        ;       ATING WITH THE SAME BUBBLE PATTERN.


000472' 010567  177402          DRACT1: MOV     R5,CSRA                 ;SAVE CSR ADDRESS
000476' 010165  000002                  MOV     R1,2(R5)                ;MOVE DATA TO OUTPUT BUFFER DBRA
000502' 020165  000002                  CMP     R1,2(R5)                ;CHECK DATA AT DBRA
000506' 001421                          BEQ     1$                      ;BRANCH IF DATA GOOD
000510' 012767  177701  177364          MOV     #177701,$BADR           ;GOOD DATA ADDRESS
000516' 062705  000002                  ADD     #2,     R5              ;ADD 2 TO GET DATA REG. ADDRESS
000522' 010567  177356                  MOV     R5,     $WASADR         ;BAD DATA ADDRESS
000526' 162705  000002                  SUB     #2,     R5              ;RESTORE R5
000532' 010167  177350                  MOV     R1,$SB                  ;MOVE 'SHOULD BE'
000536' 016567  000002  177344          MOV     2(R5),$WAS              ;MOVE 'WAS'
                                        ;***************************************************************
(1) 000544' 104404  000000'             DATERR,BEGIN            ;DATA ERROR!!!
(1)                                     ;***************************************************************
000550' 000506                          BR      4$                      ;NEXT DATA
000552' 016767  177452  177320  1$:     MOV     DRVJCB,CSRA             ;SAVE CSRB ADDRESS
000560' 010165  000006                  MOV     R1,6(R5)                ;MOVE DATA TO OUTPUT BUFFER DBRB
000564' 020165  000006                  CMP     R1,6(R5)                ;CHECK DATA AT DBRB
```

```
                                BEQ     2$                      ;BRANCH IF DATA GOOD
000570' 001421                  MOV     #177701,SBADR           ;GOOD DATA ADDRESS
000572' 012767  177701  177302   ADD     #6,     R5              ;ADD 2 TO GET DATA REG. ADDRESS
000600' 062705  000006          MOV     R5,     WASADR          ;BAD DATA ADDRESS
000604' 010567  177274          SUB     #6,     R5              ;RESTORE R5
000610' 162705  000006          MOV     R1,ASB                  ;MOVE 'SHOULD BE'
000614' 010167  177266          MOV     6(R5),AWAS              ;MOVE 'WAS'
000620' 016567  000006  177262  ;*****************************************************************
                                DATER$,BEGIN            ;DATA ERROR!!!
(1)  000626' 104404  000000'    ;*****************************************************************
(1)
                                BR      4$                      ;NEXT DATA
000632' 000455                  MOV     DRVJCC,CSRA             ;SAVE CSRC ADDRESS
000634' 016767  177374  177236  2$:     CMP     R1,12(R5)       ;CHECK RECEIVED DATA AT DBRC
000642' 020165  000012          BEQ     3$                      ;BRANCH IF DATA GOOD
000646' 001421                  MOV     #177701,SBADR           ;GOOD DATA ADDRESS
000650' 012767  177701  177224   ADD     #12,    R5              ;GET INPUT DATA ADDRESS
000656' 062705  000012          MOV     R5,     WASADR          ;BAD DATA ADDRESS
000662' 010567  177216          SUB     #12,    R5              ;RESTORE R5
000666' 162705  000012          MOV     R1,ASB                  ;MOVE 'SHOULD BE'
000672' 010167  177210          MOV     12(R5),AWAS             ;MOVE 'WAS'
000676' 016567  000012  177204  ;*****************************************************************
                                DATER$,BEGIN            ;DATA ERROR!!!
(1)  000704' 104404  000000'    ;*****************************************************************
(1)
                                BR      4$                      ;NEXT DATA
000710' 000426                  MOV     DRVJCD,CSRA             ;SAVE CSRD ADDRESS
000712' 016767  177322  177160  3$:     CMP     R1,16(R5)       ;CHECK RECEIVED DATA AT DBRD
000720' 020165  000016          BEQ     4$                      ;BRANCH IF DATA GOOD
000724' 001420                  MOV     #177701,SBADR           ;GOOD DATA ADDRESS
000726' 012767  177701  177146   ADD     #16,    R5              ;GET INPUT DATA ADDRESS
000734' 062705  000016          MOV     R5,     WASADR          ;BAD DATA ADDRESS
000740' 010567  177140          SUB     #16,    R5              ;RESTORE R5
000744' 162705  000016          MOV     R1,ASB                  ;MOVE 'SHOULD BE'
000750' 010167  177132          MOV     16(R5),AWAS             ;MOVE 'WAS'
000754' 016567  000016  177126  ;*****************************************************************
                                DATER$,BEGIN            ;DATA ERROR!!!
(1)  000762' 104404  000000'    ;*****************************************************************
(1)

                        ;CHECK CURRENT DATA PATTERN AND LOAD NEXT ACCORDINGLY
                        ;
000766' 020167  177262  4$:     CMP     R1,ALLON                ;IF JUST TESTED WITH
000772' 001405                  BEQ     5$                      ;  ALL ON/OFF PATTERN,
000774' 020167  177256          CMP     R1,ALLOFF               ;     LOAD THE CURRENT BUBBLE
001000' 001402                  BEQ     5$                      ;       FOR NEXT TEST CYCLE
001002' 010301                  MOV     R3,R1                   ;NOT BUBBLE; LOAD CURRENT ALL ON/OFF
001004' 000632                  BR      DRACT1                  ;TEST WITH IT
001006' 010201                  MOV     R2,R1                   ;SET UP BUBBLE FOR THIS CYCLE
001010' 006304          5$:     ASL     R4                      ;PRE-LOAD C-BIT FOR ROL
001012' 006102                  ROL     R2                      ;BUBBLE 0 TO LEFT ONE BIT
001014' 020267  177240          CMP     R2,BUBBLE               ;HAS IT CIRCLED ALL THE WAY?
001020' 001401                  BEQ     6$                      ;YES: SWITCH ALLON-ALLOFF OR END
001022' 000623                  BR      DRACT1                  ;TEST WITH THE BUBBLE
001024' 005703          6$:     TST     R3                      ;ALLON OR ALLOFF?
001026' 001405                  BEQ     INTEST                  ;IF ALLOFF, FINISHED: TEST INTERRUPTS
001030' 010304                  MOV     R3,R4                   ;RESET C-BIT PRE-LOAD
001032' 016703  177220          MOV     ALLOFF,R3               ;SWITCH TO ALLOFF
001036' 010301                  MOV     R3,R1                   ;SET UP FOR TEST
```

```
001040' 000614                          BR      DRACT1                  ;THEN DO IT

                                        ;CHECK INTERRUPTS ON DRV11-J

001042' 016705  177156          INTEST: MOV     DRVJCA,R5               ;R5 = CSRA ADDRESS
001046' 016767  177166  177030          MOV     DRVJCD,WASADR           ;SAVE CSRD ADDRESS WHICH WILL BE
                                                                        ;USED TO CHECK ISR REGISTER
001054' 105015                          CLRB    (R5)                    ;CHIP RESET GROUP 1
001056' 105065  000010                  CLRB    10(R5)                  ;CHIP RESET GROUP 2
                                                                        ;IMR SET TO ALL ONES
                                                                        ;ISR,IRR,ACR AND MODE
                                                                        ;REGISTER ALL CLEARED
001062' 016702  177156                  MOV     DRVECA,R2               ;STORE FIRST VECTOR TO
                                                                        ;PROGRAM ON THE DRV11-J.
001066' 006002                          ROR     R2
001070' 006002                          ROR     R2                      ;JUSTIFY THE VECTOR
                                                                        ;FOR THE DDRV11-J VECTOR MEMORY
001072' 012703  000344                  MOV     #344,R3                 ;START WITH BYTE COUNT = 0
                                                                        ;AND LEVEL 4 FOR RPLY A.
001076' 012701  000004                  MOV     #4,R1                   ;COUNTER FOR 4 VECTORS.
001102' 110365  000010          1$:     MOVB    R3,10(R5)               ;PRESET VECTOR MEMORY IN CSRC
001106' 110265  000014                  MOVB    R2,14(R5)               ;WRITE VECTOR INTO VECTOR MEMORY
                                                                        ;THROUGH CSRD
001112' 005203                          INC     R3                      ;INCREMENT IRR LEVEL
001114' 005202                          INC     R2                      ;INCREMENT VECTOR
001116' 005301                          DEC     R1                      ;FINISHED 4 VECTORS?
001120' 001370                          BNE     1$                      ;DO VECTORS FOR RPLY A,B,C,D
001122' 112765  000260  000010          MOVB    #260,10(R5)             ;PRESELECT IMR FOR WRITING
001130' 112765  000017  000014          MOVB    #17,14(R5)              ;WRITE GROUP2 IMR THROUGH CSRD
                                                                        ;TO CLEAR MASK BITS FOR RPLY A,B,C,D
001136' 112715  000241                  MOVB    #241,(R5)               ;ARM GROUP 1 CHIP TO PASS ENABLE
                                                                        ;TO GROUP2.
001142' 112765  000241  000010          MOVB    #241,10(R5)             ;ARM GROUP2 TO ALLOW INTERRUPTS

                                        ;WRITE DATA INTO OUTPUT BUFFER DBRA AND CAUSE INTERRUPT
                                        ;AT IRR6 RPLY C

001150' 012765  177777  000002  TS1NTC: MOV     #-1,2(R5)               ;WRITE INTO OUTPUT BUFFER
001156' 052715  001000                  BIS     #BIT9,(R5)              ;SET INTERRUPT ENABLE
001162' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
001166' 112765  000240  000010  CHECKC: MOVB    #240,10(R5)             ;RETURN FOR RPLY C
                                                                        ;INTERRUPT CHECK.
                                                                        ;SETUP TO READ ISR REGISTER
001174' 012701  000100                  MOV     #100,R1                 ;ISR 6 EXPECTED
001200' 116567  000014  176702          MOVB    14(R5),AWAS             ;READ ISR REGISTER THROUGH CSRD
001206' 020167  176676                  CMP     R1,AWAS                 ;COMPARE ISR DATA WITH EXPECTED
001212' 001415                          BEQ     TS1NTA                  ;BRANCH IF GOOD DATA
001214' 016767  177014  176656          MOV     DRVJCC,CSRA             ;SAVE CSRC FOR ERROR PRINT
001222' 012767  177701  176652          MOV     #177701,SBADR           ;GOOD DATA ADDRESS
001230' 016767  177004  176646          MOV     DRVJCD,WASADR           ;BAD DATA ADDRESS
001236' 010167  176644                  MOV     R1,ASB                  ;SHOULD BE DATA
                                        ;*****************************************************************
(1) 001242' 104404  000000'             DATER$,BEGIN            ;DATA ERROR!!!
(1)                                     ;*****************************************************************

```

```
                                ;READ DATA FROM INPUT BUFFER DBRC AND CAUSE INTERRUPT
                                ;AT IRR4 RPLY A
                                
001246' 012765  000160  000010  TSINTA: MOV     #160,10(R5)             ;CLEAR THE ISR REGISTER
001254' 016502  000012                  MOV     12(R5),R2               ;READ DBRC INPUT BUFFER
001260' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
001264' 112765  000240  000010  CHECKA: MOVB    #240,10(R5)             ;RETURN FOR RPLY A
                                                                        ;INTERRUPT CHECK.
                                                                        ;SETUP TO READ ISR REGISTER
001272' 012701  000020                  MOV     #20,R1                  ;ISR 4 EXPECTED
001276' 116567  000014  176604          MOVB    14(R5),AWAS             ;READ ISR REGISTER THROUGH CSRD
001304' 020167  176600                  CMP     R1,AWAS                 ;COMPARE ISR DATA WITH EXPECTED
001310' 001415                          BEQ     TSINTD                  ;BRANCH IF GOOD DATA
001312' 016767  176706  176560          MOV     DRVJCA,CSRA             ;SAVE CSRA FOR ERROR PRINT
001320' 012767  177701  176554          MOV     #177701,SBADR           ;GOOD DATA ADDRESS
001326' 016767  176706  176550          MOV     DRVJCD,WASADR           ;BAD DATA ADDRESS
001334' 010167  176546                  MOV     R1,ASB                  ;SHOULD BE DATA
                                        ;*****************************************************************
(1) 001340' 104404  000000'             DATERS,BEGIN            ;DATA ERROR!!!
(1)                                     ;*****************************************************************

                                ;WRITE DATA INTO OUTPUT BUFFER DBRB AND CAUSE INTERRUPT
                                ;AT IRR7 RPLY D

001344' 012765  000160  000010  TSINTD: MOV     #160,10(R5)             ;CLEAR THE ISR REGISTER
001352' 012765  177777  000006          MOV     #-1,6(R5)               ;WRITE INTO OUTPUT BUFFER DBRB
001360' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
001364' 112765  000240  000010  CHECKD: MOVB    #240,10(R5)             ;RETURN FOR RPLY D
                                                                        ;INTERRUPT CHECK.
                                                                        ;SETUP TO READ ISR REGISTER
001372' 012701  000200                  MOV     #200,R1                 ;ISR 7 EXPECTED
001376' 116567  000014  176504          MOVB    14(R5),AWAS             ;READ ISR REGISTER THROUGH CSRD
001404' 020167  176500                  CMP     R1,AWAS                 ;COMPARE ISR DATA WITH EXPECTED
001410' 001415                          BEQ     TSINTB                  ;BRANCH IF GOOD DATA
001412' 016767  176622  176460          MOV     DRVJCD,CSRA             ;SAVE CSRD FOR ERROR PRINT
001420' 012767  177701  176454          MOV     #177701,SBADR           ;GOOD DATA ADDRESS
001426' 016767  176606  176450          MOV     DRVJCD,WASADR           ;BAD DATA ADDRESS
001434' 010167  176446                  MOV     R1,ASB                  ;SHOULD BE DATA
                                        ;*****************************************************************
(1) 001440' 104404  000000'             DATERS,BEGIN            ;DATA ERROR!!!
(1)                                     ;*****************************************************************

                                ;READ DATA FROM INPUT BUFFER DBRD AND CAUSE INTERRUPT
                                ;AT IRR5 RPLY B.

001444' 012765  000160  000010  TSINTB: MOV     #160,10(R5)             ;CLEAR THE ISR REGISTER
001452' 016503  000016                  MOV     16(R5),R3               ;READ INPUT BUFFER DBRD
001456' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
001462' 112765  000240  000010  CHECKB: MOVB    #240,10(R5)             ;RETURN FOR RPLY B
                                                                        ;INTERRUPT CHECK.
                                                                        ;SETUP TO READ ISR REGISTER
001470' 012701  000040                  MOV     #40,R1                  ;ISR 5 EXPECTED
001474' 116567  000014  176406          MOVB    14(R5),AWAS             ;READ ISR REGISTER THROUGH CSRD
001502' 020167  176402                  CMP     R1,AWAS                 ;COMPARE ISR DATA WITH EXPECTED
001506' 001415                          BEQ     ENPS                    ;BRANCH IF GOOD DATA
001510' 016767  176514  176362          MOV     DRVJCD,CSRA             ;SAVE CSRD FOR ERROR PRINT
```

```
   772  001516' 012767  177701  176356          MOV     #177701,SBADR           ;GOOD DATA ADDRESS
   773  001524' 016767  176510  176352          MOV     DRVJCB,WASADR           ;BAD DATA ADDRESS
   774  001532' 010167  176350                  MOV     R1,ASB                  ;SHOULD BE DATA
   775                                          ;****************************************************************
   (1)  001536' 104404  000000'                 DATERR,BEGIN            ;DATA ERROR!!!
   (1)                                          ;****************************************************************
   776
   777
   778  001542' 042715  001000          ENPS:   BIC     #BIT9,  (R5)            ;CLEAR INTERRUPT ENABLE
   779  001546' 104413  000000'                 ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
   (1)                                                                  ;MONITOR SHALL TEST END OF PASS
   780  001552' 000167  176526                  JMP     RESTRT          ;BACK TO BEGINNING
   781
   782
   783                                  ;INPUT/OUTPUT SERVICE ROUTINES
   784
   785
   786  001556'                         DRASRV:
   (1)                                          ;---------------------------------------------------------------
   (1)  001556' 000004  000000' 001564'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
   (1)                                          ;---------------------------------------------------------------
   787  001564' 000167  177474          1$:     JMP     CHECKA          ;CHECK RPLY A (ISR 4) IN THE ISR REGISTER
   788
   789  001570'                         DRBSRV:
   (1)                                          ;---------------------------------------------------------------
   (1)  001570' 000004  000000' 001576'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
   (1)                                          ;---------------------------------------------------------------
   790  001576' 000167  177660          1$:     JMP     CHECKB          ;CHECK RPLY B (ISR 5) IN THE ISR REGISTER
   791
   792  001602'                         DRCSRV:
   (1)                                          ;---------------------------------------------------------------
   (1)  001602' 000004  000000' 001610'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
   (1)                                          ;---------------------------------------------------------------
   793  001610' 000167  177352          1$:     JMP     CHECKC          ;CHECK RPLY C (ISR 6) IN THE ISR REGISTER
   794
   795  001614'                         DRDSRV:
   (1)                                          ;---------------------------------------------------------------
   (1)  001614' 000004  000000' 001622'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
   (1)                                          ;---------------------------------------------------------------
   796  001622' 000167  177536          1$:     JMP     CHECKD          ;CHECK RPLY D (ISR 7) IN THE ISR REGISTER
   797
   798          000001                          .END
```

```
ACSR    000102R          512#
ADDR    000006R          512#    558
ADDR22= 001000           512#
ALLOFF  000256R          543#    651     664
ALLON   000254R          542#    585     649
ASB     000106R          512#    609*    621*    632*    643*    717*    736*    755*    774*
ASTAT   000104R          512#
AWAS    000110R          512#    610*    622*    633*    644*    711*    712     730*    731     749*    750     768*    769
BEGIN   000000R          512#    611     623     634     645     706     718     725     737     744     756     763     775
                         779     786     789     792     795
BIT0  = 000001           512#    583     584
BIT1  = 000002           512#
BIT10 = 002000           512#
BIT11 = 004000           512#
BIT12 = 010000           512#
BIT13 = 020000           512#
BIT14 = 040000           512#
BIT15 = 100000           512#
BIT2  = 000004           512#
BIT3  = 000010           512#
BIT4  = 000020           512#
BIT5  = 000040           512#
BIT6  = 000100           512#
BIT7  = 000200           512#
BIT8  = 000400           512#
BIT9  = 001000           512#    705     778
BREAK$= 104407           512#
BR1     000012R          512#    573     575     577     579
BR2     000013R          512#
BTOD$ = 104421           512#
BUBBLE  000260R          544#    586     658
CDATA$= 104412           512#
CHECKA  001264R          726#    787
CHECKB  001462R          764#    790
CHECKC  001166R          707#    793
CHECKD  001364R          745#    796
CONFIG  000056R          512#
CSRA    000100R          512#    570*    601*    613*    625*    636*    714*    733*    752*    771*
DATCK$= 104411           512#
DATER$= 104404           512#    611     623     634     645     718     737     756     775
DRACT1  000472R          601#    654     660     666
DRASRV  001556R          572     786#
DRBSRV  001570R          574     789#
DRCSRV  001602R          576     792#
DRDSRV  001614R          578     795#
DRVECA  000244R          536#    563     571     679
DRVECB  000246R          537#
DRVECC  000250R          538#
DRVECD  000252R          539#    567
DRVJBA  000226R          523#
DRVJBB  000232R          526#
DRVJBC  000236R          529#
DRVJBD  000242R          532#    561
DRVJCA  000224R          522#    557     569     671     733
DRVJCB  000230R          525#    613     771
DRVJCC  000234R          528#    625     714
```

```
DRVJCD  000240R          531#    636     672     716     735     752     754     773
DV1D1   000014R          512#
ENDITS= 104413           512#    779
ENDS    = 104410         512#
ENPS    001542R          770     778#
ERRTYP  000106R          512#
EXITS   = 104400         512#    706     725     744     763
GETPAS= 104415           512#
GWBUFS= 104414           512#
HRDCNT  000044R          512#
HRDERS= 104405           512#
HRDPAS  000050R          512#
ICONT   000036R          512#
ICOUNT  000040R          512#
IDNUM   000122R          512#
INIT    000030R          512#
INTEST  001042R          662     671#
INTR    000120R          512#    553*
MAP22S= 104416           512#
MODNAM  000000R          512#
MODSP   000224R          512#
MSGNS   = 104403         512#
MSGSS   = 104402         512#
MSGS    = 104401         512#
NULL    = 000000         512#
OPEN    = 000000         512#
OTOAS   = 104420         512#
PASCNT  000034R          512#
PIRQS   = 000004         512#    786     789     792     795
POPSP   = 005726         512#
POPSP2= 022626           512#
PRTY    = 000000         512#
PRTY0   = 000000         512#
PRTY1   = 000040         512#
PRTY2   = 000100         512#
PRTY3   = 000140         512#
PRTY4   = 000200         512#
PRTY5   = 000240         512#
PRTY6   = 000300         512#
PRTY7   = 000340         512#
PS      = 177776         512#
PSW     = 177776         512#
PUSH    = 005746         512#
PUSH2   = 024646         512#
RANDS   = 104417         512#
RANNUM  000054R          512#
RESIRT  000304R          512     557#    780
RES1    000056R          512#
RES2    000060R          512#
RSTRT   000112R          512#
SBADR   000102R          512#    605*    617*    628*    639*    715*    734*    753*    772*
SETUP2  000314R          559#    562
SETUP3  000340R          565#    568
SOFCNT  000042R          512#
SOFERS= 104406           512#
SOFPAS  000046R          512#
```

```
SPOINT  000032R          512#
SPSIZ = 000040             7#     512
SR1     000016R          512#
SR2     000020R          512#
SR3     000022R          512#
SR4     000024R          512#
START   000262R          512      553#
STAT    000026R          512#
SVR0    000062R          512#
SVR1    000064R          512#
SVR2    000066R          512#
SVR3    000070R          512#
SVR4    000072R          512#
SVR5    000074R          512#
SVR6    000076R          512#
SYSCNT  000052R          512#
TRPDFD= 000022           512#
TSINTA  001246R          713      723#
TSINTB  001444R          751      761#
TSINTC  001150R          704#
TSINTD  001344R          732      742#
VECTOR  000010R          512#     564
WASADR  000104R          512#     607*    619*    630*    641*    672*    716*    735*    754*    773*
WDFR    000116R          512#     555*
WDTO    000114R          512#     554*
XFLAG   000005R          512#
```

```
BKMOD      95#
BREAK     185#
BTOD      204#
CKDATA    240#
DATACK    249#
DATERR    138#    611     623     634     645     718     737     756     775
DFSEVN    272#    512
DSEVNT    282#    512
END       175#
ENDIT     166#    779
ENDMOD    171#
EQUATS    288#    512
EXIT      120#    706     725     744     763
GETPA     231#
GWBUFF    219#
HRDER     128#
IOMOD      91#    512
IOMODP    115#
IOMODR    111#
IOMODX    107#
MAP22     235#
MODULE      8#    512
MSG       154#
MSGN      158#
MSGS      162#
NBKMOD    103#
OTOA      190#
PIRQ      179#    786     789     792     795
RAND      124#
SBKMOD     99#
SOFER     144#


. ABS.  000000      000
        001626      001


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DSK2:XDRJA0,DSK2:XDRJA0/CRF=DDXCOM.P11,XDRJA0.P11
 RUN-TIME: 3 4 .6 SECONDS
 RUN-TIME RATIO: 45/9=5.0
 CORE USED:  7K  (13 PAGES)
```

.REM_

IDENTIFICATION

PRODUCT CODE: AC-F808B-MC

PRODUCT NAME: CXDMEB0 DMP/DMV11 DECX SLV MOD

PRODUCT DATE: AUGUST 1981

MAINTAINER: DIAGNOSTIC ENGINEERING CC:38P

AUTHORS: DAVID HOFFMAN
CHRIS BRIENEN

1. ABSTRACT

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

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

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

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

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

2. REQUIREMENTS

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

3. PASS DEFINITION

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

4. EXECUTION TIME

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

5. CONFIGURATION PARAMETERS

DEFAULT PARAMETERS :

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

SOFTWARE SWITCH REGISTER OPTIONS :

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

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

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

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

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

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

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

6. DEVICE/OPTION SETUP

DME REQUIRES NO ADDITIONAL DEVICE SETUP.

7. DME MODULE OPERATION

TEST SEQUENCE:

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

8. OPERATION OPTIONS

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

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

9. ERROR REPORTS

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

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

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

```
                                        .TITLE  DMEB DEC/X11 SYSTEM EXERCISER MODULE
                                ;       DDXCOM  VERSION  6      23-MAY-78
                                        .LIST   BIN
                                ;*****************************************************************************
000000'                         BEGIN:
000000' 046504  041105      040 MODNAM: .ASCII  /DMEB / ;MODULE NAME.
000005'     000                 XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 160170                  ADDR:   160170+0                ;1ST DEVICE ADDR.
000010' 000300                  VECTOR: 300+0                   ;1ST DEVICE VECTOR.
000012'     240                 BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013'     240                 BR2:    .BYTE   PRTY5+0         ;2ND BR LEVEL.
000014' 000001                  DVID1:  0+1                     ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN                    ;SWITCH REGISTER 4
                                ;*****************************************************************************
000026' 150000                  STAT:   150000                  ;STATUS WORD.
000030' 000466'                 INIT:   START                   ;MODULE START ADDR.
000032' 000252'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                       ;PASS COUNTER.
000036' 000001                  ICONT:  1                       ;# OF ITERATIONS PER PASS=1
000040' 000000                  ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042' 000000                  SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                  HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                  SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                  HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                  SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054' 000000                  RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:                         ;RESERVED FOR MONITOR USE
000056' 000000                  RES1:   0                       ;RESERVED FOR MONITOR USE
000060' 000000                  RES2:   0                       ;RESERVED FOR MONITOR USE
000062' 000000                  SVR0:   OPEN                    ;LOC TO SAVE R0.
000064' 000000                  SVR1:   OPEN                    ;LOC TO SAVE R1.
000066' 000000                  SVR2:   OPEN                    ;LOC TO SAVE R2.
000070' 000000                  SVR3:   OPEN                    ;LOC TO SAVE R3.
000072' 000000                  SVR4:   OPEN                    ;LOC TO SAVE R4.
000074' 000000                  SVR5:   OPEN                    ;LOC TO SAVE R5.
000076' 000000                  SVR6:   OPEN                    ;LOC TO SAVE R6.
000100' 000000                  CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102'                         SBADR:                          ;ADDR OF GOOD DATA, OR
000102' 000000                  ACSR:   OPEN                    ;CONTENTS OF CSR.
000104'                         WASADR:                         ;ADDR OF BAD DATA, OR
000104' 000000                  ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106'                         ERRTYP:                         ;TYPE OF ERROR
000106' 000000                  ASB:    OPEN                    ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN                    ;ACTUAL DATA.
000112' 002224'                 RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122' 000000                  IDNUM:  0                       ;MODULE IDENTIFICATION NUMBER=0
000124' 006244'                 RBUFVA: BUFIN                   ;READ BUFFER VIRTUAL ADDRESS
000126' 000000                  RBUFPA: OPEN                    ;READ BUFFER PHYSICAL ADDRESS
000130' 000000                  RBUFEA: OPEN                    ;READ BUFFER EA BITS
```

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

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


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

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

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

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

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


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



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

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


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


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


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

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


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


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


                        ;*****************************************************************************
                        ;* MODE DEFINITION COMMAND - BSEL6
                        ;*****************************************************************************
000000                  HDPPDC  = 0
000001                  FDPPDC  = 1
000002                  HDPP    = 2
000003                  FDPP    = 3
000004                  HDCS    = 4
000005                  FDCS    = 5
000006                  HDTS    = 6
000007                  FDTS    = 7


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



```
000112                  DERRTB  = 112
000113                  DERINB  = 113


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


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


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

```
                        CC1     = BIT1
                        CC0     = BIT0



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



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



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

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



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

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

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

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

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

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

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

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

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

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

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



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


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


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


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

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

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

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


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



```
                                ;-----------------------------------------------------------------------------
                                ; THIS IS START OF MODULE
                                ;-----------------------------------------------------------------------------
000466' 016767  177322  177570  START:  MOV     DVTD1,SELECT    ;GET ACTIVE DEVICES
000474' 001004                          BNE     SETUP           ;BR IF ANY ARE SELECTED
000476' 004767  004424          DRPMOD: JSR     PC,DISABL       ;DISABLE INTERRUPTS ON ALL SELECTED DEVICES
000502' 104410  000000'                 ENDS,BEGIN              ;;DROP THE DEVICE MODULE

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

```
 845  000742' 010362  000004                 MOV     R3,4(R2)        ;LOAD OUTPUT ISR VECTOR
 846  000746' 062762  000016  000004         ADD     #16,4(R2)       ;POINT TO OUTPUT ISR
 847  000754' 116762  177032  000006         MOVB    BR1,6(R2)       ;LOAD PRIORITY
 848  000762' 105062  000007                 CLRB    7(R2)           ;CLEAR HI BYTE OF STATUS WORD
 849                                 ;DEFINE THE MODE OF THIS DEVICE (FDX, HDX, P-P, M-P TRIB)
 850  000766' 112767  000002  177300         MOVB    #MODDEF,CSEL2   ;SET MODE DEFINITION COMMAND
 851  000774' 152760  000010  000001         BISB    #LULOOP,BSEL1(R0) ;SET LULOOP TO FORCE MODE
 852  001002' 126727  177010  000000         CMPB    SR1,#0          ;SEE IF GRP OF POINT-TO-POINT DEVICES
 853  001010' 001041                         BNE     9$              ;BR IF NOT
 854  001012' 132767  000200  177250         BITB    #ENABSW,LURG16  ;SEE IF LU SWITCHES ENABLED
 855  001020' 001410                         BEQ     6$              ;BR IF NOT
 856  001022' 132767  000020  177240         BITB    #MODSW0,LURG16  ;SEE IF HDX IN SWITCHES
 857  001030' 001404                         BEQ     6$              ;BR IF HDX
 858  001032' 112767  000003  177240         MOVB    #FDPP,CSEL6     ;SET FDX P-TO-P MODE
 859  001040' 000403                         BR      7$              ;GO SET MODE
 860  001042' 112767  000002  177230  6$:    MOVB    #HDPP,CSEL6     ;SET HDX P-TO-P MODE
 861  001050' 004767  003666          7$:    JSR     PC,INCMND       ;PERFORM MODE DEF'N CMND
 862  001054' 104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
 863  001060' 104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
 864  001064' 142760  000010  000001         BICB    #LULOOP,BSEL1(R0) ;CLEAR LULOOP
 865  001072' 032767  000001  177220         BIT     #INTIMO,ERRORS  ;SEE IF INPUT INTRPT TIMED-OUT
 866  001100' 001443                         BEQ     12$             ;BR IF NO TIME-OUT
 867  001102' 142760  000200  000001  8$:    BICB    #RUN,BSEL1(R0)  ;CLEAR RUN ON THIS DEVICE
 868  001140' 000167  177362                 JMP     DRPMOD          ;GO DROP MODULE
 869  001114' 126727  176676  000001  9$:    CMPB    SR1,#1          ;SEE IF GRP OF M-P TRIBUTARIES
 870  001122' 001020                         BNE     11$             ;BR IF NOT
 871  001124' 132767  000200  177136         BITB    #ENABSW,LURG16  ;SEE IF LU SWITCHES ENABLED
 872  001132' 001410                         BEQ     10$             ;BR IF NOT
 873  001134' 132767  000020  177126         BITB    #MODSW0,LURG16  ;SET HDX IN SWITCHES
 874  001142' 001404                         BEQ     10$             ;BR IF HDX
 875  001144' 112767  000007  177126         MOVB    #FDTS,CSEL6     ;SET FDX TRIB MODE
 876  001152' 000736                         BR      7$              ;GO SET MODE
 877  001154' 112767  000006  177116  10$:    MOVB    #HDTS,CSEL6     ;SET HDX TRIB MODE
 878  001162' 000732                         BR      7$              ;GO SET MODE
 879  001164' 004767  004000          11$:   JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
 880  001170' 012767  000007  176710         MOV     #SELERR,ERRTYP  ;SET CODE FOR SELECTION ERROR
 881                                         ;*************************************************************
 882  001176' 104405  000000' 000000        HRDER$,BEGIN,NULL       ;OPERATOR SPEC'D INVALID SR1 VALUE
 883                                         ;*************************************************************
 884  001204' 000167  177266                 JMP     DRPMOD          ;GO DROP THE MODULE
 885  001210' 005267  177040          12$:   INC     N.DEVS          ;UPDATE THE NO. OF DEVICES TO RUN
 886  001214' 105060  000000                 CLRB    BSEL0(R0)       ;DISABLE INTRPTS FROM THIS DEVICE
 887  001220' 000167  177356                 JMP     3$              ;CONTINUE SETUP
 888
 889                                 ;-------------------------------------------------------------------------
 890                                 ; ATTEMPT TO START THE PROTOCOL ON EACH LOGICAL LINK, AND IF A LOGICAL LINK
 891                                 ; CAN'T BE STARTED, DROP THE DEVICE WHICH CONTROLS THAT LINK, FROM TESTING
 892                                 ;-------------------------------------------------------------------------
 893  001224'                        STRTUP:
 894  001224' 016767  177024  177074         MOV     N.DEVS,LLKCNT   ;SET TOTAL NO. OF LOGICAL LINKS
 895  001232' 012767  000001  177074         MOV     #1,TRBADR       ;SET P-TO-P TRIB ADRS = 1
 896  001240' 012705  000354'                MOV     #INTABL+2,R5    ;INIT STATUS TABLE TRIB ADRS POINTER
 897  001244' 126727  176546  000001         CMPB    SR1,#1          ;SEE IF MULTIPOINT
 898  001252' 001003                         BNE     1$              ;BR IF NOT MULTIPOINT
 899  001254' 016767  176542  177052         MOV     SR3,TRBADR      ;INIT MULTIPOINT TRIB ADRS
 900  001262' 016700  176520          1$:    MOV     ADDR,R0         ;INIT DEVICE ADRS
```



```
 901  001266' 012767  000001  176772         MOV     #BIT0,DEVPTR    ;INIT SELECTION POINTER
 902                                 ;FIND A SELECTED DEVICE
 903  001274' 036767  176766  176762  2$:    BIT     DEVPTR,SELECT   ;SEE IF THIS DEVICE IS SELECTED
 904  001302' 001017                         BNE     22$             ;BR IF SELECTED
 905  001304' 006367  176756          20$:   ASL     DEVPTR          ;SHIFT SELECTION POINTER
 906  001310' 001002                         BNE     21$             ;BR TO KEEP SCANNING
 907  001312' 000167  000704                 JMP     19$             ;ALL DEVICES SCANNED - PROCEED
 908
 909  001316' 005767  176502          21$:   TST     SR4             ; IS THIS A DMV ?
 910  001322' 001402                         BEQ     25$
 911  001324' 062700  000010                 ADD     #10,R0          ;IF YES: INCREMENT DMV11 ADDRESS (20)
 912  001330' 062700  000010          25$:   ADD     #10,R0          ;IF NO: INCREMENT DMP11 ADDRESS (10)
 913  001334' 062705  000004                 ADD     #4,R5           ;INCR STATUS TABLE TRIB ADRS POINTER
 914  001340' 000755                         BR      2$              ;CONTINUE
 915                                 ;SET UP FOR CONTROL IN COMMAND
 916  001342' 112767  000001  176724  22$:    MOVB    #CTLIN,CSEL2    ;SET CONTROL IN CMND
 917  001350' 116767  176760  176717         MOVB    TRBADR,CSEL2+1  ;SET TRIB ADRS
 918  001356' 016715  176752                 MOV     TRBADR,(R5)     ;STORE THE TRIB ADRS IN STATUS TABLE
 919  001362' 005067  176710                 CLR     CSEL4
 920  001366' 005067  176706                 CLR     CSEL6
 921                                 ;ESTABLISH A LOGICAL LINK
 922  001372' 112767  000001  176700         MOVB    #ESTRIB,CSEL6   ;SET CODE FOR ESTABLISH TRIB CMND
 923  001400' 042767  000002  176710         BIC     #OUTINT,FLAGS   ;CLEAR OUTPUT INTRPT FLAG
 924  001406' 004767  003330                 JSR     PC,INCMND       ;ESTABLISH THIS TRIBUTARY
 925                                 ;CHANGE THE SELECTION INTERVAL TO 5 SECONDS
 926  001412' 005767  176406                 TST     SR4             ; IS THIS A DMV ?
 927  001416' 001404                         BEQ     23$
 928  001420' 012767  000764  176650         MOV     #500.,CSEL4     ;IF YES: CONSTANT = 500.
 929  001426' 000403                         BR      24$
 930  001430' 012767  011610  176640  23$:    MOV     #5000.,CSEL4    ;IF NO: CONSTANT = 5000.
 931  001436' 012767  000236  176634  24$:    MOV     #WRITSS+36,CSEL6 ;WRITE TSS ADDRESS 36 (SEL INTERVAL).
 932  001444' 042767  000002  176644         BIC     #OUTINT,FLAGS   ;CLEAR OUTPUT INTRPT FLAG.
 933  001452' 004767  003264                 JSR     PC,INCMND       ;CHANGE THE INTERVAL.
 934                                 ;ISTART THE LOGICAL LINK
 935  001456' 005067  176614                 CLR     CSEL4           ;CLEAR INTERVAL.
 936  001462' 012767  000010  176640         MOV     #8.,ISTCNT      ;INIT ISTART COUNTER FOR APPROX. 5 MINUTES
                                             ;   STARTUP (EACH COUNT = 35. SEC.)
 937
 938  001470' 112767  000003  176602  3$:    MOVB    #ISTART,CSEL6   ;SET CODE FOR ISTART REQUEST
 939  001476' 042767  000002  176612         BIC     #OUTINT,FLAGS   ;CLEAR OUTPUT INTRPT FLAG
 940  001504' 004767  003232                 JSR     PC,INCMND       ;PERFORM ISTART CMND
 941  001510' 012767  177777  176606         MOV     #-1,OUTIMR      ;INIT OUTPUT INTERRUPT TIMER
 942  001516'                        4$:
 943  001516' 104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
 944  001522' 104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
 945  001526' 132767  000002  176562         BITB    #OUTINT,FLAGS   ;SEE IF OUTPUT INTRPT SERVICED YET
 946  001534' 001015                         BNE     5$              ;BR IF YES
 947  001536' 005367  176562                 DEC     OUTIMR          ;DECREMENT TIMER
 948  001542' 001365                         BNE     4$              ;BR IF OUTPUT INTRPT DIDN'T TIME-OUT YET
 949  001544' 004767  003420                 JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
 950  001550' 012767  000023  176330         MOV     #NOINTR,ERRTYP  ;CODE FOR DEVICE FAILED TO INTRPT
 951                                         ;*************************************************************
 952  001556' 104405  000000' 000452'       HRDER$,BEGIN,REGTBL     ;OUTPUT INTERRUPT TIMED-OUT
 953                                         ;*************************************************************
 954  001564' 000167  000360                 JMP     14$             ;GO DROP MODULE
 955  001570' 142767  000370  176510  5$:    BICB    #CMDMSK,RSEL2   ;MASK FOR OUTPUT COMMAND FIELD
 956  001576' 126727  176504  000001         CMPB    RSEL2,#CTLOUT   ;CHK FOR CTL OUT CMND RETURNED
```

```
                        BEQ     6$              ;BR IF YES
                        JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
                        MOV     #NOTDFN,ERRTYP  ;CODE FOR ERROR NOT DEFINED (MUST INTERPRET
                                                ;   CSR CONTENTS)
                        ;*****************************************************************
                        HRDER$,BEGIN,REGTBL     ;UNEXPECTED OUTPUT CMND
                        ;*****************************************************************
                        JMP     14$             ;GO DROP MODULE
                    ;CHECK FOR LOGICAL LINK RUNNING YET
                    6$: CMPB    RSEL6,#RUNST    ;SEE IF PROTOCOL RUNNING ON THIS LOG LINK YET
                        BNE     7$              ;BR IF NOT
                        JMP     15$             ;GO SELECT NEXT LOGICAL LINK
                    ;CHECK FOR TRANSMIT THRESHOLD ERROR
                    7$: CMPB    RSEL6,#TTHRER   ;SEE IF TRANSMIT THRESHOLD ERROR (TRIB NOT RESPONDING)
                        BEQ     8$              ;BR IF YES
                        CMPB    RSEL6,#STHRER   ;SEE IF SELECT THRESHOLD ERROR
                        BEQ     8$              ;BR IF YES
                        JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
                        MOV     #NOTDFN,ERRTYP  ;CODE FOR ERROR NOT DEFINED (MUST INTERPRET
                                                ;   CSR CONTENTS)
                        ;*****************************************************************
                        HRDER$,BEGIN,REGTBL     ;UNEXPECTED CTL OUT CMND
                        ;*****************************************************************
                        JMP     14$             ;GO DROP MODULE
                    8$: DEC     ISTCNT          ;DECREMENT ISTART COUNTER
                        BNE     9$              ;BR IF COUNT NOT OVERFLOWED YET
                        JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
                        MOV     #NOTDFN,ERRTYP  ;CODE FOR ERROR NOT DEFINED (MUST INTERPRET
                                                ;   CSR CONTENTS)
                        ;*****************************************************************
                        HRDER$,BEGIN,REGTBL     ;TIMED-OUT ATTEMPTING TO START LINK
                        ;*****************************************************************
                        JMP     14$             ;GO DROP MODULE
                    ;HALT PROTOCOL PRIOR TO ISTART RETRY
                    9$: MOVB    #HALTST,CSEL6   ;SET CODE FOR HALT REQUEST
                        BIC     #OUTINT,FLAGS   ;CLEAR OUTPUT INTRPT FLAG
                        JSR     PC,INCMND       ;PERFORM HALT CMND TO RE-INIT LINK STATUS
                        MOV     #20000,OUTIMR   ;INIT OUTPUT INTERRUPT TIMER
                    10$:
                        BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
                        BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
                        BIT     #OUTINT,FLAGS   ;SEE IF OUTPUT INTRPT SERVICED YET
                        BNE     11$             ;BR IF YES
                        DEC     OUTIMR          ;DECREMENT TIMER
                        BNE     10$             ;BR IF OUTPUT INTRPT DIDN'T TIME-OUT YET
                        JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
                        MOV     #NOINTR,ERRTYP  ;CODE FOR DEVICE FAILED TO INTRPT
                        ;*****************************************************************
                        HRDER$,BEGIN,REGTBL     ;OUTPUT INTERRUPT TIMED-OUT
                        ;*****************************************************************
                        JMP     14$             ;GO DROP MODULE
                    11$: BICB   #CMDMSK,RSEL2   ;MASK FOR OUT CMND RETURNED
                    ;CHECK FOR INFO OUT WITH "BUFFERS RETURNED" CODE
                        CMPB    RSEL2,#INFOUT   ;CHK FOR INFO OUT CMND RETURNED
                        BNE     13$             ;BR IF NOT
                        BICB    #RTNMSK,RSEL6   ;MASK FOR RETURN KEY
```



```
                        CMPB    RSEL6,#BUFRTC   ;CHK FOR BUFFER RETURN COMPLETE
                        BNE     12$             ;BR IF NOT BUFFER RETURN COMPLETE
                        JMP     3$              ;GO TRY TO ISTART AGAIN
                    12$: JSR    PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
                        MOV     #NOTDFN,ERRTYP  ;CODE FOR ERROR NOT DEFINED (MUST INTERPRET
                                                ;   CSR CONTENTS)
                        ;*****************************************************************
                        HRDER$,BEGIN,REGTBL     ;UNEXPECTED INFO OUT CMND
                        ;*****************************************************************
                        BR      14$             ;GO DROP MODULE
                    13$: BIC    #OUTINT,FLAGS   ;CLEAR OUTPUT INTRPT FLAG
                        MOV     #20000,OUTIMR   ;INIT OUTPUT INTRPT TIMER
                        CLRB    BSEL2(R0)       ;RELEASE THE PORT AGAIN
                        BR      10$             ;KEEP LOOKING FOR INFO OUT CMND
                    14$: CLR    SEL0(R0)        ;CLEAR RUN, DISABLE INTRPTS ON THIS DEVICE
                        JMP     DRPMOD          ;GO DROP MODULE
                    ;SELECT NEXT LOGICAL LINK TO START
                    15$: CLRB   BSEL2(R0)       ;CLEAR BSEL2 (INCLUDING RDYO)
                        CMPB    SR1,#1          ;SEE IF MULTIPOINT
                        BNE     16$             ;BR IF NOT MULTIPOINT
                        INC     TRBADR          ;INCREMENT TRIB ADRS
                        TSTB    TRBADR          ;SEE IF TRIB ADRS OVERFLOW
                        BNE     16$             ;BR IF NOT
                        INC     TRBADR          ;IF TRIB ADRS = 0, MAKE IT 1
                                                ;   TRIB ADRS = 0 IS ILLEGAL
                    16$: CLRB   BSEL0(R0)       ;DISABLE INTRPTS FROM THIS DEVICE
                        JMP     20$             ;GO START NEXT LOGICAL LINK
                    19$: BR     REENTR          ;GO BEGIN TESTING ON ALL LOGICAL LINKS
                    ;-----------------------------------------------------------------------
                    ; RESTART ADDRESS FOR A PASS ON ALL DEVICES
                    ;-----------------------------------------------------------------------
                    RESTRT: MOV #16.,R5         ;INIT INTRPT STATUS TABLE ENTRY COUNTER
                        MOV     #INTABL,R4      ;INIT TABLE POINTER
                    1$: MOV     #MDFDON,(R4)    ;SET MODE DEF'N DONE FLAG IN STATUS BYTE
                        ADD     #4,R4           ;INCR STATUS TABLE INDEX
                        DEC     R5              ;DECR TABLE ENTRY COUNTER
                        BNE     1$              ;BR IF MORE TO DO
                        BR      ENABLE

                    ;-----------------------------------------------------------------------
                    ; PROGRAM ONLY COMES HERE AT START OF FIRST PASS
                    ;-----------------------------------------------------------------------
                    REENTR: MOV #16.,R5         ;INIT INTRPT STATUS TABLE ENTRY COUNTER
                        MOV     #INTABL,R4      ;INIT TABLE POINTER
                    1$: MOV     #MDFDON!1STDUN,(R4)     ;INIT DONE FLAGS FOR ITERATION
                        ADD     #4,R4           ;INCR STATUS TABLE INDEX
                        DEC     R5              ;DECREMENT COUNTER
                        BNE     1$              ;BR IF MORE TO DO

                    ;-----------------------------------------------------------------------
                    ;GET A WRITE BUFFER, ENABLE INTERRUPTS, REQUEST INPUTS ON ALL DEVICES
                    ;-----------------------------------------------------------------------
                    ENABLE: CLR INDON           ;INIT INPUT DONE COUNT TO 0
                        CLR     OUTDON          ;INIT OUTPUT DONE COUNT TO 0
                        GETPA$,BEGIN, RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
                        MOV     RBUFEA,BUFFEX
```

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



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

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



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

```
                                        ;*****************************************************************
003554' 104405  000000' 000452'         HRDER$,BEGIN,REGTBL     ;INCORRECT TX EA BITS RETURNED
                                        ;*****************************************************************
003562' 000425                          BR      7$
003564' 016001  000006          6$:     MOV     SEL6(R0),R1     ;GET SEL6 CONTENTS
003570' 032767  000002  174226          BIT     #MODE22,SR4     ;* IF THIS IS A DMV IN 022 MODE...
003576' 001402                          BEQ     28$             ;* THEN GET CHARACTER COUNT
003600' 016001  000010                  MOV     SEL10(R0),R1    ;* BITS FROM A DIFFERENT CSR.
003604' 042701  140000          28$:    BIC     #CCMSK,R1       ;MASK FOR CHAR COUNT BITS
003610' 020127  002000                  CMP     R1,#1024.       ;CHK FOR CORRECT CHAR CNT RETURNED
003614' 001410                          BEQ     7$              ;BR IF CORRECT
003616' 004767  001346                  JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
003622' 012767  000037  174256          MOV     #BADWRT,ERRTYP  ;CODE FOR UNABLE TO EXEC WRITE FUNCTION
                                        ;*****************************************************************
003630' 104405  000000' 000452'         HRDER$,BEGIN,REGTBL     ;INCORRECT TX CHAR CNT BITS RETURNED
                                        ;*****************************************************************
003636' 012764  000005  000352' 7$:     MOV     #MDFDON!ISTDON,INTABL(R4) ;RESET STATUS FOR NEXT ITERATION
003644' 152760  000200  000000          BISB    #RQI,BSEL0(R0)  ;ENABLE INPUTS AGAIN
003652' 000167  000354                  JMP     21$
                                ;HANDLE BACCOR
003656' 026067  000004  174242  8$:     CMP     SEL4(R0),RBUFPA ;CHK FOR CORRECT RCV BUFFER LO BITS RETURNED
003664' 001411                          BEQ     9$              ;BR IF CORRECT
003666' 004767  001276                  JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
003672' 012767  000036  174206          MOV     #BADRED,ERRTYP  ;CODE FOR UNABLE TO EXEC READ FUNCTION
                                        ;*****************************************************************
003700' 104405  000000' 000452'         HRDER$,BEGIN,REGTBL     ;INCORRECT RCV BA BITS RETURNED
                                        ;*****************************************************************
003706' 000454                          BR      11$
003710' 116001  000007          9$:     MOVB    BSEL7(R0),R1    ;GET BSEL7 CONTENTS
003714' 142701  000077                  BICB    #EAMSK,R1       ;MASK FOR BUS ADRS EA BITS
003720' 032767  000002  174076          BIT     #MODE22,SR4     ;* IF THIS IS A DMV IN 022 MODE...
003726' 001402                          BEQ     31$             ;* THEN GET EXTENDED ADDRESS (EA)
003730' 116001  000006                  MOVB    BSEL6(R0),R1    ;* BITS FROM A DIFFERENT CSR.
003734' 120167  174312          31$:    CMPB    R1,BUFFEX       ;CHK FOR CORRECT EA BITS RETURNED
003740' 001411                          BEQ     10$             ;BR IF CORRECT
003742' 004767  001222                  JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
003746' 012767  000036  174132          MOV     #BADRED,ERRTYP  ;CODE FOR UNABLE TO EXEC READ FUNCTION
                                        ;*****************************************************************
003754' 104405  000000' 000452'         HRDER$,BEGIN,REGTBL     ;INCORRECT RCV EA BITS RETURNED
                                        ;*****************************************************************
003762' 000426                          BR      11$
003764' 016001  000006          10$:    MOV     SEL6(R0),R1     ;GET SEL6 CONTENTS
003770' 032767  000002  174026          BIT     #MODE22,SR4     ;* IF THIS IS A DMV IN 022 MODE...
003776' 001402                          BEQ     29$             ;* THEN GET CHARACTER COUNT
004000' 016001  000010                  MOV     SEL10(R0),R1    ;* BITS FROM A DIFFERENT CSR.
004004' 042701  140000          29$:    BIC     #CCMSK,R1       ;MASK FOR CHAR COUNT BITS
004010' 020127  002000                  CMP     R1,#1024.       ;CHK FOR CORRECT CHAR CNT RETURNED
004014' 001411                          BEQ     11$             ;BR IF CORRECT
004016' 004767  001146                  JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
004022' 012767  000036  174056          MOV     #BADRED,ERRTYP  ;CODE FOR UNABLE TO EXEC READ FUNCTION
                                        ;*****************************************************************
004030' 104405  000000' 000452'         HRDER$,BEGIN,REGTBL     ;INCORRECT RCV CHAR CNT BITS RETURNED
                                        ;*****************************************************************
004036' 000400                          BR      11$
004040' 052764  000040  000352' 11$:    BIS     #BURDON,INTABL(R4) ;SET BACCOR DONE IN STATUS WORD
004046' 152760  000200  000000          BISB    #RQI,BSEL0(R0)  ;ENABLE INPUTS AGAIN
```



```
004054' 000466                          BR      21$             ;PROCEED
                                ;HANDLE CTL OUT
004056' 126027  000006  000010  12$:    CMPB    BSEL6(R0),#STARUN ;SEE IF START RECEIVED FROM MASTER
                                                                ;       WHILE RUNNING
004064' 001004                          BNE     13$             ;BR IF NOT
004066' 142760  000200  000000          BICB    #RQI,BSEL0(R0)  ;DISABLE INPUTS TO AWAIT OUTPUT
004074' 000456                          BR      21$             ;PROCEED
004076' 126027  000006  000024  13$:    CMPB    BSEL6(R0),#RUNST ;SEE IF PROTOCOL JUST ENTERED THE RUNNING
                                                                ;   STATE
004104' 001004                          BNE     14$             ;BR IF NOT
004106' 152760  000200  000000          BISB    #RQI,BSEL0(R0)  ;ENABLE INPUTS AGAIN
004114' 000446                          BR      21$             ;PROCEED
004116' 004767  001046  14$:    JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
004122' 012767  000000  173756          MOV     #NOTDFN,ERRTYP  ;CODE FOR ERROR NOT DEFINED (MUST INTERPRET
                                                                ;   CSR CONTENTS)
                                        ;*****************************************************************
004130' 104405  000000' 000452'         HRDER$,BEGIN,REGTBL     ;UNEXPECTED CTL OUT CMND
                                        ;*****************************************************************
004136' 000435                          BR      21$
                                ;HANDLE INFO OUT
004140' 116001  000006  15$:    MOVB    BSEL6(R0),R1    ;GET BSEL6 CODE RETURNED
004144' 142701  000340                  BICB    #RTNMSK,R1      ;MASK FOR RETURN CODE
004150' 120127  000020                  CMPB    R1,#BUFRTC      ;CHECK FOR BUFFER RETURN COMPLETE
004154' 001016                          BNE     19$             ;BR IF NOT
004156' 005267  174162                  INC     OUTDON          ;INCR CNT OF LOG LNKS RESTARTED BY MASTER
004162' 026767  174156  174136          CMP     OUTDON,LLKCNT   ;SEE IF ALL LOG LNKS DONE YET
004170' 002420                          BLT     21$             ;BR IF ALL NOT DONE
004172' 105060  000002                  CLRB    BSEL2(R0)       ;CLEAR BSEL2 (INCLUDING RDYO)
004176' 004767  000724                  JSR     PC,DISABL       ;DISABLE INTRPTS ON ALL DEVICES
004202' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION,
                                                                ;MONITOR SHALL TEST END OF PASS
004206' 000167  176012                  JMP     RESTRT          ;GO START NEW PASS
004212' 004767  000752  19$:    JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
004216' 012767  000000  173662          MOV     #NOTDFN,ERRTYP  ;CODE FOR ERROR NOT DEFINED (MUST INTERPRET
                                                                ;   CSR CONTENTS)
                                        ;*****************************************************************
004224' 104405  000000' 000452'         HRDER$,BEGIN,REGTBL     ;UNEXPECTED INFO OUT CMND
                                        ;*****************************************************************
004232' 105060  000002  21$:    CLRB    BSEL2(R0)       ;CLEAR BSEL2 (INCLUDING RDYO)
004236' 000167  176722                  JMP     CHKMOR          ;GO SEE IF RDYI OR RDYO STILL SET,
                                                                ;   AND RETURN TO DEC/X11 MONITOR IF NOT


                                ;*************************************************************************
                                ;* DMVLUP - THIS SUBROUTINE ENTERS M-LOOP, SETS TTL LOOPBACK AND TIMER 1 FOR
                                ;*      56K BPS, AND EXITS MLOOP. (DMV ONLY)
                                ;*************************************************************************
004242' 112760  000301  000001  DMVLUP: MOVB    #301,BSEL1(R0)  ;ENTER MAINTENANCE LOOP
004250'                         1$:
004250' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
004254' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
004260' 132760  000200  000002          BITB    #RDY,BSEL2(R0)  ;MAKE SURE MLOOP COMPLETE
004266' 001770                          BEQ     1$
004270' 012760  000006  000002          MOV     #TTLOOP,BSEL2(R0) ;ISSUE TTLOOP CONFIGURE COMMAND
004276' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
004302' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
```

```
                        NOP
                        NOP
                        RTS     PC              ;RETURN


;*************************************************************************
;* DMVRD - THIS SUBROUTINE FORCES THE DMV WHOSE ADRS IS PASSED IN R0 ON ENTRY
;*      TO ENTER THE M-LOOP AND READ THE INTERNAL DMV LOCATION PASSED IN THE
;*      WORD FOLLOWING THE CALL.
;*************************************************************************
DMVRD:  MOVB    #301,BSEL1(R0)  ;ENTER MAINTENANCE LOOP
1$:
        BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
        BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
        BITB    #MRDY,BSEL2(R0) ;MAKE SURE MLOOP COMPLETE
        BEQ     1$
        MOV     @(SP),SEL4(R0)  ;PASS DMV INTERNAL ADDRESS
        MOV     #MEDLOC,BSEL2(R0) ;ISSUE READ COMMAND
2$:
        BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
        BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
        BITB    #MRDY,BSEL2(R0) ;MAKE SURE MLOOP READ IS COMPLETE
        BEQ     2$
        ADD     #2,(SP)         ;FIX UP RETURN PC
        RTS     PC              ;RETURN


;*************************************************************************
;* EXECUT - THIS SUBROUTINE FORCES THE MICROPROCESSOR WHOSE ADRS IS PASSED
;*      IN R0 ON ENTRY TO EXECUTE AN INSTRUCTION WHICH IS PASSED IN THE WORD
;*      FOLLOWING THE CALL.
;*************************************************************************
EXECUT: BISB    #ROMO!ROMI,BSEL1(R0)    ;SET ROMO, ROMI BITS IN BSEL1
        MOV     @(SP),SEL6(R0)  ;PUT INSTRUCTION INTO SEL6
        BISB    #ROMO!ROMI!STEPMP,BSEL1(R0) ;SET ROMO, ROMI, STEPMP IN BSEL1
        BICB    #ROMO!ROMI!STEPMP,BSEL1(R0) ;CLEAR ROMO, ROMI, STEPMP IN BSEL1
        ADD     #2,(SP)         ;FIX UP RETURN PC
        RTS     PC              ;RETURN


;*************************************************************************
;* INIDMP - THIS SUBROUTINE ISSUES A MASTER CLEAR AND CLEARS THE CSR'S,
;*      FOR THE DEVICE WHOSE ADDRESS IS PASSED IN R0 ON ENTRY.
;*************************************************************************
INIDMP: MOV     R1,-(SP)        ;SAVE R1
        JSR     PC,CLRCSR       ;CLEAR BSEL0, SEL2,4,6
        MOVB    #MCLR,BSEL1(R0) ;SET MASTER CLEAR BIT
        TST     SR4             ; IF DMV...
        BNE     6$              ; THEN  DON'T SET RUN
;-------------------------------------------------------------------------
;;;     MOVB    #RUN,BSEL1(R0)  ;PATCH TO SET RUN BIT IF USING M8206
        NOP                     ; 112760
        NOP                     ;    200
        NOP                     ;      1
;-------------------------------------------------------------------------
```



```
6$:     BIC     #BADINI,ERRORS  ;CLEAR BAD INIT ERROR FLAG
        MOV     #20000,R1       ;INITIALIZE INIT TIMER
1$:     BITB    #RUN,BSEL1(R0)  ;SEE IF RUN IS SET AGAIN
        BNE     3$              ;BR IF NOT SET
        DEC     R1              ;DECR TIMER
        BEQ     2$              ;BR IF INIT TIMED-OUT
        BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
        BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
        BR      1$              ;GO CHECK AGAIN
2$:     JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
        MOV     #NOINIT,ERRTYP  ;CODE FOR DEVICE WILL NOT INIT
        ;*****************************************************************
        HRDER$,BEGIN,REGTBL     ;TIMED-OUT WAITING FOR RUN TO SET
        ;*****************************************************************
        BR      4$
3$:     CMPB    BSEL6(R0),#GOODIN ;SEE IF INIT COMPLETE CODE SET BY MICRO-CPU
        BEQ     5$              ;BR IF YES
        JSR     PC,GETERR       ;LOAD ERROR INFORMATION FOR PRINTOUT
        MOV     #NOINIT,ERRTYP  ;CODE FOR DEVICE WILL NOT INIT
        ;*****************************************************************
        HRDER$,BEGIN,REGTBL     ;INIT COMPLETE CODE NOT SET BY MICRO-CPU
        ;*****************************************************************
4$:     BIS     #BADINI,ERRORS  ;SET BAD INIT ERROR FLAG
5$:     JSR     PC,CLRCSR       ;CLEAR BSEL0, SEL2,4,6
        MOV     (SP)+,R1        ;RESTORE R1
        RTS     PC              ;RETURN


;*************************************************************************
;* READ16 - THIS SUBROUTINE FORCES THE MICROPROCESSOR WHOSE ADDRESS IS
;*      PASSED IN R0 ON ENTRY, TO EXECUTE AN INSTRUCTION WHICH READS DMP LINE
;*      UNIT IBUS REG 16 INTO BSEL4, AND THEN IT PLACES THE CONTENTS INTO LOC.
;*      LURG16 (IF DMV: EXECUTE MLOOP, "SWPBOT" => LURG16, AND ADJUST).
;*************************************************************************
READ16: TST     SR4             ;IS THIS A DMV ?
        BNE     1$              ;BRANCH IF YES

        JSR     PC,EXECUT       ;DMP: EXECUTE MOVE INSTRUCTION
        .WORD   RDLU16
        MOVB    BSEL4(R0),LURG16 ;GET REG 16 CONTENTS INTO LURG16
        BR      2$

1$:     JSR     PC,DMVRD        ;DMV: READ "SWPBOT" SWITCHES
        .WORD   SWPBOT
        MOVB    BSEL6(R0),LURG16 ;PUT IT IN LURG16
        CLC                     ;ADJUST "SWPBOT" BITS
        ASRB    LURG16          ; TO LOOK LIKE THE DMP
        BCC     2$              ;  REG 16 SWITCHES
        BISB    #ENABSW,LURG16

2$:     CLRB    LURG16+1        ;CLEAR HI BYTE
        RTS     PC              ;RETURN
```

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


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


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



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


                                ;*******************************************************************************
                                ;* DISABL - DISABLE INTERRUPTS ON ALL DEVICES
                                ;*******************************************************************************
005126' 016700  172654          DISABL: MOV     ADDR,R0         ;INIT DEVICE ADDRESS
005132' 012767  000001  173126          MOV     #BIT0,DEVPTR    ;INIT SELECTION POINTER
005140' 036767  173122  173116  1$:     BIT     DEVPTR,SELECT   ;SEE IF THIS DEVICE IS SELECTED
005146' 001402                          BEQ     2$              ;BR IF NOT SELECTED
005150' 105060  000000                  CLRB    BSEL0(R0)       ;CLEAR RQI, IEI, IEO
005154' 062700  000010          2$:     ADD     #10,R0          ;INCR DEVICE ADDRESS
005160' 006367  173102                  ASL     DEVPTR          ;SHIFT SELECTION POINTER
005164' 001365                          BNE     1$              ;BR IF MORE TO SCAN
005166' 000207                          RTS     PC              ;RETURN


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


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

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


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

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

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


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

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

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


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

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

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



```


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

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

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


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

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

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


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

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

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


```

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



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

```
1823                                    ;-------------------------------------------------------------
1824  006062' 016700  000022           1$:   MOV     ADDR14,R0       ;PUT DEVICE 14 CSR ADDRESS IN R0
1825  006066' 000167  174366                 JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 14
1826
1827  006072'                         OISR14:
1828                                    ;-------------------------------------------------------------
1829  006072' 000004  000000' 006100'        PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
1830                                    ;-------------------------------------------------------------
1831  006100' 016700  000004           1$:   MOV     ADDR14,R0       ;PUT DEVICE 14 CSR ADDRESS IN R0
1832  006104' 000167  175116                 JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 14
1833
1834  006110' 000000                  ADDR14: .WORD   0               ;DEVICE CSR ADDRESS STORED HERE
1835
1836
1837
1838  006112'                         IISR15:
1839                                    ;-------------------------------------------------------------
1840  006112' 000004  000000' 006120'        PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
1841                                    ;-------------------------------------------------------------
1842  006120' 016700  000022           1$:   MOV     ADDR15,R0       ;PUT DEVICE 15 CSR ADDRESS IN R0
1843  006124' 000167  174330                 JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 15
1844
1845  006130'                         OISR15:
1846                                    ;-------------------------------------------------------------
1847  006130' 000004  000000' 006136'        PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
1848                                    ;-------------------------------------------------------------
1849  006136' 016700  000004           1$:   MOV     ADDR15,R0       ;PUT DEVICE 15 CSR ADDRESS IN R0
1850  006142' 000167  175060                 JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 15
1851
1852  006146' 000000                  ADDR15: .WORD   0               ;DEVICE CSR ADDRESS STORED HERE
1853
1854
1855
1856  006150'                         IISR16:
1857                                    ;-------------------------------------------------------------
1858  006150' 000004  000000' 006156'        PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
1859                                    ;-------------------------------------------------------------
1860  006156' 016700  000022           1$:   MOV     ADDR16,R0       ;PUT DEVICE 16 CSR ADDRESS IN R0
1861  006162' 000167  174272                 JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 16
1862
1863  006166'                         OISR16:
1864                                    ;-------------------------------------------------------------
1865  006166' 000004  000000' 006174'        PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
1866                                    ;-------------------------------------------------------------
1867  006174' 016700  000004           1$:   MOV     ADDR16,R0       ;PUT DEVICE 16 CSR ADDRESS IN R0
1868  006200' 000167  175022                 JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 16
1869
1870  006204' 000000                  ADDR16: .WORD   0               ;DEVICE CSR ADDRESS STORED HERE
1871
1872
1873
1874  006206'                         IISR17:
1875                                    ;-------------------------------------------------------------
1876  006206' 000004  000000' 006214'        PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
1877                                    ;-------------------------------------------------------------
1878  006214' 016700  000022           1$:   MOV     ADDR17,R0       ;PUT DEVICE 17 CSR ADDRESS IN R0
```



```
1879  006220' 000167  174234                 JMP     INISR           ;GO SERVICE INPUT INTERRUPT ON DEVICE 17
1880
1881  006224'                         OISR17:
1882                                    ;-------------------------------------------------------------
1883  006224' 000004  000000' 006232'        PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
1884                                    ;-------------------------------------------------------------
1885  006232' 016700  000004           1$:   MOV     ADDR17,R0       ;PUT DEVICE 17 CSR ADDRESS IN R0
1886  006236' 000167  174764                 JMP     OUTISR          ;GO SERVICE OUTPUT INTERRUPT ON DEVICE 17
1887
1888  006242' 000000                  ADDR17: .WORD   0               ;DEVICE CSR ADDRESS STORED HERE
1889
1890
1891
```

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



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

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



```
IISR14  006054R         1820#
IISR15  006112R         1838#
IISM16  006150R         1856#
IISR17  006206R         1874#
IISR2   005400R         1640#
IISR3   005436R         1658#
IISR4   005474R         1676#
IISR5   005532R         1694#
IISR6   005570R         1712#
IISR7   005626R         1730#
ILGINT= 000011           350#  1132   1139   1239
IMODX.= 000000           300#
INCMND  004742R          861    924    933    940    993   1511#
INDON   000342R          759#  1065*  1178*
INFOUT= 000002           396#  1010   1235
INIDMP  004442R          837   1420#
ININT  = 000001          700#  1113   1513   1523
INISR   002460R         1106#  1186   1609   1627   1645   1663   1681   1699   1717   1735   1753   1771   1789
                        1807   1825   1843   1861   1879
INIT    000030R          255#
INTABL  000352R          776#   807    809    896   1045   1056   1122   1123   1125   1127   1129   1149*  1162*
                        1176*  1279*  1317*
INTFDG= 000022           469#
INTIMU= 000001           722#   865   1511   1532
INTIMR  000322R          749#  1518*  1525*
INTR    000120R          287#
IRDY   = 000020          645#
ISTART= 000003           464#   938   1148
ISTCNT  000330R          754#   936*   981*
ISTDON= 000004           711#  1057   1127   1149   1279
KILLTR= 000002           463#
LATPS  = 000040          480#
LLKCNT  000326R          751#   815*   894*  1344
LNKDON  000340R          758#
LULOOP= 000010           435#   835    851    864
LULP   = 000040          591#   644#
LURG16  000270R          736#   854    856    871    873   1471*  1476*  1478*  1480*  1482*
MAINT  = 000004          465#
MAINT1= 000010           600#
MAINT2= 000004           601#
MAP22$= 104416           302#
MCLK   = 000002          692#
MCLR   = 000100          433#   830   1422
MDFDON= 000001           710#  1046   1057   1129   1279
MNTHLT= 000014           539#
MNTRUN= 000012           538#
MODDEF= 000002           380#   850
MODESW= 000361           409#
MODE22= 000002           341#  1071   1157   1170   1256   1268   1293   1305
MODMSK= 000370           404#
MODNAM  000000R          242#
MODR   = 000010          658#
MODSP   000252R          256    300#
MODSW0= 000020           678#   856    873
MODSW1= 000040           679#
MODSW2= 000100           680#
```

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



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

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

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


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

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

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

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


. ABS.  000000    000
        010340    001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

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

```
.NLIST
.NLIST  SEQ,LD,BIN
.LIST
.REM    %
```


IDENTIFICATION

PRODUCT CODE: AC-F087D-MC

PRODUCT NAME: CXRPDD0 RH70/RP04,5,6 MODULE

PRODUCT DATE: JANUARY 1980

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

RPD IS AN IOMODX WHICH WILL EXERSIZE UP TO 8 RP04/5/6 DISK DRIVES ON AN RH11/RH70 CONTROLLER, OR DUAL-PORTED BETWEEN TWO SUCH CONTROLLERS. IN SINGLE-PORT MODE, THE ENTIRE SURFACE OF EACH DISK WILL BE TESTABLE (RANDOMLY OR SYSTEMATICALLY).

IN DUAL-PORT MODE, THE "A" CONTROLLER MODULE WILL ACCESS ONLY THE LOW HALF OF THE DISK, AND THE "B" CONTROLLER MODULE WILL ACCESS ONLY THE HIGH HALF OF THE DISK; NO COMMUNICATION WILL BE ATTEMPTED BETWEEN THE TWO PORTS. ONE COPY OF THIS MODULE MUST BE CONFIGURED FOR EACH PORT.

SINGLE/DUAL PORT MODE WILL BE AUTOMATICALLY SELECTED ON THE BASIS OF THE RPDT REGISTER OF EACH SELECTED DRIVE.

DURING TESTING, ANY BAD SECTOR FOUND WILL BE STORED AUTOMATICALLY AND WITHOUT OPERATOR INTERVENTION INTO THE MODULE'S BADSPOT TABLE. A MESSAGE CAN BE PRINTED TO NOTIFY THE OPERATOR OF THE BADSPOT ADDRESS (SEE OPERATOR OPTIONS). SECTORS LISTED IN THE TABLE WILL NOT BE TESTED ON ANY DRIVE.

THE WBUFRQ WORD IN THE HEADER MAY BE CHANGED AT WILL TO ALTER THE MODULE'S TRANSFER SIZE (AND OF COURSE A AND B-PORT SIZES NEED NOT MATCH): THIS WILL NOT ALTER THE EFFECTIVENESS OF THE BADSPOT TABLE OR CAUSE THE MODULE TO ATTEMPT ACCESS OF OUT-OF-RANGE DISK ADDRESSES. WITHIN THIS FRAMEWORK, THE MODULES WILL ACCESS AS MUCH OF THE DISK AS POSSIBLE. RANDOM SEEKS WILL BE USED UNLESS THIS OPTION IS DESELECTED IN SR1.

2. REQUIREMENTS

HARDWARE: 1 TO 8 RP04/5/6 DRIVES ON 1 OR 2 RH11/RH70 CONTROLLERS

STORAGE:: RPD REQUIRES:

DECIMAL WORDS:	1806
OCTAL WORDS:	3416
OCTAL BYTES:	7034

3. PASS DEFINITION

- ONE PASS CONSISTS OF 300 ITERATIONS.
- AN ITERATION CONSISTS OF THE FOLLOWING STEPS EXECUTED ON EACH SELECTED DRIVE:
  - A. WRITE DATA TO DISK. IN DUAL-PORT MODE, PORT "A" MODULE WILL TEST THE LOW HALF OF THE DISK (CYLINDERS 0 TO 407 FOR RP06, CYLINDERS 0 TO 205 FOR RP04/5), PORT "B" WILL EXERSIZE THE HIGH HALF (CYLINDERS 206 TO 410 FOR RP04/5, CYLINDERS 408 TO 814 FOR RP06). IN SINGLE-PORT MODE, THE MODULE WILL ACCESS THE ENTIRE DISK SURFACE (CYLINDERS 0 TO 410 FOR RP04/5, CYLINDERS 0 TO 814 FOR RP06).
  - B. WRITE-CHECK DATA JUST WRITTEN.
  - C. READ 256 WORDS (1 SECTOR) INTO MODULE READ BUFFER.
  - D. DO IN-CORE COMPARE OF READ BUFFER WITH FIRST 256 WORDS OF WRITE BUFFER.
  - E. RELEASE THE DRIVE FOR OTHER PORT TO TEST.

4. EXECUTION TIME

ONE PASS OF RPD RUNNING ALONE ON A PDP-11/70 TAKES ABOUT A MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:
DVA-176700 VCT-254 BR1-5 DVC-1

REQUIRED PARAMETERS:
SR1 MUST BE SET UP TO INDICATE WHICH PORT (A/B) THIS COPY OF THE MODULE IS TO TEST, IF ANY DRIVES ARE DUAL-PORTED. ALSO, BIT 7 MUST BE SET IF THE CONTROLLER IS AN RH70, AND CLEARED IF IT IS NOT AN RH70.

6. DEVICE/OPTION SETUP

MAKE SURE ALL DRIVES ARE POWERED UP, WRITE ENABLED, AND READY.
FOR DUAL-PORTED DRIVES: MAKE SURE CONTROLLER SELECT SWITCH IS IN A/B POSITION. IF DRIVE IS NOT CYCLED UP WITH SWITCH IN CORRECT POSITION, PLACE SWITCH CORRECTLY, AND DISABLE THEN RE-ENABLE DRIVE USING DRIVE DISABLE SWITCH.

7. OPERATOR OPTIONS

SR1

BIT 2
- CLEAR(0):
  TYPE OUT DATA LATE ERRORS (HARD ERROR) AND COUNT THEM INTERNALLY.
- SET(1):
  COUNT DATA LATE ERRORS INTERNALLY, BUT DO NOT TYPE OUT.

BIT 4
- CLEAR(0):
  THIS COPY OF RPD TESTS A-PORT ON ANY DUAL-PORTED DRIVES
- SET(1):
  THIS COPY OF RPD TESTS B-PORT ON ANY DUAL-PORTED DRIVES

BIT 5
- CLEAR(0):
  TEST DISKS WITH RANDOM SEEKS
- SET(1):
  DISABLE RANDOM SEEKS. INCREMENT SECTOR BY ONE EACH CYCLE.

BIT 6

CLEAR(0):
DO NOT TYPE OUT BADSPOTS.
SET(1):
ALWAYS TYPE OUT ANY BADSPOT NOT ALREADY RECORDED ON BADSPOT TABLE.

BIT 7
CLEAR(0):
CONTROLLER IS RH11
SET(1):
CONTROLLER IS RH70 (HAS RHBAE AND RHCS3)

8. NON-STANDARD PRINTOUTS

A. MOST PRINTOUTS HAVE THE STANDARD FORMATS.

B. ERROR MESSAGE DUMP RH11 REGISTERS IN THE FOLLOWING ORDER:

RHCS1	RHWC	RHBA	RPDA	RHCS2	RPDS	RPER1	RPAS
RPLA	RHDB	RPMR	RPDT	RPSN	RPOF	RPDC	RPCC
RPER2	RPER3	RPEC1	RPEC2				

ADDITIONALLY FOR THE RH70 RPBAE RHCS3

C. THE BAD SECTOR MESSAGE (WHEN ENABLED) LOOKS LIKE:
'DRIVE X: BADSPOT AT (OCTAL) CYL: XX, TRK: XX, SEC: XX'
WHERE 'X' IS AN OCIAL DIGIT. NO ERROR IS ASSOCIATED WITH THIS MESSAGE.

10. CHANGE HISTORY

NOTE: HISTORY STARTS WITH REV. D0

D0: ENHANCED ERROR REPORTING ROUTINES.

*

```
                                .SBTTL  MODULE HEADER BLOCK                                                     ;DRB001
000000'                         IDMODX <RPDD  >,176700,254,5,0,0,1500,104,RBUF,256.,256.
000000'                         MODULE  150000,RPDD  ,176700,254,5,0,0,1500,104,RBUF,256.,256.
                                .TITLE  RPDD DEC/X11 SYSTEM EXERCISER MODULE
                        ;       DDXCOM  VERSION  6      23-MAY-78
                                .LIST   BIN
                                ;*****************************************************************************
000000'                         BEGIN:
000000' 050122  042104     040  MODNAM: .ASCII  /RPDD / ;MODULE NAME.
000005'     000                 XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 176700                  ADDR:   176700+0                ;1ST DEVICE ADDR.
000010' 000254                  VECTOR: 254+0                   ;1ST DEVICE VECTOR.
000012'     240                 BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013'     000                 BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014' 000001                  DVID1:  0+1                     ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN                    ;SWITCH REGISTER 4
                                ;*****************************************************************************
000026' 150000                  STAT:   150000                  ;STATUS WORD.
000030' 002054'                 INIT:   START                   ;MODULE START ADDR.
000032' 000252'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                       ;PASS COUNTER.
000036' 001500                  ICONT:  1500                    ;# OF ITERATIONS PER PASS=1500
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                  A*AS:   OPEN                    ;ACTUAL DATA.
000112' 002236'                 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' 000104                  IDNUM:  104                     ;MODULE IDENTIFICATION NUMBER=104
000124' 000722'                 RBUFVA: RBUF                    ;READ BUFFER VIRTUAL ADDRESS
```



```
000126' 000000                  RBUFPA: OPEN                    ;READ BUFFER PHYSICAL ADDRESS
000130' 000000                  RBUFEA: OPEN                    ;READ BUFFER EA BITS
000132' 000400                  RBUFSZ: 256.                    ;SIZE OF THE READ BUFFER
000134' 000000                  WBUFPA: OPEN                    ;WRITE BUFFER PHYSICAL ADDRESS
000136' 000000                  WBUFEA: OPEN                    ;WRITE BUFFER EA BITS
000140' 000400                  WBUFRQ: 256.                    ;WRITE BUFFER SIZE REQUESTED
000142' 000000                  WBUFSZ: OPEN                    ;WRITE BUFFER SIZE AVAILABLE
000144' 000000                  CDERCT: OPEN                    ;CDATA/DATCK ERROR COUNT
000146' 000000                  CDWDCT: OPEN                    ;CDATA/DATCK WORD COUNT
000150' 000000                  FREE:   OPEN                    ;RESERVED FOR FUTURE USE
        000040                          .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                        .NLIST
                                        .WORD   0
                                        .LIST
                                        .ENDR
000252'                         MODSP:
                                ;*****************************************************************************
```

```
                                .SBTTL  MODULE PRIVATE DATA                                             ;DRB001
000252'                         BADSPT:                 ;BAD SPOT TABLE.  THERE IS ROOM FOR 48          ;DRB001
                                                        ;BAD SPOTS, IN TWO-WORD FORMAT, CYLINDER        ;DRB001
                                                        ;FIRST, THEN SECTOR IN LOW BYTE OF SECOND       ;DRB001
                                                        ;WORD, AND TRACK IN HIGH BYTE.                  ;DRB001
000252' 177777                          177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000254' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000256' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000260' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000262' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000264' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000266' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000270' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000272' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000274' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000276' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000300' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000302' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000304' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000306' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000310' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000312' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000314' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000316' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000320' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000322' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000324' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000326' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000330' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000332' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000334' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000336' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000340' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000342' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000344' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000346' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000350' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000352' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000354' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000356' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000360' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000362' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000364' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000366' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000370' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000372' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000374' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000376' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000400' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000402' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000404' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000406' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000410' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000412' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000414' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000416' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
```



```
(1) 000420' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000422' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000424' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000426' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000430' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000432' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000434' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000436' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000440' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000442' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000444' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000446' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000450' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000452' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000454' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000456' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000460' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000462' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000464' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000466' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000470' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000472' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000474' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000476' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000500' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000502' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000504' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000506' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000510' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000512' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000514' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000516' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000520' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000522' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000524' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000526' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000530' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000532' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000534' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000536' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000540' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000542' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000544' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
(1) 000546' 177777                      177777          ;CYLINDER ADDRESS                               ;DRB001
(1) 000550' 177777                      177777          ;TRACK/SECTOR ADDRESS                           ;DRB001
000552'                         ENDBBK:                                                                 ;DRB001
                                                                                                        ;DRB001
        000004                  DLATE   = BIT2                                                          ;DRB001
        000020                  BPORT   = BIT4                                                          ;DRB001
        000040                  NORAND  = BIT5                                                          ;DRB001
        000100                  BDSPT   =BIT6                                                           ;DRB001
        000200                  RH70    = BIT7                                                          ;DRB001
        137400                  HRDCS2  =137400
        177777                  HRDER1  =177777
        167777                  HRDER2  =167777
        140153                  HRDER3  =140153
```

```
  392                                                                    ;**-16                  ;DRB001
  393  000552' 000000  000000  000000  S:      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
       000560' 000000  000000  000000
       000566' 000000  000000  000000
       000574' 000000  000000  000000
       000602' 000000  000000  000000
       000610' 000000  000000  000000
       000616' 000000  000000  000000
       000624' 000000
  394                          ;DO NOT CHANGE THE ORDER OF THE NEXT 4 LOCATIONS
  395                          ;NEEDED FOR MAP22 ROUTINE
  396  000626' 000000          PA18:   0
  397  000630' 000000          XMEM:   0
  398  000632' 000000          PA22:   0
  399  000634' 000000          EA22:   0
  400                                                                    ;**-5                   ;DRB001
  401  000636' 000252'          NXTBBK: BADSPT          ; NEXT OPEN BADSPOT IN TABLE           ;DRB001
  402                                                                                            ;DRB001
  403  000640' 000000          CYLLIM: 0               ; UPPER DISK LIMIT (DEPENDS ON DISK TYPE);DRB001
  404  000642' 000000          CYLNDR: 0               ;ABSOLUTE CYLINDER ADDRESS OF CURRENT OP.;DRB001
  405  000644' 000000          CYL:    0               ;RELATIVE DISK ADDRESS                  ;DRB001
  406  000646'                 DSKADR:                 ;FOR REFERENCING TRACK/SECTOR AS WORD   ;DRB001
  407  000646'     000         SECTOR: .BYTE   0       ;SECTOR ADDRESS OF CURRENT OPERATION    ;DRB001
  408  000647'     000         TRACK:  .BYTE   0       ;TRACK ADDRESS OF CURRENT OPERATION     ;DRB001
  409                                                                                            ;DRB001
  410  000650' 000000          BADCYL: 0               ;FROM DRIVE LOCATION REGISTERS, ON FINDING BADSPOT;DRB001
  411  000652' 000000          BADSEC: 0               ;FROM RPDA ON BADSPOT OCCURANCE         ;DRB001
  412  000654' 000000          BADTRK: 0               ;FROM RPDA+1 ON BADSPOT         ;DRB001
  413                                                                                            ;DRB001
  414  000656' 000000          LASCYL: 0               ;LAST CYLINDER WHICH CAN BE WRITTEN     ;DRB001
  415  000660' 000000          LASTRK: 0       ;LAST TRACK WHICH CAN BE WRITTEN                ;DRB001
  416  000662' 000000          LASSEC: 0       ;LAST SECTOR WHICH CAN BE WRITTEN               ;DRB001
  417                                                                                            ;DRB001
  418  000664' 000000          SIZTRK: 0       ;NUMBER OF TRACKS IN WBUFSZ                     ;DRB001
  419  000666' 000000          SIZSEC: 0       ;REMAINDER OF WBUFSZ, IN SECTORS                ;DRB001
  420  000670' 000000          FLAG:   0
  421  000672' 000000          DVICE:  0
  422  000674' 000000          DRIVE:  0
  423  000676' 000000          UNITNO: 0
  424                                                                    ;**-1                   ;DRB001
  425  000700' 000000          DLTCNT: 0
  426  000702' 000000          FUNC:   0
  427  000704' 000000          TIMER:  0
  428  000706' 000000          ZERO:   0
  429  000710' 000000          FERADR: 0
  430  000712' 000000          CLK:    0
  431  000714' 000000          TRY:    0
  432  000716' 000000          STORE:  0
  433  000720' 000000          HRDERR: 0
  434  000722' 000400          RBUF:   .BLKW   256.
  435
  436  001722'                 TABLE:
  437  001722' 000552'                 S
  438  001724' 000554'                 S+2
  439  001726' 000556'                 S+4
  440  001730' 000560'                 S+6
```



```
  441  001732' 000562'                 S+10
  442  001734' 000564'                 S+12
  443  001736' 000566'                 S+14
  444  001740' 000570'                 S+16
  445  001742' 000572'                 S+20
  446  001744' 000574'                 S+22
  447  001746' 000576'                 S+24
  448  001750' 000600'                 S+26
  449  001752' 000602'                 S+30
  450  001754' 000604'                 S+32
  451  001756' 000606'                 S+34
  452  001760' 000610'                 S+36
  453  001762' 000612'                 S+40
  454  001764' 000614'                 S+42
  455  001766' 000616'                 S+44
  456  001770' 000620'                 S+46
  457  001772' 000622'                 S+50
  458  001774' 000624'                 S+52
  459                                                                    ;**-1                   ;DRB001
  460  001776' 177777                  177777
  461  002000' 000000          RHCS1:  0
  462  002002' 000000          RHWC:   0
  463  002004' 000000          RHBA:   0
  464  002006' 000000          RPDA:   0
  465  002010' 000000          RHCS2:  0
  466  002012' 000000          RPDS:   0
  467  002014' 000000          RPER1:  0
  468  002016' 000000          RPAS:   0
  469  002020' 000000          RPLA:   0
  470  002022' 000000          RHDB:   0
  471  002024' 000000          RPMR:   0
  472  002026' 000000          RPDT:   0
  473  002030' 000000          RPSN:   0
  474  002032' 000000          RPOF:   0
  475  002034' 000000          RPDC:   0
  476  002036' 000000          RPCC:   0
  477  002040' 000000          RPER2:  0
  478  002042' 000000          RPER3:  0
  479  002044' 000000          RPEC1:  0
  480  002046' 000000          RPEC2:  0
  481  002050' 000000          RHBAE:  0
  482  002052' 000000          RHCS3:  0
  483                                                                    ;**-4                   ;DRB001
```

```
                                .SBTTL  MODULE CODE                                                     ;DRB001
002054' 012767  000400  176032  START:  MOV     #256.,WDTC      ;256 WORDS TO MEM/ITERATION
002062' 012767  000400  176026          MOV     #256.,WDFR      ;256 WORDS FROM MEM/ITERATION
002070' 012767  000004  176022          MOV     #4,INTR         ;4 INTERRUPTS/ITERATION
002076' 012701  000252'                 MOV     #BADSPT,R1      ;BAD SPOT TABLE                 REV D
002102' 022721  177777          4$:     CMP     #-1,(R1)+       ;END YET ?                      REV D
002106' 001375                          BNE     4$                                      ;REV D
002110' 005741                          TST     -(R1)                                   ;REV D
002112' 010167  176520                  MOV     R1,NXTBBK       ;RESET POINTER          ;REV D
002116' 016767  175672  176546          MOV     DV101,DVICE     ;GET DRIVES TO TEST
002124' 123727  000041  000011          CMPB    @#41,#11        ;IF RP IS LOAD MEDIUM THEN
002132' 001021                          BNE     3$              ;BEGIN
002134' 113700  000040                  MOVB    @#40,R0         ;   GET LOAD-DEVICE NUMBER
002140' 012701  000001                  MOV     #1,R1           ;   INITIALIZE DEVICE MASK
002144' 105700                  1$:     TSTB    R0              ;   WHILE NOT POINTING AT LOAD-DEVICE DO
002146' 001403                          BEQ     2$              ;   BEGIN
002150' 006301                          ASL     R1              ;       SHIFT MASK TO NEXT DEVICE
002152' 105300                          DECB    R0              ;       KEEP TRACK OF SHIFTING
002154' 000773                          BR      1$              ;   END
002156' 130167  176510          2$:     BITB    R1,DVICE        ;   IF LOAD DEVICE IS SELECTED THEN
002162' 001405                          BEQ     3$              ;   BEGIN
002164' 113767  000040  176504          MOVB    @#40,UNITNO     ;       MOVE LOAD-DEVICE NUMBER TO DRYVE
002172' 004767  002266                  JSR     PC,DROP         ;       DROP THE LOAD-DEVICE
                                                                ;   END
002176'                         3$:                             ;END
002176' 005067  176442                  CLR     CYL             ;START AT CYLINDER 0                    ;DRB001
002202' 012767  000377  176436          MOV     #377,DSKADR     ;START AT TRACK 0, SECTOR 0             ;DRB001
                                                                ;(* INCREMENT BEFORE TEST *)             ;DRB001
002210' 004767  004500                  JSR     PC,SETUP
002214' 004767  004356                  JSR     PC,REZET        ;CLEAR THE RH
002220' 032767  000200  175570          BIT     #RH70,SR1       ;RH70 ?                                 ;DJM01
002226' 001003                          BNE     RESTRT          ;NO DONT PRI BAE & CS3                  ;DJM01
002230' 012767  177777  176364          MOV     #-1,S+50        ;SHORTEN TEMP STORAGE                   ;DJM01
002236'                         RESTRT:

002236' 104415  000000' 000124'         GETPAS,BEGIN, RBUFVA             ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA

002244' 012767  177777  176424  LOOP1:  MOV     #-1,UNITNO      ;PRE-SET UNIT NUMBER
002252' 104414  000000'                 GWBUFS, BEGIN           ;GET WRITE BUFFER INFORMATION
                                                                        ;**-1                           ;DRB001
002256'                         LOOP2:
002256' 004767  000744          3$:     JSR     PC,PICKDR       ;GO PICK A DRIVE
002262' 103407                          BCS     1$              ;RETURNS HERE IF ALL DRIVES DONE        ;DRB001
002264' 004767  001572                  JSR     PC,WRTLIM       ;GET DISK LIMITS FOR THIS TYPE OF DISK  ;DRB001
002270' 005067  176420                  CLR     TRY             ;ELSE CLR RE TRY COUNT
002274' 004767  000022                  JSR     PC,CYCLE        ;GO DO A CYCLE ON THIS DRIVE
002300' 000766                          BR      LOOP2           ;DO IT TO NEXT DRIVE

002302' 005767  176364          1$:     TST     DVICE           ;ANYBODY LEFT TO CHECK?
002306' 001002                          BNE     2$              ;BR IF YES
002310' 104410  000000'                 ENDS,BEGIN              ;
002314'                         2$:
(1) 002314' 104413  000000'             ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
(1)                                                             ;MONITOR SHALL TEST END OF PASS
002320' 000751                          BR      LOOP1           ;BR BACK IF NO
```



```
002322'                         CYCLE:
                                                                        ;**-4                           ;DRB001
002322' 004767  000772                  JSR     PC,PICKBK       ;SELECT A SECTOR TO TEST                ;DRB001
002326' 004567  000026                  JSR     R5,WRITE        ;GO WRITE A BLOCK
002332' 004567  000116                  JSR     R5,WRITCK       ;GO DO WRITE CHECK
002336' 004567  000206                  JSR     R5,READ         ;GO READ A BLOCK
002342' 104412  000000' 000126'         CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
(1) 002350' 002352'                     .+2                     ; IF ERROR, CONTINUE
002352' 004767  000266                  JSR     PC,RELEASE      ;RELEASE THE DRIVE                      ;DRB001
002356' 000207                          RTS     PC              ;END CYCLE                              ;DRB001
```

```
                                                                        ;**-66          ;DRB001
                                        .MACRO  LINEUP  EABITS  ; LINE UP EA BITS FOR RHCS1
                                ;       LINEUP  EABITS          ; LINE UP EA BITS FOR RHCS1
                                        .NLIST
                                        MOV     EABITS,R0       ; GET EXTENDED MEMORY BITS
                                        ASL     R0              ; SHIFT 4 PLACES TO THE LEFT
                                        ASL     R0              ; TO LINE UP WITH RHCS1
                                        ASL     R0              ;
                                        ASL     R0              ;
                                        MOV     R0,XMEM         ; SAVE THE SHIFTED BITS
                                        .LIST
                                        .ENDM   LINEUP

002360' 012767  000161  176314  WRITE:  MOV     #161,FUNC       ; LOAD WRITE FUNCTION
002366' 012767  002360' 176314          MOV     #WRITE,FERADR   ;SAVE WHERE WE WERE
002374' 016746  175542                  MOV     WBUFSZ,-(SP)    ;GET WRITE SIZE
002400' 005416                          NEG     (SP)            ;NEGATE IT
002402' 012677  177374                  MOV     (SP)+,@RHWC     ; LOAD WORD COUNT
002406' 016777  175522  177370          MOV     WBUFPA,@RHBA    ; LOAD BUFFER ADDRESS
002414' 016777  176226  177364          MOV     DSKADR,@RPDA    ; LOAD DISK ADDRESS
002422' 016777  176214  177404          MOV     CYLNDR,@RPDC    ; LOAD CYLINDER ADDRESS                 ;DRB001
                                ;       LINEUP  WBUFEA          ; LINE UP EA BITS FOR RHCS1
002450' 000167  000302                  JMP     GO              ; CONTINUE
002454' 012767  000151  176220  WRITCK: MOV     #151,FUNC       ; LOAD WRITE-CHECK FUNCTION
002462' 012767  002454' 176220          MOV     #WRITCK,FERADR  ;SAVE WHERE WE WERE
002470' 016746  175446                  MOV     WBUFSZ,-(SP)    ;GET WRITE SIZE
002474' 005416                          NEG     (SP)            ;NEGATE IT
002476' 012677  177300                  MOV     (SP)+,@RHWC     ; LOAD WORD COUNT
002502' 016777  175426  177274          MOV     WBUFPA,@RHBA    ; LOAD BUFFER ADDRESS
002510' 016777  176132  177270          MOV     DSKADR,@RPDA    ; LOAD DISK ADDRESS
002516' 016777  176120  177310          MOV     CYLNDR,@RPDC    ; LOAD CYLINDER ADDRESS                 ;DRB001
                                ;       LINEUP  WBUFEA          ; LINE UP EA BITS FOR RHCS1
002544' 000167  000206                  JMP     GO              ; CONTINUE
002550' 012767  000171  176124  READ:   MOV     #171,FUNC       ; LOAD READ FUNCTION
002556' 012767  002550' 176124          MOV     #READ,FERADR    ;SAVE WHERE WE WERE
002564' 016746  175342                  MOV     RBUFSZ,-(SP)    ;GET READ SIZE
002570' 005416                          NEG     (SP)            ;NEGATE IT
002572' 012677  177204                  MOV     (SP)+,@RHWC     ; LOAD WORD COUNT
002576' 016777  175324  177200          MOV     RBUFPA,@RHBA    ; LOAD BUFFER ADDRESS
002604' 016777  176036  177174          MOV     DSKADR,@RPDA    ; LOAD DISK ADDRESS
002612' 016777  176024  177214          MOV     CYLNDR,@RPDC    ; LOAD CYLINDER ADDRESS                 ;DRB001
                                ;       LINEUP  RBUFEA          ; LINE UP EA BITS FOR RHCS1
002640' 000167  000112                  JMP     GO              ; CONTINUE
002644' 016777  176026  177136  RELEASE: MOV     UNITNO,@RHCS2   ;SET UP UNIT TO RELEASE                 ;DRB001
002652' 012777  000013  177120          MOV     #13,@RHCS1      ;EXECUTE RELEASE COMMAND                ;DRB001
002660' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
(1) 002664' 104407  000000'             BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002670' 000207                          RTS     PC              ;IT'S RELEASED                          ;DRB001
                                                                        ;**-33          ;DRB001

002672' 016777  176000  177110  CLEAR:  MOV     UNITNO,@RHCS2   ; LOAD UNIT ADDRESS
002700' 012777  000011  177072          MOV     #11,@RHCS1      ; ISSUE A DRIVE CLEAR
002706' 000240                          NOP                     ;WAIT
002710' 000240                          NOP                     ;FOR DRIVE CLEAR TO FINISH
002712' 012777  000023  177060          MOV     #23,@RHCS1      ;ISSUE A PACK ACK
```



```
002720' 105777  177054          1$:     TSTB    @RHCS1
002724' 100401                          BMI     2$
002726' 000774                          BR      1$              ; NO, WAIT TILL DONE
002730' 017746  177062          2$:     MOV     @RPAS,-(SP)     ;CLEAR AS BIT
002734' 012677  177056                  MOV     (SP)+,@RPAS
002740' 012777  040000  177032          MOV     #BIT14,@RHCS1   ; CLEAR ANY CONTROLLER ERRORS
002746' 012777  010000  177056          MOV     #BIT12,@RPOF    ; SET BIT FOR 11 FORMAT
002754' 000205                          RTS     R5              ; RETURN

002756' 016777  175714  177024  GO:     MOV     UNITNO,@RHCS2   ; LOAD UNIT SELECT
002764' 032767  001000  175064          BIT     #ADDR22,RCS1    ;22 BIT SUPPORT?                        ;DRB001
002772' 001434                          BEQ     1$              ;NO                                     ;DRB001
002774' 017767  177004  175624          MOV     @RHBA,PA18      ;GET 18 BIT ADDR
003002' 006267  175622                  ASR     XMEM            ;SHIFT EA BITS TO POSITION 4,5
003006' 006267  175616                  ASR     XMEM
003012' 006267  175612                  ASR     XMEM
003016' 006267  175606                  ASR     XMEM
003022' 104416  000000' 000626'         MAP22$, BEGIN,PA18              ; GET 22-BIT ADDR FROM 18-BIT ADDR
003030' 016777  175576  176746          MOV     PA22,@RHBA      ;LOAD BA REG
003036' 016777  175572  177004          MOV     EA22,@RHBAE     ;LOAD BAE REG
003044' 042767  000034  175562          BIC     #34,EA22        ;CLEAR UNWANTED BITS
003052' 000367  175556                  SWAB    EA22            ;LOAD INTO BITS 8,9
003056' 056767  175552  175544          MOV     EA22,XMEM       ;LOAD XMEM TO SET INTO FUNCTION CODE
003064' 056767  175540  175610  1$:     BIS     XMEM,FUNC       ; LOAD EXTENDED MEMORY BITS
003072' 016777  175604  176700          MOV     FUNC,@RHCS1     ; EXECUTE THE FUNCTION
003100' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
003104' 010046                  NTRUPT: MOV     R0,-(SP)        ;SAVE R0
003106' 016700  175564                  MOV     UNITNO,R0
003112' 116000  003310'                 MOVB    BITTAB(R0),R0
003116' 017746  176674                  MOV     @RPAS,-(SP)
003122' 040016                          BIC     R0,(SP)
003124' 012677  176666                  MOV     (R6)+,@RPAS
003130' 012600                          MOV     (SP)+,R0
                                        ;-----------------------------------------------------------------
(1) 003132' 000004  000000' 003140'     PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
(1)                                     ;-----------------------------------------------------------------

003140' 004767  002656          1$:     JSR     PC,ERRORS       ;GO CHECK FOR ERRORS
003144' 103401                          BCS     2$              ;ERRORS DETECTED                        ;DRB001
003146' 000205                          RTS     R5              ;OTHERWISE,  RETURN OK
003150'                         2$:

003150' 105767  175544                  TSTB    HRDERR          ;DON'T RETRY ON HARD ERROR      REV D
003154' 100415                          BMI     4$                                                      ;DRB001
003156' 005267  175532                  INC     TRY             ;COUNT AN ERROR
003162' 026727  175526  000003          CMP     TRY,#3          ;TOO MANY FOR THIS CYCLE?               ;DRB001
003170' 002002                          BGE     5$              ;BR IF SO
003172' 000177  175512                  JMP     @FERADR         ;RE-EXECUTE THE DRIVER ROUTINE          ;DRB001
003176' 104403  000000' 011106' 5$:     MSG$,BEGIN,EXCED
003204' 004767  002664                  JSR     PC,BAD          ;UPDATE BAD BLOCK INFO.
003210' 012605                  4$:     MOV     (SP)+,R5        ;RESTORE R5
                                                                        ;**-7           ;DRB001
003212' 004767  000102          3$:     JSR     PC,PICKBK       ;TRY A DIFFERENT BLOCK
003216' 005367  175454                  DEC     UNITNO          ;WANT TO RE-DO SAME DRIVE WERE ON
003222' 000167  177030                  JMP     LOOP2           ;GO DO IT
                                                                        ;**-2           ;DRB001
```

```
                        ;       -------------------------
                                                                ;**-17                  ;DRB001
003226'                 PICKDR:
003226' 005267 175444           1$:     INC     UNITNO          ;POINT TO NEXT DRIVE
003232' 026727 175440 000010            CMP     UNITNO,#8.              ;DONE LOOKING?
003240' 001002                          BNE     2$              ;BR IF NO, ELSE
003242' 000261                          SEC                     ;SET CARRY FOR NO-MORE-DRIVES   ;DRB001
003244' 000420                          BR      5$              ;EXIT ROUTINE                   ;DRB001
003246' 016700 175424           2$:     MOV     UNITNO,R0       ;USE AS AN INDEX
003252' 136067 003310' 175412           BITB    BITTAB(R0),DVICE        ;TEST THIS DEVICE?
003260' 001001                          BNE     3$              ;BR IF YES, ELSE
003262' 000761                          BR      1$
003264'                 3$:
                                                                ;**-1                   ;DRB001
003264' 004567 003132                   JSR     R5,READY        ;SEE IF DRIVE IS READY
003270' 103402                          BCS     4$              ;BR IF IT WAS READY             ;DRB001
003272' 004767 002730                   JSR     PC,NOTRDY       ;ELSE GO CLEAR IT AND CHECK AGAIN
003276' 004767 000700           4$:     JSR     PC,GETDVT       ;IS THIS LEGAL DRIVE TYPE?      ;DRB001
003302' 103751                          BCS     1$              ;ILLEGAL TYPE: DROPPED.  TRY NEXT ;DRB001
003304' 000241                          CLC                     ;RETURN, NEXT DRIVE AVAILABLE   ;DRB001
003306' 000207                  5$:     RTS     PC              ;GO HOME                        ;DRB001
003310' 001001                  BITTAB: 1001
003312' 004004                          4004
003314' 020020                          20020
003316' 100100                          100100

003320'                 PICKBK:                                 ;**-11                  ;DRB001
                                ;(* FUNCTION: SELECT NEXT DISK ADDRESS TO TEST.       *)  ;DRB001
                                ;(* AVOID WRITING OVER KNOWN BAD SPOTS OR DISK'S BAD  *)  ;DRB001
                                ;(* SPOT FILE (IF OPERATOR TELLS US IT'S THERE).  CYL *)  ;DRB001
                                ;(* IS "RELATIVE ADDRESS", MAX RANGE 0-410 (RP04/5) OR*)  ;DRB001
                                ;(* 0-407 (RP06);  THIS IS MOVED TO CYLNDR AND INCRE- *)  ;DRB001
                                ;(* MENTED TO SECOND HALF IF THIS COPY OF RPD IS IN   *)  ;DRB001
                                ;(* PORT B MODE.                                     *)  ;DRB001
                                                                                        ;DRB001
003320' 032767 000040 174470            BIT     #NORAND,SR1     ;IF SEQUENTIAL SEEKS THEN       ;DRB001
003326' 001403                          BEQ     1$              ;BEGIN                          ;DRB001
003330' 105267 175312                   INCB    SECTOR          ; SECTOR := SECTOR + 1          ;DRB001
003334' 000432                          BR      2$              ;END                            ;DRB001
003336'                 1$:                                     ;ELSE (* RANDOM SEEKS *)        ;DRB001
                                                                ;BEGIN (* FUDGE 3 UN-RELATED    ;DRB001
                                                                ;  (* NUMBERS FROM 1 RAND CALL  *)  ;DRB001
003336' 104417 000000'                  RAND$,BEGIN
003342' 016700 174506                   MOV     RANNUM,R0       ; R0 := RANNUM                  ;DRB001
003346' 042700 176000                   BIC     #176000,R0      ; R0 := R0 MOD 1024.            ;DRB001
003352' 010067 175266                   MOV     R0,CYL          ; CYL := R0                     ;DRB001
003356' 016700 174472                   MOV     RANNUM,R0       ; "RECHARGE" R0                 ;DRB001
003362' 000300                          SWAB    R0              ; R0<0:8> := R0<8:8>            ;DRB001
003364' 042700 177740                   BIC     #177740,R0      ; R0 := R0 MOD 32.              ;DRB001
003370' 110067 175252                   MOVB    R0,SECTOR       ; SECTOR := R0                  ;DRB001
003374' 016700 174454                   MOV     RANNUM,R0       ; AND AGAIN                     ;DRB001
003400' 000241                          CLC                     ; CARRY := 0                    ;DRB001
```



```
003402' 006100                          ROL     R0              ; CARRY := R0<15:1>             ;DRB001
003404' 006100                          ROL     R0                                              ;DRB001
003406' 006100                          ROL     R0              ;                               ;DRB001
003410' 006100                          ROL     R0              ;                               ;DRB001
003412' 042700 177740                   BIC     #177740,R0      ; R0 := (R0<0:13>*8.) + R0<13:3> ;DRB001
                                                                ; (* NOTE: THE FORMULAE USED ARE *)  ;DRB001
                                                                ; (* ABSOLUTELY IRRELEVANT      *)  ;DRB001
003416' 110067 175225                   MOVB    R0,TRACK        ; TRACK := R0                   ;DRB001
003422'                 2$:                                     ;END                            ;DRB001
003422' 126727 175220 000025            CMPB    SECTOR,#21.     ;IF SECTOR GT 21. THEN          ;DRB001
003430' 003412                          BLE     3$              ;BEGIN                          ;DRB001
003432' 105267 175211                   INCB    TRACK           ; TRACK := TRACK + 1            ;DRB001
003436' 016700 175204                   MOV     SECTOR,R0       ; R0 := SECTOR                  ;DRB001
003442' 012701 000026                   MOV     #22.,R1         ; R1 := # OF SECTORS IN TRACK   ;DRB001
003446' 004767 000760                   JSR     PC,MODLUS       ; R0 := R0 MOD R1               ;DRB001
003452' 110067 175170                   MOVB    R0,SECTOR       ; SECTOR := R0                  ;DRB001
003456'                 3$:                                     ;END                            ;DRB001
                                ;(* NOW, CUT VALUES DOWN TO LEGAL RANGES FOR CURRENT DRIVE TYPE *) ;DRB001
003456' 126727 175165 000022            CMPB    TRACK,#18.      ;IF TRACK GT 18. THEN           ;DRB001
003464' 003412                          BLE     4$              ;BEGIN                          ;DRB001
003466' 005267 175152                   INC     CYL             ; CYL := CYL + 1                ;DRB001
003472' 116700 175151                   MOVB    TRACK,R0        ; R0 := TRACK                   ;DRB001
003476' 012701 000023                   MOV     #19.,R1         ; R1 := 19.                     ;DRB001
003502' 004767 000724                   JSR     PC,MODLUS       ; R0 := R0 MOD R1               ;DRB001
003506' 110067 175135                   MOVB    R0,TRACK        ; TRACK := R0                   ;DRB001
003512'                 4$:                                     ;END                            ;DRB001
003512' 026767 175126 175120            CMP     CYL,CYLLIM      ;IF CYL GT # OF CYLINDERS IN RANGE THEN ;DRB001
003520' 002420                          BLT     5$              ;BEGIN                          ;DRB001
003522' 032767 000040 174266            BIT     #NORAND,SR1     ; IF SEQUENTIAL SEEKS THEN      ;DRB001
003530' 001404                          BEQ     6$              ; BEGIN                         ;DRB001
003532' 005067 175106                   CLR     CYL             ;  RESET CYLINDER               ;DRB001
003536' 005067 175104                   CLR     DSKADR          ;  TRACK := 0, SECTOR := 0      ;DRB001
003542'                 6$:                                     ; ELSE                          ;DRB001
                                                                ; BEGIN                         ;DRB001
003542' 016700 175076                   MOV     CYL,R0          ;  R0 := CYL                    ;DRB001
003546' 016701 175066                   MOV     CYLLIM,R1       ;  R1 := CYLLIM                 ;DRB001
003552' 004767 000654                   JSR     PC,MODLUS       ;  R0 := R0 MOD R1              ;DRB001
003556' 010067 175062                   MOV     R0,CYL          ;  CYL := R0                    ;DRB001
003562'                 7$:                                     ; END                           ;DRB001
003562'                 5$:                                     ;END                            ;DRB001
003562' 016767 175056 175052            MOV     CYL,CYLNDR      ;CYLNDR := CYL                  ;DRB001
003570' 032767 000020 174220            BIT     #BPORT,SR1      ;IF BPORT AND DUAL PORT THEN
003576' 001407                          BEQ     8$              ;BEGIN                          ;DRB001
003600' 032767 000001 175062            BIT     #BIT0,FLAG      ;                               ;DRB001
003606' 001403                          BEQ     8$              ;                               ;DRB001
003610' 066767 175024 175024            ADD     CYLLIM,CYLNDR   ; INCREMENT TO SECOND HALF OF DISK ;DRB001
003616'                 8$:                                     ;END                            ;DRB001
003616' 026767 175020 175032            CMP     CYLNDR,LASCYL   ;IF CYLNDR GT LASCYL THEN       ;DRB001
003624' 003431                          BLE     20$             ;BEGIN          ;REV. D0
003626' 016767 175024 175006            MOV     LASCYL,CYLNDR   ; CYLNDR := LASCYL (* TRUNCATE *) ;DRB001
003634' 032767 000040 174154            BIT     #NORAND,SR1     ; IF SEQUENTIAL THEN            ;DRB001
003642' 001404                          BEQ     10$             ; BEGIN                         ;DRB001
003644' 005067 174774                   CLR     CYL             ;  RESET CYLINDER               ;DRB001
003650' 005067 174772                   CLR     DSKADR          ;  TRACK := 0, SECTOR := 0      ;DRB001
003654'                 10$:                                    ; END                           ;DRB001
003654' 126767 174767 174776            CMPB    TRACK,LASTRK    ; IF TRACK GT LASTRKK THEN      ;DRB001
```

```
                        BLE     11$             ; BEGIN                                         ;DRB001
003662' 003412          MOVB    LASTRK,TRACK    ;  TRACK := LASTRK                              ;DRB001
003664' 116767 174770 174755  CMPB    SECTOR,LASSEC   ;  IF SECTOR GT LASSEC THEN             ;DRB001
003672' 126767 174750 174762  BLE     12$             ;  BEGIN                                ;DRB001
003700' 003403          MOVB    LASSEC,SECTOR   ;    SECTOR := LASSEC                          ;DRB001
003702' 116767 174754 174736  12$:                   ;  END                                  ;DRB001
003710'                 11$:                    ; END                                           ;DRB001

003710'                 20$:                                                                    ;DRB001
                                                                                                ;DRB001
                        ;(* NOW, CHECK BADSPOT TABLE *)                                        ;DRB001
003710' 012700 000252'  MOV     #BADSPT,R0      ;R0 := ADDRESS OF TABLE                         ;DRB001
003714' 016701 174716   MOV     NXTBBK,R1       ;R1 := END OF TABLE                             ;DRB001
003720' 116702 174722   MOVB    SECTOR,R2       ;R2 := SECTOR                                   ;DRB001
003724' 116703 174717   MOVB    TRACK,R3        ;R3 := TRACK                                    ;DRB001
003730' 016704 174706   MOV     CYLNDR,R4       ;R4 := CYLNDR                                   ;DRB001
                        ;(* CALCULATE UPPER LIMIT OF THIS WRITE *)                             ;DRB001
003734' 066702 174726   ADD     SIZSEC,R2       ;R2 := R2 + # OF SECTORS IN WRITE               ;DRB001
003740' 020227 000025   CMP     R2,#21.         ;IF R2 GT 21. THEN                              ;DRB001
003744' 003403          BLE     13$             ;BEGIN                                          ;DRB001
003746' 005203          INC     R3              ; R3 := R3 + 1 (INCR. UPPER TRACK)              ;DRB001
003750' 162702 000026   SUB     #22.,R2         ; TRUNCATE R2                                   ;DRB001
003754'                 13$:                    ;END                                            ;DRB001
003754' 066703 174704   ADD     SIZTRK,R3       ;R3 := R3 + TRKSIZ                              ;DRB001
003760' 020327 000022   CMP     R3,#18.         ;IF R3 GT 18. THEN                              ;DRB001
003764' 003403          BLE     14$             ;BEGIN                                          ;DRB001
003766' 005204          INC     R4              ; R4 := R4 + 1 (INCR. UPPER CYLINDER)           ;DRB001
003770' 162703 000023   SUB     #19.,R3         ; TRUNCATE R3                                   ;DRB001
003774'                 14$:                    ;END                                            ;DRB001
003774' 020001          15$:    CMP     R0,R1           ;WHILE R0 LT R1 DO                      ;DRB001
003776' 002030          BGE     16$             ;BEGIN                                          ;DRB001
004000' 026710 174636   CMP     CYLNDR,(R0)     ; IF CYLNDR LE (R0) AND R4 GE (R0)              ;DRB001
004004' 003022          BGT     17$             ;  AND TRACK LE 3(R0) AND R3 GE 3(R0)           ;DRB001
004006' 020410          CMP     R4,(R0)         ;  AND SECTOR LE 2(R0) AND R2 GE 2(R0) THEN;DRB001
004010' 002420          BLT     17$             ;                                               ;DRB001
004012' 126760 174631 000003  CMPB    TRACK,3(R0)     ;                                       ;DRB001
004020' 003014          BGT     17$                                                             ;DRB001
004022' 120360 000003   CMPB    R3,3(R0)        ;                                               ;DRB001
004026' 002411          BLT     17$             ;                                               ;DRB001
004030' 126760 174612 000002  CMPB    SECTOR,2(R0)    ;                                       ;DRB001
004036' 003005          BGT     17$             ;                                               ;DRB001
004040' 120260 000002   CMPB    R2,2(R0)        ;                                               ;DRB001
004044' 002402          BLT     17$             ; BEGIN                                         ;DRB001
004046' 000167 177246   JMP     PICKBK          ;  TRY ANOTHER SECTOR                           ;DRB001
004052'                 17$:                    ; END                                           ;DRB001
004052' 062700 000004   ADD     #4,R0           ; INCREMENT TO NEXT BAD SPOT                    ;DRB001
004056' 000746          BR      15$             ;END                                            ;DRB001
004060'                 16$:                    ;                                               ;DRB001
004060' 000207          RTS     PC              ;CPU GO HOME!!!!                                ;DRB001
                                                                                                ;DRB001
                                                                                                ;DRB001
004062'                 WRTLIM:                                                                 ;DRB001
                        ;(* WE ALLOW USER TO CHANGE WBUFSZ "AT WILL"--THEREFORE *)             ;DRB001
                        ;(* WE DO NOT AUTOMATICALLY KNOW THE TRANSFER SIZE.    *)             ;DRB001
                        ;(* COMPUTATION IS MADE EASIER, FURTHERMORE, IF WE HAVE *)             ;DRB001
                        ;(* THE SIZE BROKEN INTO TRACKS AND SECTORS (SINCE      *)             ;DRB001
```



```
                        ;(* WBUFSZ IS ONE WORD, THE TRANSFER CAN'T BE GREATER   *)             ;DRB001
                        ;(* THAN 12 TRACKS).  THIS ROUTINE ALSO CALCULATES THE  *)             ;DRB001
                        ;(* UPPER LIMITS ON DISK WRITES SO AS NOT TO OVERFLOW   *)             ;DRB001
                        ;(* HIGH CYLINDER.                                      *)             ;DRB001
                                                                                                ;DRB001
004062' 016700 174054   MOV     WBUFSZ,R0       ;R0 := ALLOCATED WRITE BUFFER                   ;DRB001
004066' 000300          SWAB    R0              ;R0 := R0 / 256.                                ;DRB001
004070' 042700 177400   BIC     #177400,R0      ; (* R0 IS WBUFSZ IN SECTORS *)                ;DRB001
004074' 005067 174564   CLR     SIZTRK          ;SIZTRK := 0                                    ;DRB001
004100' 020027 000026   1$:     CMP     R0,#22.         ;WHILE R0 GE 22. DO                     ;DRB001
004104' 002405          BLT     2$              ;BEGIN                                          ;DRB001
004106' 105267 174552   INCB    SIZTRK          ; SIZTRK := SIZTRK + 1                          ;DRB001
004112' 162700 000026   SUB     #22.,R0         ; DECREMENT SECTORS BY 1 TRACK                  ;DRB001
004116' 000770          BR      1$              ;END                                            ;DRB001
004120' 010067 174542   2$:     MOV     R0,SIZSEC       ;SIZSEC := REMAINDER                    ;DRB001
004124' 012701 000022   MOV     #18.,R1         ;R1 := HIGH TRACK                               ;DRB001
004130' 012702 000025   MOV     #21.,R2         ;R2 := HIGH SECTOR                              ;DRB001
004134' 160002          SUB     R0,R2           ;R2 := R2 - SIZSEC                              ;DRB001
004136' 005702          TST     R2              ;IF R2 LT 0 THEN                                ;DRB001
004140' 002003          BGE     3$              ;BEGIN                                          ;DRB001
004142' 062702 000026   ADD     #22.,R2         ; INCREMENT BACK UP                             ;DRB001
004146' 005301          DEC     R1              ; CAN'T FIT THAT LAST TRACK...                  ;DRB001
004150'                 3$:                     ;END                                            ;DRB001
004150' 166701 174510   SUB     SIZTRK,R1       ;R1 := R1 - SIZTRK                              ;DRB001
004154' 005701          TST     R1              ;IF R1 LT 0 THEN                                ;DRB001
004156' 002004          BGE     4$              ;BEGIN                                          ;DRB001
004160' 062701 000023   ADD     #19.,R1         ; INCREMENT R1 BACK UP                          ;DRB001
004164' 005367 174466   DEC     LASCYL          ; LASCYL := LASCYL - 1                          ;DRB001
004170'                 4$:                     ;END                                            ;DRB001
004170' 010167 174464   MOV     R1,LASTRK       ;LASTRK := HIGHEST TRACK WE CAN WRITE           ;DRB001
004174' 010267 174462   MOV     R2,LASSEC       ;LASSEC := HIGHEST SECTOR WE CAN WRITE          ;DRB001
004200' 000207          RTS     PC              ;END OF ROUTINE                                 ;DRB001
004202'                 GETDVT:                                                                 ;DRB001
                        ;(* FIND WHICH TYPE OF DRIVE THIS IS.  IF RP04/5, SET CYLLIM *)        ;DRB001
                        ;(* TO 206.  IF RP06, SET CYLLIM TO 408.  DROP THE DRIVE IF  *)        ;DRB001
                        ;(* IT IS NOT A RP04/5/6                                    *)        ;DRB001
                                                                                                ;DRB001
004202' 016777 174470 175600  MOV     UNITNO,@RHCS2   ;SELECT THE DRIVE                       ;DRB001
004210' 027727 175612 024022  CMP     @RPD1,#24022    ;IF IT IS AN RP06 DUAL PORT THEN        ;DRB001
004216' 001012          BNE     1$              ;BEGIN                                          ;DRB001
004220' 052767 000001 174442  BIS     #BIT0,FLAG      ; DUAL PORT                             ;DRB001
004226' 012767 000627 174404  MOV     #407.,CYLLIM    ; CYLLIM := 408.                        ;DRB001
004234' 012767 001455 174414  MOV     #813.,LASCYL    ; LASCYL := DISK HIGH LIM               ;DRB001
004242' 000471          BR      2$              ;END                                            ;DRB001
004244'                 1$:                     ;ELSE                                           ;DRB001
004244' 027727 175556 024021  CMP     @RPDT,#24021    ;IF IT IS RP04/5 DUAL PORT THEN         ;DRB001
004252' 001404          BEQ     3$              ;                                               ;DRB001
004254' 027727 175546 024020  CMP     @RPDT,#24020    ;                                       ;DRB001
004262' 001012          BNE     4$              ;BEGIN                                          ;DRB001
004264'                 3$:                                                                     ;DRB001
004264' 052767 000001 174376  BIS     #BIT0,FLAG      ; DUAL PORT                             ;DRB001
004272' 012767 000315 174340  MOV     #205.,CYLLIM    ; CYLLIM := 206.                        ;DRB001
004300' 012767 000631 174350  MOV     #409.,LASCYL    ; LASCYL := DISK HIGH LIM               ;DRB001
004306' 000447          BR      5$              ;END                                            ;DRB001
004310'                 4$:                     ;ELSE                                           ;DRB001
004310' 027727 175512 020022  CMP     @RPDT,#20022    ;IF RP06 SINGLE PORT THEN               ;DRB001
```

```
                                        BNE     6$                      ;BEGIN                                  ;DRB001
                                        BIC     #BIT0,FLAG              ; SINGLE PORT                           ;DRB001
                                        MOV     #814.,CYLLIM            ; CAN USE ENTIRE DISK                   ;DRB001
                                        MOV     #813.,LASCYL            ; LASCYL := DISK HIGH LIMIT             ;DRB001
                                        BR      4$                      ;END                                    ;DRB001
                                6$:                                     ;ELSE                                   ;DRB001
                                        CMP     @RPDT,#20021            ;IF RP04/5 SINGLE PORT THEN             ;DRB001
                                        BEQ     8$                      ;                                       ;DRB001
                                        CMP     @RPDT,#20020            ;                                       ;DRB001
                                        BNE     9$                      ;                                       ;DRB001
                                8$:                                     ;BEGIN                                  ;DRB001
                                        BIC     #BIT0,FLAG              ; SINGLE PORT                           ;DRB001
                                        MOV     #410.,CYLLIM            ; USE ENTIRE DISK                       ;DRB001
                                        MOV     #409.,LASCYL            ; LASCYL := DISK HIGH LIMIT             ;DRB001
                                        BR      2$                      ;END                                    ;DRB001
                                9$:                                     ;ELSE                                   ;DRB001
                                                                        ;BEGIN                                  ;DRB001
                                        JSR     PC,DROP                 ; DROP THE DRIVE                        ;DRB001
                                        MSGN$,BEGIN,BADTYP              ;ASCII MESSAGE CALL WITH COMMON HEADER
                                        SEC                             ; ERROR RETURN                          ;DRB001
                                        BR      7$                      ;                                       ;DRB001
                                5$:                                     ;END                                    ;DRB001
                                2$:                                     ;                                       ;DRB001
                                        CLC                             ;NO ERROR                               ;DRB001
                                7$:                                     ;                                       ;DRB001
                                        RTS     PC                      ;RETURN                                 ;DRB001
                                MODLUS:                                                                         ;DRB001
                                        ;(* THIS LITTLE ROUTINE COMPUTES R0 MODULUS R1       *)              ;DRB001
                                        ;(*  I.E., THE REMAINDER OF R0/R1, AND RETURNS THE  *)              ;DRB001
                                        ;(* ANSWER IN R0.                                   *)              ;DRB001
                                1$:     CMP     R0,R1                   ;WHILE R0 GE R1 DO                      ;DRB001
                                        BLT     2$                      ;BEGIN                                  ;DRB001
                                        SUB     R1,R0                   ; R0 := R0-R1                           ;DRB001
                                        BR      1$                      ;END                                    ;DRB001
                                2$:                                     ;                                       ;DRB001
                                        RTS     PC                      ;THAT'S ALL                             ;DRB001

                                CLRRB:  MOV     #RBUF,R0                ;CLEAR RBUF BUFFER
                                        MOV     RBUFSZ,R1               ;GET ITS ADDR AND SIZE
                                CLRCOM: CLR     (R0)+                   ;CLEAR ANOTHER
                                        DEC     R1                      ;COUNT ANOTHER
                                        BNE     CLRCOM                  ;BR BACK TILL DONE
                                        RTS     PC

                                DROP:   MOV     #1,R1                   ; INITIALIZE DROP PICKER
                                        MOV     UNITNO,R0               ; GET THE DRIVE NUMBER
                                        BEQ     4$                      ; IF DRIVE 0 GO DROP IT
                                1$:     ASL     R1                      ; POINT TO NEXT DRIVE
                                        DEC     R0                      ; IS THIS THE ONE ?
                                        BNE     1$                      ; NO, LOOK AGAIN
                                2$:     BIC     R1,DVICE                ; DROP THE DRIVE
                                        ;***************************************************************
(1)                                                                     ;CONVERT UNITNO TO ASCII AND
(1)                                                                     ;STORE AT ADR1
(1)                                     OTOA$,BEGIN,UNITNO,ADR1
(1)
```



```
(1)                                     ;***************************************************************
                                        RTS     PC                      ; RETURN
                                ;       ------------------------


                                        ;////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\
                                        ;////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\

                                                ;R E V  D       ;MCP




                                ERDR:   CLR     DLTFLG                  ;INITIALIZE DLT FLAG
                                        MSGN$,BEGIN,ASTER               ;ASCII MESSAGE CALL WITH COMMON HEADER
                                        CLR     HRDERR                          ;RESET ERROR INDICATOR
                                        BIT     #BIT15,S+10                     ;
                                        BEQ     14$                             ;
                                        INC     DLTCNT                          ;
                                        BIS     #1,DLTFLG                       ;SET FLAG FOR DATA LATE
                                        BIT     #DLATE,SR1                      ;
                                        BNE     14$                             ;
                                        BIS     #-1,HRDERR                      ;
                                        MSGN$,BEGIN,DLTERR              ;ASCII MESSAGE CALL WITH COMMON HEADER
                                14$:    BIT     #BIT13,@HRCS1                   ;MASSBUS CONTROL PARITY ERR?
                                        BEQ     5$
                                        BIS     #-1,HRDERR                      ;SET HARD ERR
                                        MSGN$,BEGIN,MCPERR              ;ASCII MESSAGE CALL WITH COMMON HEADER
                                5$:     BIT     #HRDCS2,S+10                    ;ANY ERRORS ?
                                        BEQ     1$
                                        JSR     PC,CHKCS2                       ;GO SEE THE BITS
                                1$:     BIT     #BIT14,S+12                     ;ANY SET IN THE ERROR REG'S ?
                                        BEQ     4$
                                        BIT     #HRDER1,S+14
                                        BEQ     2$
                                        JSR     PC,CHKER1
                                2$:     BIT     #HRDER2,S+40                    ;ANY ERRORS IN ER2
                                        BEQ     3$
                                        BIS     #-1,HRDERR
                                        MSGN$,BEGIN,ER2ERR              ;ASCII MESSAGE CALL WITH COMMON HEADER
                                3$:     BIT     #HRDER3,S+42                    ;TEST ER3
                                        BEQ     4$
                                        BIS     #-1,HRDERR                      ;SET HARD ERRROR INDICATOR
                                        MSGN$,BEGIN,ER3ERR              ;ASCII MESSAGE CALL WITH COMMON HEADER
                                4$:     BIT     #BIT14,S+10                     ; WCE ?
                                        BEQ     6$
                                        BIS     #2,DLTFLG                       ;SET FLAG FOR SOFT TO BE USED IF DLT ALSO OCCURE
                                        MSGN$,BEGIN,WCEERR              ;ASCII MESSAGE CALL WITH COMMON HEADER
                                6$:     BIT     #BIT14,S                        ;TRE  ?
                                        BEQ     7$
                                        BIS     #2,DLTFLG                       ;SET SOFT FLAG
                                        MSGN$,BEGIN,TRERR               ;ASCII MESSAGE CALL WITH COMMON HEADER
                                7$:     BIT     #BIT15,S                        ;SC ?
```

```
  993  005020' 001406                          BEQ     8$
  994  005022' 052767  000002  000124          BIS     #2,DLTFLG               ;SET FLAG
  995  005030' 104403  000000' 011224'         MSGN$,BEGIN,SCERR       ;ASCII MESSAGE CALL WITH COMMON HEADER
  996  005036' 005767  173520          8$:     TST     5+10
  997  005042' 100405                          BMI     10$
  998  005044' 016767  174752  173644          MOV     RHDB,STORE
  999  005052' 005067  174744                  CLR     RHDB
 1000  005056' 005767  173636          10$:    TST     HRDERR                  ;ANNY HARD ERROR'S ?
 1001  005062' 100414                          BMI     12$                     ;IF TRUE THEN BR
 1002  005064' 032767  000001  000062          BIT     #1,DLTFLG               ;DLT ?  #DLATE IN SR1 MUST BE SET IF TRUE
 1003  005072' 001404                          BEQ     22$
 1004  005074' 022767  000003  000052          CMP     #3,DLTFLG               ;I; (SOFT & DLT NOT COUNTED )  ?
 1005  005102' 001007                          BNE     11$
 1006  005104'                         22$:
 1007                                          ;*****************************************************************
  (1)  005104' 104406  000000' 001722'         SOFER$,BEGIN,TABLE      ;
  (1)                                          ;*****************************************************************
 1008  005112' 000403                          BR      11$                     ;
 1009  005114'                         12$:
  (1)                                          ;*****************************************************************
  (1)  005114' 104405  000000' 001722'         HRDER$,BEGIN,TABLE      ;
  (1)                                          ;*****************************************************************
 1010  005122' 005767  174674          11$:    TST     RHDB
 1011  005126' 001003                          BNE     9$
 1012  005130' 016767  173562  174664          MOV     STORE,RHDB
 1013  005136' 004767  001064          9$:     JSR     PC,NOTRDY
 1014  005142' 000261                          SEC
 1015
 1016  005144' 104403  000000' 011360'         MSGN$,BEGIN,ASTER       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1017  005152' 000207                          RTS     PC
 1018
 1019  005154' 000000                  DLTFLG: .WORD   0               ;DLT FLAG TO BE USED FOR SOFT ERROR
 1020                                                                  ; & INHIBIT DLT COMPARISONS
 1021
 1022  005156' 052767  177777  173534  CHKCS2: BIS     #-1,HRDERR              ;SET HARD ERROR
 1023  005164' 104403  000000' 011230'         MSGN$,BEGIN,CS2HD       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1024  005172' 032767  020000  173362  1$:     BIT     #BIT13,S+10             ;UPE ?
 1025  005200' 001403                          BEQ     2$
 1026  005202' 104403  000000' 011234'         MSGN$,BEGIN,UPER        ;ASCII MESSAGE CALL WITH COMMON HEADER
 1027  005210' 032767  010000  173344  2$:     BIT     #BIT12,S+10             ;NED ?
 1028  005216' 001403                          BEQ     3$
 1029  005220' 104403  000000' 011240'         MSGN$,BEGIN,NEDER       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1030  005226' 032767  004000  173326  3$:     BIT     #BIT11,S+10             ;NXM ?
 1031  005234' 001403                          BEQ     4$
 1032  005236' 104403  000000' 011244'         MSGN$,BEGIN,NXM ;ASCII MESSAGE CALL WITH COMMON HEADER
 1033  005244' 032767  002000  173310  4$:     BIT     #BIT10,S+10
 1034  005252' 001403                          BEQ     5$
 1035  005254' 104403  000000' 011250'         MSGN$,BEGIN,PGERR       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1036  005262' 032767  001000  173272  5$:     BIT     #BIT9,S+10              ;MXF ?
 1037  005270' 001403                          BEQ     6$
 1038  005272' 104403  000000' 011254'         MSGN$,BEGIN,MXFER       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1039
 1040  005300' 032767  000400  173254  6$:     BIT     #BIT8,S+10
 1041  005306' 001403                          BEQ     7$
 1042  005310' 104403  000000' 011260'         MSGN$,BEGIN,MDPER       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1043  005316' 000207                  7$:     RTS     PC
```



```
 1044
 1045  005320'                         CHKER1:
  (1)  005320' 104403  000000' 011264'         MSGN$,BEGIN,ER1HD       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1046  005326' 032767  100000  173232          BIT     #BIT15,S+14             ;DATA CHECK ERROR ?
 1047  005334' 001406                          BEQ     1$
 1048  005336' 052767  000002  177610          BIS     #2,DLTFLG               ;SET SOFT ERR
 1049  005344' 104403  000000' 011270'         MSGN$,BEGIN,DCKER       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1050  005352' 032767  000020  173206  1$:     BIT     #BIT4,S+14              ;FORMAT ERROR ?
 1051  005360' 001423                          BEQ     2$
 1052  005362' 104403  000000' 011274'         MSGN$,BEGIN,FERER       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1053  005370' 016767  173172  173320          MOV     S+14,STORE      ;SAVE ER1
 1054  005376' 042767  077757  173312          BIC     #77757,STORE
 1055  005404' 022767  100020  173304          CMP     #BIT15 ! BIT4,STORE
 1056  005412' 001003                          BNE     21$
 1057  005414' 052767  177777  173276          BIS     #-1,HRDERR              ;SET HARD ERROR INDICATOR
 1058  005422' 052767  000002  177524  21$:    BIS     #2,DLTFLG               ;SET SOFT ERR
 1059
 1060  005430' 032767  000400  173130  2$:     BIT     #BIT8,S+14              ;HCRC ?
 1061  005436' 001406                          BEQ     3$
 1062  005440' 052767  000002  177506          BIS     #2,DLTFLG               ;SET SOFT ERR
 1063  005446' 104403  000000' 011300'         MSGN$,BEGIN,HCRC        ;ASCII MESSAGE CALL WITH COMMON HEADER
 1064  005454' 032767  000200  173104  3$:     BIT     #BIT7,S+14              ;HEADER COMP ERR ?
 1065  005462' 001420                          BEQ     5$
 1066  005464' 104403  000000' 011304'         MSGN$,BEGIN,HCE ;ASCII MESSAGE CALL WITH COMMON HEADER
 1067  005472' 016767  173070  173216          MOV     S+14,STORE
 1068  005500' 042767  177177  173210          BIC     #177177,STORE           ;SAVE ONLY HCRC & HCE
 1069  005506' 022767  000600  173202          CMP     #600,STORE
 1070  005514' 001403                          BEQ     5$                      ;SOFT ERROR IF BOTH SET
 1071  005516' 052767  177777  173174          BIS     #-1,HRDERR              ;SET HRD ERR .
 1072  005524' 032767  077157  173034  5$:     BIT     #77157,S+14             ;ONLY HARD'S LEFT
 1073  005532' 001001                          BNE     6$
 1074  005534' 000207                          RTS     PC
 1075
 1076  005536' 052767  177777  173154  6$:     BIS     #-1,HRDERR              ;SET HRD ERR .
 1077  005544' 032767  040000  173014          BIT     #BIT14,S+14             ;UNS ?
 1078  005552' 001403                          BEQ     7$
 1079  005554' 104403  000000' 011310'         MSGN$,BEGIN,UNSER       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1080  005562' 032767  020000  172776  7$:     BIT     #BIT13,S+14             ;OPI ?
 1081  005570' 001403                          BEQ     8$
 1082  005572' 104403  000000' 011314'         MSGN$,BEGIN,OPIER       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1083  005600' 032767  010000  172760  8$:     BIT     #BIT12,S+14             ;DTE ?
 1084  005606' 001403                          BEQ     9$
 1085  005610' 104403  000000' 011320'         MSGN$,BEGIN,DTER        ;ASCII MESSAGE CALL WITH COMMON HEADER
 1086  005616' 032767  004000  172742  9$:     BIT     #BIT11,S+14             ;WLE ?
 1087  005624' 001403                          BEQ     10$
 1088  005626' 104403  000000' 011324'         MSGN$,BEGIN,WLER        ;ASCII MESSAGE CALL WITH COMMON HEADER
 1089  005634' 032767  002000  172724  10$:    BIT     #BIT10,S+14             ;IAE ?
 1090  005642' 001403                          BEQ     20$
 1091  005644' 104403  000000' 011330'         MSGN$,BEGIN,IAER        ;ASCII MESSAGE CALL WITH COMMON HEADER
 1092  005652' 032767  001000  172706  20$:    BIT     #BIT9,S+14              ;               >
 1093  005660' 001403                          BEQ     17$                     ;               >
 1094  005662' 104403  000000' 011364'         MSGN$,BEGIN,AOE ;ASCII MESSAGE CALL WITH COMMON HEADER
 1095  005670' 032767  000040  172670  17$:    BIT     #BIT5,S+14              ;
 1096  005676' 001403                          BEQ     11$                     ;
 1097  005700' 104403  000000' 011370'         MSGN$,BEGIN,WCF ;ASCII MESSAGE CALL WITH COMMON HEADER
 1098  005706' 032767  000100  172652  11$:    BIT     #BIT6,S+14              ;ECH ?
```

```
1099  005714' 001405                          BEQ     12$
1100  005716' 104403  000000' 011334'         MSGN$,BEGIN,ECHR        ;ASCII MESSAGE CALL WITH COMMON HEADER
1101  005724' 004767  000144                  JSR     PC,BAD
1102  005730' 032767  000010  172630  12$:    BIT     #BIT3,S+14              ;PAR  ?
1103  005736' 001403                          BEQ     13$
1104  005740' 104403  000000' 011340'         MSGN$,BEGIN,PARR        ;ASCII MESSAGE CALL WITH COMMON HEADER
1105  005746' 032767  000004  172612  13$:    BIT     #BIT2,S+14              ;RMR ?
1106  005754' 001403                          BEQ     14$
1107  005756' 104403  000000' 011344'         MSGN$,BEGIN,RMRR        ;ASCII MESSAGE CALL WITH COMMON HEADER
1108  005764' 032767  000002  172574  14$:    BIT     #BIT1,S+14              ;ILR ?
1109  005772' 001403                          BEQ     15$
1110  005774' 104403  000000' 011350'         MSGN$,BEGIN,ILRR        ;ASCII MESSAGE CALL WITH COMMON HEADER
1111  006002' 032767  000001  172556  15$:    BIT     #BIT0,S+14              ;ILF ?
1112  006010' 001403                          BEQ     16$
1113  006012' 104403  000000' 011354'         MSGN$,BEGIN,ILFR        ;ASCII MESSAGE CALL WITH COMMON HEADER
1114  006020' 000207                  16$:    RTS     PC
1115
1116                                          ;       R E V  D   END OF UPDATE
1117
1118
1119                                          ;////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\
1120                                          ;////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\
1121
1125
1126
1127  006022'                         ERRORS:                                                 ;DRB001
1128  006022' 005777  173752                  TST     @RHCS1          ;IF NO ERROR THEN       ;DRB001
1129  006026' 100402                          BMI     1$              ;BEGIN                  ;DRB001
1130  006030' 000241                          CLC                     ; SET NO ERROR          ;DRB001
1131  006032' 000207                          RTS     PC
1132
1133  006034'                         1$:                             ;ELSE                   ;DRB001
1134  006034' 000261                          SEC                     ;SET C BIT TO INDICATE ERROR
1135  006036' 005000                          CLR     R0              ;INIT INDEX
1136  006040' 016701  173734                  MOV     RHCS1,R1        ; BEGIN OF REGISTER POINTERS    ;DRB001
1137  006044'                         3$:                             ; REPEAT                ;DRB001
1138                                                                  ; BEGIN                 ;DRB001
1139  006044' 012160  000552'                 MOV     (R1)+,S(R0)     ;  MOV REGISTER TO TEMPORARY    ;DRB001
1140  006050' 005720                          TST     (R0)+           ;  ADVANCE POINTER      ;DRB001
1141  006052' 022760  177777  000552'         CMP     #-1,S(R0)               ;END ?    REV D
1142  006060' 001401                          BEQ     33$             ;YES END OF TABLE       ;OJM01
1143  006062' 000770                          BR      3$              ;  NO GO DO IT AGAIN    ;OJM01
1144  006064' 004767  000470          33$:    JSR     PC,ERSUB1       ; GET SOME TRIVIAL INFO ;DRB001
1145
1146
1147
1148  006070' 000167  176426                  JMP     ERDR            ;               REV. DO  ;MCP
1149
1150  006074' 032767  000100  171714  BAD:    BIT     #BDSPT,SR1      ;  IF OPERATOR WANTS TYPEOUT THEN       ;DRB001
1151  006102' 001434                          BEQ     12$             ;                       ;DRB001
1152  006104'                         11$:                            ;  BEGIN                ;DRB001
1153  006104' 116767  172450  172540          MOVB    S+6,BADSEC      ;   SET CURRENT SECTOR  ;DRB001
1154  006112' 116767  172443  172534          MOVB    S+7,BADTRK      ;   SET CURRENT TRACK   ;DRB001
1155  006120' 016767  172464  172522          MOV     S+36,BADCYL     ;   SET CURRENT CYLINDER ;DRB001
1156                                          ;*****************************************************************
 (1)                                          ;CONVERT UNITNO TO ASCII AND
```



```
 (1)                                          ;STORE AT ADR1
 (1)  006126' 104420  000000' 000676'         OTOA$,BEGIN,UNITNO,ADR1
 (1)  006134' 011374'
 (1)                                          ;*************************************************************
1157                                          ;*************************************************************
 (1)                                          ;CONVERT BADCYL TO ASCII AND
 (1)                                          ;STORE AT CYLNO
 (1)  006136' 104420  000000' 000650'         OTOA$,BEGIN,BADCYL,CYLNO
 (1)  006144' 007472'
 (1)                                          ;*************************************************************
1158                                          ;*************************************************************
 (1)                                          ;CONVERT BADTRK TO ASCII AND
 (1)                                          ;STORE AT TRKNO
 (1)  006146' 104420  000000' 000654'         OTOA$,BEGIN,BADTRK,TRKNO
 (1)  006154' 007510'
 (1)                                          ;*************************************************************
1159                                          ;*************************************************************
 (1)                                          ;CONVERT BADSEC TO ASCII AND
 (1)                                          ;STORE AT SECNO
 (1)  006156' 104420  000000' 000652'         OTOA$,BEGIN,BADSEC,SECNO
 (1)  006164' 007526'
 (1)                                          ;*************************************************************
1160  006166' 104403  000000' 011122'         MSGN$,BEGIN,BADMES      ;ASCII MESSAGE CALL WITH COMMON HEADER
1161  006174'                         12$:                            ;  END                  ;DRB001
1162  006174' 026727  172436  000552'         CMP     NXTBBK,#ENDBBK  ;  IF MORE SPACE IN TABLE THEN  ;DRB001
1163  006202' 002010                          BGE     13$             ;  BEGIN                ;DRB001
1164  006204' 016700  172426                  MOV     NXTBBK,R0       ;   R0 POINTS TO NEXT FREE SLOT IN TABLE;DRB001
1165  006210' 016720  172434                  MOV     BADCYL,(R0)+    ;   STUFF CYLINDER NUMBER       ;DRB001
1166  006214' 016720  172340                  MOV     S+6,(R0)+       ;   STUFF REST OF DISK ADDRESS  ;DRB001
1167
1168  006220' 010067  172412                  MOV     R0,NXTBBK       ;   UPDATE POINTER              ;DRB001
1169  006224'                         13$:                            ;  END                  ;DRB001
1170  006224' 000207                          RTS     PC
1171
1172
1173
1174
1175
1176  006226' 012767  077777  172456  NOTRDY: MOV     #77777,CLK      ; SET THE TIMER
1177  006234'                         4$:
1178  006234' 032777  004000  173536          BIT     #BIT11,@RHCS1   ;DO I HAVE THE DRIVE DVA?
1179  006242' 001406                          BEQ     6$              ;NO
1180  006244' 010046                          MOV     R0,-(SP)
1181  006246' 012600                          MOV     (SP)+,R0        ;JUST WASTE A LITTLE TIME
1182  006250' 032777  004000  173522          BIT     #BIT11,@RHCS1           ;STILL GOT IT?
1183  006256' 001017                          BNE     2$
1184  006260'                         6$:
 (1)  006260' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
 (1)  006264' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
1185  006270' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
 (1)  006274' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
1186  006300' 005367  172406                  DEC     CLK             ;COUNT # OF TRIES
1187  006304' 001353                          BNE     4$              ;NOT DONE YET
1188  006306' 104403  000000' 011146'         MSGN$,BEGIN,NOT ;ASCII MESSAGE CALL WITH COMMON HEADER
1189  006314' 000426                          BR      7$              ;COULD NOT GET DRIVE
1190  006316' 004567  174350          2$:     JSR     R5,CLEAR        ;SET THE CONTROLLER AND DRIVE
```

```
1191  006322' 004567  000074          JSR     R5,READY        ; IS DRIVE READY ?
1192  006326' 000434                   BR      1$              ; YES, CONTINUE
1193  006330' 004767  000224  5$:     JSR     PC,ERSUB1       ; LOAD ERROR INFORMATION
1194  006334' 005000                   CLR     R0              ;MOVE DRIVE REG INTO TABLE
1195  006336' 016701  173436           MOV     RHCS1,R1
1196  006342' 012160  000552' 22$:     MOV     (R1)+,S(R0)
1197  006346' 005720                   TST     (R0)+
1198  006350' 022700  000046           CMP     #46,R0
1199  006354' 001372                   BNE     22$
1200  006356' 012767  000006  171522   MOV     #6,ERRTYP       ;DRIVE NOT READY
1201                                    ;*****************************************************************
 (1)  006364' 104405  000000' 001722'  HRDERS,BEGIN,TABLE      ; DRIVE  NOT READY
 (1)                                    ;*****************************************************************
1202                                                                    ;**-1                   ;DRB001
1203  006372' 004767  174246  7$:      JSR     PC,RELESE       ;RELEASE THE DRIVE              ;DRB001
1204  006376' 004767  176062           JSR     PC,DROP         ; NO, DROP THE DRIVE
1205  006402' 104403  000000' 011164'  MSGNS,BEGIN,DMP ;ASCII MESSAGE CALL WITH COMMON HEADER
1206  006410' 016706  171416           MOV     SPOINT,R6
1207  006414' 000167  173636           JMP     LOOP2
1208  006420' 000207           1$:     RTS     PC              ;RETURN
1209                                    ;       ------------------------
1210
1211
1212  006422' 016777  172250  173360  READY:  MOV     UNITNO,@RHCS2   ; LOAD UNIT ADDRESS
1213  006430' 017700  173344           MOV     @RHCS1,R0
1214  006434' 032700  004000           BIT     #BIT11,R0
1215  006440' 001433                   BEQ     1$
1216  006442' 017700  173344           MOV     @RPDS,R0        ; SAVE STATUS IN R0
1217  006446' 105700                   TSTB    R0              ; DRIVE READY ?
1218  006450' 100027                   BPL     1$              ; NO
1219  006452' 032700  000100           BIT     #BIT6,R0        ; VOLUME VALID ?
1220  006456' 001424                   BEQ     1$              ; NO
1221  006460' 032700  000400           BIT     #BIT8,R0        ; DRIVE PRESENT ?
1222  006464' 001421                   BEQ     1$              ; NO
1223  006466' 032700  004000           BIT     #BIT11,R0       ; WRITE LOCKED ?
1224  006472' 001016                   BNE     1$              ; YES
1225  006474' 032700  010000           BIT     #BIT12,R0       ; MEDIUM ON LINE ?
1226  006500' 001413                   BEQ     1$              ; NO
1227  006502' 032700  040000           BIT     #BIT14,R0       ; ANY ERRORS ?
1228  006506' 001010                   BNE     1$              ; YES
1229  006510' 005700                   TST     R0              ; ATTENTION SET ?
1230  006512' 100406                   BMI     1$              ; YES
1231  006514' 032777  004000  173256   BIT     #BIT11,@RHCS1   ;DVA SET?
1232  006522' 001402                   BEQ     1$              ;BR IF NOT
1233                                                                    ;**-3                   ;DRB001
1234  006524' 000261                   SEC                     ; READY                         ;DRB001
1235  006526' 000401                   BR      2$              ; RETURN                        ;DRB001
1236  006530' 000241           1$:     CLC                     ; NOT READY                     ;DRB001
1237  006532' 000205           2$:     RTS     R5              ;RETURN                         ;DRB001
1238                                    ;       ------------------------
1239
1240
1241  006534' 014167  171346  ERSUB2: MOV     -(R1),ASB       ; LOAD THE DATA
1242  006540' 010167  171336           MOV     R1,SBADR        ; LOAD ADDRESS OF DATA WRITTEN
1243  006544' 014267  171340           MOV     -(R2),AWAS      ; LOAD THE DATA
1244  006550' 010267  171330           MOV     R2,WASADR       ; LOAD ADDRESS OF DATA READ
```



```
1245  006554' 005721                   TST     (R1)+           ; RESET REG. 1
1246  006556' 005722                   TST     (R2)+           ; RESET REG. 2
1247
1248  006560' 016767  173214  171312  ERSUB1: MOV     RHCS1,CSRA      ; LOAD ADR OF CURRENT CSR
1249  006566' 017767  173206  171306           MOV     @RHCS1,ACSR     ; LOAD CONTENTS OF CURRENT CSR
1250  006574' 000207                   RTS     PC              ; RETURN
1251
1252
1253  006576' 017700  173176  REZET:  MOV     @RHCS1,R0
1254  006602' 032700  004000           BIT     #BIT11,R0
1255  006606' 001005                   BNE     3$
1256  006610' 104407  000000'          BREAKS,BEGIN            ;TEMPORARY RETURN TO MONITOR....
 (1)  006614' 104407  000000'          BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
1257  006620' 000766                   BR      REZET
1258  006622' 012777  000040  173160  3$:     MOV     #BIT5,@RHCS2    ; ISSUE AN RH11 INIT
1259  006630' 012767  077777  172054           MOV     #77777,CLK      ; SET THE TIMER
1260  006636' 105777  173136           1$:     TSTB    @RHCS1          ; CONTROLLER READY ?
1261  006642' 100417                   BMI     2$              ; YES, CONTINUE
1262  006644' 104407  000000'          BREAKS,BEGIN            ;TEMPORARY RETURN TO MONITOR....
 (1)  006650' 104407  000000'          BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
1263  006654' 005367  172032           DEC     CLK             ; WAIT SOME MORE ?
1264  006660' 001366                   BNE     1$              ; YES
1265  006662' 012767  000003  171216   MOV     #3,ERRTYP       ;CONTROLLER NOT READY
1266                                    ;*****************************************************************
 (1)  006670' 104405  000000' 000000   HRDERS,BEGIN,NULL       ; CONTROLLER NOT READY
 (1)                                    ;*****************************************************************
1267  006676' 104410  000000'          ENDS,BEGIN              ;
1268  006702' 017746  173110  2$:      MOV     @RPAS,-(SP)
1269  006706' 012677  173104           MOV     (SP)+,@RPAS
1270  006712' 000207                   RTS     PC              ; RETURN
1271                                    ;       ------------------------
1272
1273
1274  006714' 016700  171066  SETUP:  MOV     ADDR,R0         ; GET DEVICE ADDRESS
1275  006720' 010067  173054           MOV     R0,RHCS1        ; GENERATE REGISTER ADDRESSES
1276  006724' 062700  000002           ADD     #2,R0
1277  006730' 010067  173046           MOV     R0,RHWC
1278  006734' 062700  000002           ADD     #2,R0
1279  006740' 010067  173040           MOV     R0,RHBA
1280  006744' 062700  000002           ADD     #2,R0
1281  006750' 010067  173032           MOV     R0,RPDA
1282  006754' 062700  000002           ADD     #2,R0
1283  006760' 010067  173024           MOV     R0,RHCS2
1284  006764' 062700  000002           ADD     #2,R0
1285  006770' 010067  173016           MOV     R0,RPDS
1286  006774' 062700  000002           ADD     #2,R0
1287  007000' 010067  173010           MOV     R0,RPER1
1288  007004' 062700  000002           ADD     #2,R0
1289  007010' 010067  173002           MOV     R0,RPAS
1290  007014' 062700  000002           ADD     #2,R0
1291  007020' 010067  172774           MOV     R0,RPLA
1292  007024' 062700  000002           ADD     #2,R0
1293  007030' 010067  172766           MOV     R0,RHDB
1294  007034' 062700  000002           ADD     #2,R0
1295  007040' 010067  172760           MOV     R0,RPMR
1296  007044' 062700  000002           ADD     #2,R0
```

```
1297  007050' 010067  172752          MOV     R0,RPDT
1298  007054' 062700  000002          ADD     #2,R0
1299  007060' 010067  172744          MOV     R0,RPSN
1300  007064' 062700  000002          ADD     #2,R0
1301  007070' 010067  172736          MOV     R0,RPOF
1302  007074' 062700  000002          ADD     #2,R0
1303  007100' 010067  172730          MOV     R0,RPDC
1304  007104' 062700  000002          ADD     #2,R0
1305  007110' 010067  172722          MOV     R0,RPCC
1306  007114' 062700  000002          ADD     #2,R0
1307  007120' 010067  172714          MOV     R0,RPER2
1308  007124' 062700  000002          ADD     #2,R0
1309  007130' 010067  172706          MOV     R0,RPER3
1310  007134' 062700  000002          ADD     #2,R0
1311  007140' 010067  172700          MOV     R0,RPEC1
1312  007144' 062700  000002          ADD     #2,R0
1313  007150' 010067  172672          MOV     R0,RPEC2
1314                                                                   ;**-7           ;DRB001
1315  007154' 032767  001000  170674  BIT     #ADDR22,RES1    ;DO WE HAVE 22-BIT ADDR SUPPORT?  ;DRB001
1316  007162' 001415                  BEQ     1$              ;NO                     ;DRB001
1317  007164' 032767  000200  170624  BIT     #RH70,SR1       ;IS THIS AN RH70        ;DRB001
1318  007172' 001411                  BEQ     1$              ;NO                     ;DRB001
1319  007174' 062700  000002          ADD     #2,R0
1320  007200' 010067  172644          MOV     R0,RHBAE
1321  007204' 062700  000002          ADD     #2,R0
1322  007210' 010067  172636          MOV     R0,RHCS3
1323
1324  007214' 000403                  BR      2$                              ;DRB001
1325  007216' 042767  001000  170632  1$:     BIC     #ADDR22,RES1    ; CAN'T USE 22 BIT ADDRESSING  ;DRB001
1326  007224' 016700  170560          2$:     MOV     VECTOR,R0       ; GET VECTOR ADDRESS
1327  007230' 012720  003104'          MOV     #NTRUPT,(R0)+   ; SET POINTER JUST IN CASE
1328  007234' 116710  170552          MOVB    BR1,(R0)        ; SET PRIORITY
1329  007240' 000207                  RTS     PC              ; RETURN
1330
```



```
1332                                    .SBTTL  MODULE MESSAGES                 ;DRB001
1333                                                                    ;**-40  ;DRB001
1334  007242' 042440  051122  051117  MES1:   .ASCIZ  ' ERROR%'                 ;DRB001
      007250' 000045
1335  007252' 050040  051101  052111  MES2:   .ASCIZ  ' PARITY'                 ;DRB001
      007260' 000131
1336  007262' 042040  044522  042526  MES3:   .ASCIZ  ' DRIVE '                 ;DRB001
      007270' 000040
1337  007272' 046440  051501  041123  MES4:   .ASCIZ  ' MASSBUS'                ;DRB001
      007300' 051525     000
1338  007303'    040  040504  040524  MES5:   .ASCIZ  ' DATA'                   ;DRB001
      007310'    000
1339  007311'    040  042522  051124  MES6:   .ASCIZ  ' RETRY EXCEEDED%'        ;DRB001
      007316' 020131  054105  042503
      007324' 042105  042105  000045
1340  007332' 046040  052101  000105  MES7:   .ASCIZ  ' LATE'                   ;DRB001
1341  007340' 047516  020124  042522  MES8:   .ASCIZ  'NOT READY%'             ;DRB001
      007346' 042101  022531     000
1342  007353'    040  047503  046125  MES9:   .ASCIZ  ' COULD NOT GET'          ;DRB001
      007360' 020104  047516  020124
      007366' 042507  000124
1343  007372' 042040  047522  050120  MES10:  .ASCIZ  ' DROPPED'               ;DRB001
      007400' 042105     000
1344  007403'    072  047040  052117  MES11:  .ASCIZ  ': NOT RP04/5/6%'        ;DRB001
      007410' 051040  030120  027464
      007416' 027465  022466     000
1345  007423'    040  051124  047101  MES12:  .ASCIZ  ' TRANSFER'              ;DRB001
      007430' 043123  051105     000
1346  007435'    045     000  MES13:  .ASCIZ  '%'                             ;DRB001
1347  007437'    072  041040  042101  MES15:  .ASCIZ  ': BADSPOT AT (OCTAL) CYL: '  ;DRB001
      007444' 050123  052117  040440
      007452' 020124  047450  052103
      007460' 046101  020051  054503
      007466' 035114  000040
1348  007472' 000002          CYLNO:  .BLKB   2                               ;DRB001
1349  007474' 020040  020040  020054  MES16:  .ASCIZ  '    , TRK: '           ;DRB001
      007502' 051124  035113  000040
1350  007510' 000004          TRKNO:  .BLKB   4                               ;DRB001
1351  007514' 020040  020054  042523  MES17:  .ASCIZ  '  , SEC: '             ;DRB001
      007522' 035103  000040
1352  007526' 000004          SECNO:  .BLKB   4                               ;DRB001
1353  007532' 020040  000045  MES18:  .ASCIZ  '  %'                           ;DRB001
1354  007536' 052440  045516  047516  MES19:  .ASCIZ  ' UNKNOWN'              ;DRB001
      007544' 047127     000
1355
1356
1357
1361                                    ;/////\\\\\/////\\\\\/////\\\\\/////\\\\\/////\\\\\
1362                                    ;\\\\\/////\\\\\/////\\\\\/////\\\\\/////\\\\\/////
1363
1364
1365                                    ;REV D  UPDATES          ;MCP
1366
1367  007547'    110  051101  020104  MES20:  .ASCIZ  'HARD ERROR(S) SET IN RPER2  %'
      007554' 051105  047522  024122
      007562' 024523  051440  052105
```

```
                007570' 044440  020116  050122
                007576' 051105  020062  022440
                007604'    000
 1368           007605'    110  051101  020104  MES21:  .ASCIZ  'HARD ERROR(S) SET IN RPER3  % '
                007612' 051105  047522  024122
                007620' 024523  051440  052105
                007626' 044440  020116  050122
                007634' 051105  020063  022440
                007642' 000040
 1369           007644' 051127  052111  020105  MES22:  .ASCIZ  'WRITE CHECK ERROR  % '
                007652' 044103  041505  020113
                007660' 051105  047522  020122
                007666' 022440  000040
 1370           007672' 041523  051440  052105  MES23:  .ASCIZ  'SC SET  %'
                007700' 020040  000045
 1371           007704' 044124  020105  047506  MES24:  .ASCIZ  'THE FOLLOWING ERROR BITS ARE SET IN RPCS2 % % '
                007712' 046114  053517  047111
                007720' 020107  051105  047522
                007726' 020122  044502  051524
                007734' 040440  042522  051440
                007742' 052105  044440  020116
                007750' 050122  051503  020062
                007756' 020045  020045     000
 1372           007763'    125  044516  052502  MES25:  .ASCIZ  'UNIBUS PARITY ERROR % '
                007770' 020123  040520  044522
                007776' 054524  042440  051122
                010004' 051117  022440  000040
 1373           010012' 047516  020116  054105  MES26:  .ASCIZ  'NON EXISTANT DRIVE % '
                010020' 051511  040524  052116
                010026' 042040  044522  042526
                010034' 022440  000040
 1374           010040' 054116  020115  022440  MES27:  .ASCIZ  'NXM  %'
                010046'    000
 1375           010047'    120  047522  051107  MES28:  .ASCIZ  'PROGRAM ERROR  PGE  % '
                010054' 046501  042440  051122
                010062' 051117  020040  043520
                010070' 020105  022440  000040
 1376           010076' 044515  051523  042105  MES29:  .ASCIZ  'MISSED TRANSFER  %  '
                010104' 052040  040522  051516
                010112' 042506  020122  022440
                010120' 020040     000
 1377           010123'    115  051501  041123  MES30:  .ASCIZ  'MASSBUS DATA LINE PARITY ERROR  % '
                010130' 051525  042040  052101
                010136' 020101  044514  042516
                010144' 050040  051101  052111
                010152' 020131  051105  047522
                010160' 020122  022440  000040
 1378           010166' 044124  020105  047506  MES31:  .ASCIZ  'THE FOLLOWING BITS ARE SET IN RPER1  % % '
                010174' 046114  053517  047111
                010202' 020107  044502  051524
                010210' 040440  042522  051440
                010216' 052105  044440  020116
                010224' 050122  051105  020061
                010232' 022440  022440  000040
 1379           010240' 040504  040524  041440  MES32:  .ASCIZ  'DATA CHECK ERROR  % '
                010246' 042510  045503  042440
```



```
                010254' 051122  051117  020040
                010262' 020045     000
 1380           010265'    106  051117  040515  MES33:  .ASCIZ  'FORMAT ERROR  %  '
                010272' 020124  051105  047522
                010300' 020122  022440  020040
                010306'    000
 1381           010307'     40  042510  042101  MES34:  .ASCIZ  ' HEADER CRC ERROR % '
                010314' 051105  041440  041522
                010322' 042440  051122  051117
                010330' 022440  000040
 1382           010334' 042510  042101  051105  MES35:  .ASCIZ  'HEADER COMPARE ERROR  % '
                010342' 041440  046517  040520
                010350' 042522  042440  051122
                010356' 051117  020040  020045
                010364'    000
 1383           010365'    104  044522  042526  MES36:  .ASCIZ  'DRIVE UNSAFE  % '
                010372' 052440  051516  043101
                010400' 020105  022440  000040
 1384           010406' 050117  051105  052101  MES37:  .ASCIZ  'OPERATION INCOMPLETE % '
                010414' 047511  020116  047111
                010422' 047503  050115  042514
                010430' 042524  022440  000040
 1385           010436' 051104  053111  020105  MES38:  .ASCIZ  'DRIVE TIMING ERROR % '
                010444' 044524  044515  043516
                010452' 042440  051122  051117
                010460' 022440  000040
 1386           010464' 051127  052111  020105  MES39:  .ASCIZ  'WRITE LOCK ERROR % '
                010472' 047514  045503  042440
                010500' 051122  051117  022440
                010506' 000040
 1387           010510' 047111  040526  044514  MES40:  .ASCIZ  'INVALID ADDRESS ERROR % '
                010516' 020104  042101  051104
                010524' 051505  020123  051105
                010532' 047522  020122  020045
                010540'    000
 1388           010541'    105  041503  044040  MES41:  .ASCIZ  'ECC HARD ERROR % '
                010546' 051101  020104  051105
                010554' 047522  020122  020045
                010562'    000
 1389           010563'    120  051101  052111  MES42:  .ASCIZ  'PARITY ERROR % '
                010570' 020131  051105  047522
                010576' 020122  020045     000
 1390           010603'    122  043505  051511  MES43:  .ASCIZ  'REGISTER MODIFICATION REFUSED  % '
                010610' 042524  020122  047515
                010616' 044504  044506  040503
                010624' 044524  047117  051040
                010632' 043105  051525  042105
                010640' 020040  020045     000
 1391           010645'    111  046114  043505  MES44:  .ASCIZ  'ILLEGAL REGISTER % '
                010652' 046101  051040  043505
                010660' 051511  042524  020122
                010666' 020045     000
 1392           010671'    111  046114  043505  MES45:  .ASCIZ  'ILLEGAL FUNCTION % '
                010676' 046101  043040  047125
                010704' 052103  047511  020116
                010712' 020045     000
```

```
1393  010715'  040  022445  020045  MES46:  .ASCIZ  ' %%%  ********************************** %% '
      010722' 020040  025052  025052
      010730' 025052  025052  025052
      010736' 025052  025052  025052
      010744' 025052  025052  025052
      010752' 025052  025052  025052
      010760' 025052  025052  025052
      010766' 025052  022440  020045
      010774'  000
1394  010775'  101  042104  042522  MES47:  .ASCIZ  'ADDRESS OVERFLOW ERROR % '
      011002' 051523  047440  042526
      011010' 043122  047514  020127
      011016' 051105  047522  020122
      011024' 020045  000
1395  011027'  127  044522  042524  MES48:  .ASCIZ  'WRITE LOCK FAILURE % '
      011034' 046040  041517  020113
      011042' 040506  046111  051125
      011050' 020105  020045  000
1396
1397
1398                                  ; R E V  D   END OF  UPDATE
1399
1400                                  ;/////\\\\\/////\\\\\/////\\\\\/////\\\\\
1401                                  ;\\\\\/////\\\\\/////\\\\\/////\\\\\/////
1402
1403
1407          011056'                 .EVEN                                                ;DRB001
1408                                                                                       ;DRB001
1409  011056' 007423'                 TRERR:  MES12                                        ;DRB001
1410  011060' 007242'                         MES1                                         ;DRB001
1411  011062' 177777                          177777                                       ;DRB001
1412                                                                                       ;DRB001
1413  011064' 007272'                 MCPERR: MES4                                         ;DRB001
1414  011066' 007252'                         MES2                                         ;DRB001
1415  011070' 007242'                         MES1                                         ;DRB001
1416  011072' 177777                          177777                                       ;DRB001
1417                                                                                       ;DRB001
1418  011074' 007272'                 MDPERR: MES4                                         ;DRB001
1419  011076' 007303'                         MES5                                         ;DRB001
1420  011100' 007252'                         MES2                                         ;DRB001
1421  011102' 007242'                         MES1                                         ;DRB001
1422  011104' 177777                          177777                                       ;DRB001
1423                                                                                       ;DRB001
1424  011106' 007311'                 EXCED:  MES6                                         ;DRB001
1425  011110' 177777                          177777                                       ;DRB001
1426                                                                                       ;DRB001
1427  011112' 007303'                 DLTERR: MES5                                         ;DRB001
1428  011114' 007332'                         MES7                                         ;DRB001
1429  011116' 007242'                         MES1                                         ;DRB001
1430  011120' 177777                          177777                                       ;DRB001
1431                                                                                       ;DRB001
1432  011122' 007262'                 BADMES: MES3                                         ;DRB001
1433  011124' 011401'                         NUMB                                         ;DRB001
1434  011126' 007437'                         MES15                                        ;DRB001
1435  011130' 007474'                         MES16                                        ;DRB001
1436  011132' 007514'                         MES17                                        ;DRB001
```



```
1437  011134' 007532'                         MES18                                        ;DRB001
1438  011136' 177777                          177777                                       ;DRB001
1439                                                                                       ;DRB001
1440  011140' 007536'                 UNKNWN: MES19                                        ;DRB001
1441  011142' 007242'                         MES1                                         ;DRB001
1442  011144' 177777                          177777                                       ;DRB001
1443                                                                                       ;DRB001
1444  011146' 007262'                 NOT:    MES3                                         ;DRB001
1445  011150' 007340'                         MES8                                         ;DRB001
1446  011152' 177777                          177777                                       ;DRB001
1447                                                                                       ;DRB001
1448  011154' 007353'                 TOUT:   MES9                                         ;DRB001
1449  011156' 007262'                         MES3                                         ;DRB001
1450  011160' 007435'                         MES13                                        ;DRB001
1451  011162' 177777                          177777                                       ;DRB001
1452                                                                                       ;DRB001
1453  011164' 007262'                 DRP:    MES3                                         ;DRB001
1454  011166' 011401'                         NUMB                                         ;DRB001
1455  011170' 007372'                         MES10                                        ;DRB001
1456  011172' 007435'                         MES13                                        ;DRB001
1457  011174' 177777                          177777                                       ;DRB001
1458                                                                                       ;DRB001
1459  011176' 007262'                 BADTYP: MES3                                         ;DRB001
1460  011200' 011401'                         NUMB                                         ;DRB001
1461  011202' 007372'                         MES10                                        ;DRB001
1462  011204' 007403'                         MES11                                        ;DRB001
1463  011206' 177777                          177777                                       ;DRB001
1464
1465
1469                                  ;/////\\\\\/////\\\\\/////\\\\\/////\\\\\
1470                                  ;\\\\\/////\\\\\/////\\\\\/////\\\\\/////
1471
1472                                  ;R E V  D  UPDATES
1473
1474  011210' 007547'                 ER2ERR: MES20
1475  011212' 177777                          177777
1476
1477  011214' 007605'                 ER3ERR: MES21
1478  011216' 177777                          177777
1479
1480  011220' 007644'                 WCEERR: MES22
1481  011222' 177777                          177777
1482
1483  011224' 007672'                 SCERR:  MES23
1484  011226' 177777                          177777
1485
1486  011230' 007704'                 CS2HD:  MES24
1487  011232' 177777                          177777
1488
1489  011234' 007763'                 UPER:   MES25
1490  011236' 177777                          177777
1491
1492  011240' 010012'                 NEDER:  MES26
1493  011242' 177777                          177777
1494
1495  011244' 010040'                 NXM:    MES27
```

```
1496  011246' 177777                          177777
1497
1498  011250' 010047'                 PGERR:  MES28
1499  011252' 177777                          177777
1500
1501  011254' 010076'                 MXFER:  MES29
1502  011256' 177777                          177777
1503
1504  011260' 010123'                 MDPER:  MES30
1505  011262' 177777                          177777
1506
1507  011264' 010166'                 ER1HD:  MES31
1508  011266' 177777                          177777
1509
1510  011270' 010240'                 DCKER:  MES32
1511  011272' 177777                          177777
1512
1513  011274' 010265'                 FERER:  MES33
1514  011276' 177777                          177777
1515
1516  011300' 010307'                 HCRC:   MES34
1517  011302' 177777                          177777
1518
1519  011304' 010334'                 HCE:    MES35
1520  011306' 177777                          177777
1521
1522  011310' 010365'                 UNSER:  MES36
1523  011312' 177777                          177777
1524
1525  011314' 010406'                 OPIER:  MES37
1526  011316' 177777                          177777
1527
1528  011320' 010436'                 DTER:   MES38
1529  011322' 177777                          177777
1530  011324' 010464'                 WLER:   MES39
1531  011326' 177777                          177777
1532  011330' 010510'                 IAER:   MES40
1533  011332' 177777                          177777
1534
1535  011334' 010541'                 ECHR:   MES41
1536  011336' 177777                          177777
1537
1538  011340' 010563'                 PARR:   MES42
1539  011342' 177777                          177777
1540
1541  011344' 010603'                 RMRR:   MES43
1542  011346' 177777                          177777
1543
1544  011350' 010645'                 ILRR:   MES44
1545  011352' 177777                          177777
1546
1547  011354' 010671'                 ILFR:   MES45
1548  011356' 177777                          177777
1549  011360' 010715'                 ASTER:  MES46
1550  011362' 177777                          177777
1551
```



```
1552  011364' 010775'                 AOE:    MES47                   ;       REV. D0
1553  011366' 177777                          177777
1554
1555  011370' 011027'                 WCF:    MES48                   ;       REV. D0
1556  011372' 177777                          177777
1557
1558
1559
1560                                          ;R E V  D  END UPDATE
1561
1562                                          ;////\\\\////\\\\////\\\\////\\\\////\\\\
1563                                          ;////\\\\////\\\\////\\\\////\\\\////\\\\
1564
1565                                                                                  ;DRB001
1566  011374' 000005                  ADR1:   .BLKB   5                             ;DRB001
1567  011401'     040     000         NUMB:   .ASCIZ  / /                          ;DRB001
1568          011404'                         .EVEN                                 ;DRB001
1569                                                                                ;DRB001
1570                                  ;-----------------------------                ;DRB001
1571                                                                                ;DRB001
1572          000001                          .END                                  ;DRB001
```

```
ACSR    000102R          370#  1249#
ADDR    000006R          370#  1274
ADDR22= 001000           370#   615   1315   1325
ADR1    011374R           934  1156   1566#
AOE     011364R          1094  1552#
ASB     000106R          370#  1241#
ASTAT   000104R          370#
ASTER   011360R           954  1016   1549#
AWAS    000110R          370#  1243#
BAD     006074R           652  1101   1150#
BADCYL  000650R          410#  1155#  1157   1165
BADMES  011122R          1160  1432#
BADSEC  000652R          411#  1153#  1159
BADSPT  000252R          373#   401    489    782
BADTRK  000654R          412#  1154#  1158
BADTYP  011176R           901  1459#
BDSPT = 000100           386#  1150
BEGIN   000000R          370#   520    523    535    536    545    596    622    630    638    651    705    901
                          934   954    963    967    979    983    987    991    995   1007   1009   1016   1023
                         1026  1029   1032   1035   1038   1042   1045   1049   1052   1063   1066   1079   1082
                         1085  1088   1091   1094   1097   1100   1104   1107   1110   1113   1156   1157   1158
                         1159  1160   1184   1185   1188   1201   1205   1256   1262   1266   1267
BITTAB  003310R          633    671   683#
BIT0  = 000001           370#   758    867    877    884    894   1111
BIT1  = 000002           370#  1108
BIT10 = 002000           370#  1033   1089
BIT11 = 004000           370#  1030   1086   1178   1182   1214   1223   1231   1254
BIT12 = 010000           370#   611   1027   1083   1225
BIT13 = 020000           370#   964   1024   1080
BIT14 = 040000           370#   610    971    984    988   1077   1227
BIT15 = 100000           370#   956    992   1046   1055
BIT2  = 000004           370#   383   1105
BIT3  = 000010           370#  1102
BIT4  = 000020           370#   384   1050   1055
BIT5  = 000040           370#   385   1095   1258
BIT6  = 000100           370#   386   1098   1219
BIT7  = 000200           370#   387   1064
BIT8  = 000400           370#  1040   1060   1221
BIT9  = 001000           370#  1036   1092
BPORT = 000020           384#   756
BREAK$= 104407           370#   596   1184   1185   1256   1262
BR1     000012R          370#  1328
BR2     000013R          370#
BTOD$ = 104421           370#
CDATA$= 104412           370#   545
CDERCT  000144R          370#
CDWDCT  000146R          370#
CHKCS2  005156R           970  1022#
CHKER1  005320R           975  1045#
CLEAR   002672R           600#  1190
CLK     000712R          430#  1176#  1186#  1259#  1263#
CLRCOM  004454R           922#   924
CLRRB   004444R           920#
CONFIG  000056R          370#
CSRA    000100R          370#  1248#
CS2HD   011230R          1023  1486#
```



```
CYCLE   002322R           530   539#
CYL     000644R          405#  510#   708#   735#   741   745#   749   752#   755   767#
CYLLIM  000640R          403#   741    750    760   868#  878#  885#  895#
CYLNDR  000642R          404#   571    581    591   755#  760#   762   764#   786   802
CYLNO   007472R          1157  1348#
DATCKS= 104411           370#
DATERS= 104404           370#
DCKER   011270R          1049  1510#
DLATE = 000004           383#   960
DLTCNT  000700R          425#   958#
DLTERR  011112R           963  1427#
DLTFLG  005154R           953#   959#   986#   990#   994#  1002   1004  1019#  1048#  1058#  1062#
DRIVE   000674R          422#
DROP    004464R           507    900    927#  1204
DRP     011164R          1205  1453#
DSKADR  000646R          406#   511#   570    580    590   746#   768#
DTER    011320R          1085  1528#
DVICE   000672R          421#   494#   504    533    671   933#
DVID1   000014R          370#   494
EA22    000634R          399#   624    625#   626#   627
ECHR    011334R          1100  1535#
ENDBBK  000552R          381#  1162
ENDITS= 104413           370#   536
END$  = 104410           370#   535   1267
EROR    004522R           953#  1148
ERRORS  006022R           640   1127#
ERRTYP  000106R          370#  1200#  1265#
ERSUB1  006560R          1144   1193   1248#
ERSUB2  006534R          1241#
ER1HD   011264R          1045  1507#
ER2ERR  011210R           979  1474#
ER3ERR  011214R           983  1477#
EXCED   011106R           651  1424#
EXITS = 104400           370#   630
FERADR  000710R          429#   565#   575#   585#   650
FERER   011274R          1052  1513#
FLAG    000670R          420#   758    867#   877#   884#   894#
FREE    000150R          370#
FUNC    000702R          426#   564#   574#   584#   628#   629
GETDVT  004202R           679    859#
GETPA$= 104415           370#   520
GO      002756R           573    583    593    614#
GWBUF$= 104414           370#   523
HCE     011304R          1066  1519#
HCRC    011300R          1063  1516#
HRDCNT  000044R          370#
HRDCS2= 137400           388#   968
HRDERR  000720R          433#   645    955#   962#   966#   978#   982#  1000  1022#  1057#  1071#  1076#
HRDER$= 104405           370#  1009   1201   1266
HRDER1= 177777           389#   973
HRDER2= 167777           390#   976
HRDER3= 140153           391#   980
HRDPAS  000050R          370#
IAER    011330R          1091  1532#
ICUNT   000036R          370#
ICOUNT  000040R          370#
```

```
IDNUM   000122R          370#
ILFR    011354R         1113    1547#
ILRR    011350R         1110    1544#
IMODX.= 000000           370#    523
INIT    000030R          370#
INTR    000120R          370#    488#
LASCYL  000656R          414#    762     764     854#    869#    879#    886#    896#
LASSEC  000662R          416#    773     775     857#
LASTRK  000660R          415#    770     772     856#
LOOP1   002244R          522#    537
LOOP2   002256R          525#    531     657     1207
MAP22S= 104416           370#    622
MCPERR  011064R          967    1413#
MDPER   011260R         1042    1504#
MDPERR  011074R         1418#
MES1    007242R         1334#    1410    1415    1421    1429    1441
MES10   007372R         1343#    1455    1461
MES11   007403R         1344#    1462
MES12   007423R         1345#    1409
MES13   007435R         1346#    1450    1456
MES15   007437R         1347#    1434
MES16   007474R         1349#    1435
MES17   007514R         1351#    1436
MES18   007532R         1353#    1437
MES19   007536R         1354#    1440
MES2    007252R         1335#    1414    1420
MES20   007547R         1367#    1474
MES21   007605R         1368#    1477
MES22   007644R         1369#    1480
MES23   007672R         1370#    1483
MES24   007704R         1371#    1486
MES25   007763R         1372#    1489
MES26   010012R         1373#    1492
MES27   010040R         1374#    1495
MES28   010047R         1375#    1498
MES29   010076R         1376#    1501
MES3    007262R         1336#    1432    1444    1449    1453    1459
MES30   010123R         1377#    1504
MES31   010166R         1378#    1507
MES32   010240R         1379#    1510
MES33   010265R         1380#    1513
MES34   010307R         1381#    1516
MES35   010334R         1382#    1519
MES36   010365R         1383#    1522
MES37   010406R         1384#    1525
MES38   010436R         1385#    1528
MES39   010464R         1386#    1530
MES4    007272R         1337#    1413    1418
MES40   010510R         1387#    1532
MES41   010541R         1388#    1535
MES42   010563R         1389#    1538
MES43   010603R         1390#    1541
MES44   010645R         1391#    1544
MES45   010671R         1392#    1547
MES46   010715R         1393#    1549
MES47   010775R         1394#    1552
```



```
MES48   011027R         1395#    1555
MES5    007303R         1338#    1419    1427
MES6    007311R         1339#    1424
MES7    007332R         1340#    1428
MES8    007340R         1341#    1445
MES9    007353R         1342#    1448
MODLUS  004432R          729      738     751     909#
MODNAM  000000R          370#
MODSP   000252R          370#
MSGN$ = 104403           370#    651     901     954     963     967     979     983     987     991     995    1016    1023
                        1026    1029    1032    1035    1038    1042    1045    1049    1052    1063    1066    1079    1082
                        1085    1088    1091    1094    1097    1100    1104    1107    1110    1113    1160    1188    1205
MSGS$ = 104402           370#
MSG$  = 104401           370#
MXFER   011254R         1038    1501#
NEDER   011240R         1029    1492#
NORAND= 000040           385#     698     743     765
NOT     011146R         1188    1444#
NOTRDY  006226R          678    1013    1176#
NTRUPT  003104R          631#    1327
NULL  = 000000           370#    1266
NUMB    011401R         1433    1454    1460    1567#
NXM     011244R         1032    1495#
NXTBBK  000636R          401#     493*    783    1162    1164    1168*
OPEN  = 000000           370#
OPIER   011314R         1082    1525#
OTOA$ = 104420           370#     934    1156    1157    1158    1159
PARR    011340R         1104    1538#
PASCNT  000034R          370#
PA18    000626R          396#     617#    622
PA22    000632R          398#     623
PGERR   011250R         1035    1498#
PICKBK  003320R          541      655     689#    814
PICKDR  003226R          526      664#
PIRQ$ = 000004           370#     638
POPSP = 005726           370#
POPSP2= 022626           370#
PRTY  = 000000           370#
PRTY0 = 000000           370#
PRTY1 = 000040           370#
PRTY2 = 000100           370#
PRTY3 = 000140           370#
PRTY4 = 000200           370#
PRTY5 = 000240           370#
PRTY6 = 000300           370#
PRTY7 = 000340           370#
P$    = 177776           370#
P$#   = 177776           370#
PUSH  = 005746           370#
PUSH2 = 024646           370#
RAND$ = 104417           370#    705
RANDUM  000054R          370#    706     709     713
RBUF    000722R          370     434#    920
RBUFEA  000140R          370#    542
RBUFPA  000126R          370#    545     589
RBUFSZ  000132R          370#    586     921
```

```
RBUFVA  000124R          370#   520
READ    002550R          544    584#   585
READY   006422R          676   1191   1212#
RELESE  002644R          546    594#  1203
RESTRT  002236R          370    516    518#
RES1    000056R          370#   615   1315   1325*
RES2    000060R          370#
REZET   006576R          514   1253#  1257
RHBA    002004R          463#   569*   579*   589*   617    623*   1279*
RHBAE   002050R          481#   624*  1320*
RHCS1   002000R          461#   595*   601*   604*   605    610*   629*   964   1128   1136   1178   1182   1195
                        1213   1231   1248   1249   1253   1260   1275*
RHCS2   002010R          465#   594*   600*   614*   864*  1212*  1258*  1283*
RHCS3   002052R          482#  1322*
RHDB    002022R          470#   998    999*  1010   1012*  1293*
RHWC    002002R          462#   568*   578*   588*  1277*
RH70  = 000200           387#   515   1317
RMRR    011344R         1107  1541#
RPAS    002016R          468#   608    609*   634    636*  1268   1269*  1289*
RPCC    002036R          476#  1305*
RPDA    002006R          464#   570*   580*   590*  1281*
RPDC    002034R          475#   571*   581*   591*  1303*
RPDS    002012R          466#  1216   1285*
RPDT    002026R          472#   865    872    874    882    889    891   1297*
RPEC1   002044R          479#  1311*
RPEC2   002046R          480#  1313*
RPEH1   002014R          467#  1287*
RPER2   002040R          477#  1307*
RPER3   002042R          478#  1309*
RPLA    002020R          469#  1291*
RPMR    002024R          471#  1295*
RPOF    002032R          474#   611*  1301*
RPSN    002030R          473#  1299*
RSTRT   000112R          370#
S       000552R          393#   437    438    439    440    441    442    443    444    445    446    447    448
                         449    450    451    452    453    454    455    456    457    458    517*   956    968
                         971    973    976    980    984    988    992    996   1024   1027   1030   1033   1036
                        1040   1046   1050   1053   1060   1064   1067   1072   1077   1080   1083   1086   1089
                        1092   1095   1098   1102   1105   1108   1111   1139*  1141   1153   1154   1155   1166
                        1196*
SBADR   000102R          370#  1242*
SCERR   011224R          995   1483#
SECNO   007526R         1159  1352#
SECTOR  000646R          407#   700*   712*   724    727    730*   773    775*   784    810
SETUP   006714R          513   1274#
SIZSEC  000666R          419#   788    841#
SIZTRK  000664R          418#   794    835*   838*   850
SOFCNT  000042R          370#
SOFER$= 104406           370#  1007
SOFPAS  000046R          370#
SPOINT  000032R          370#  1206
SPSIZ = 000040             7#   370
SR1     000016R          370#   515    698    743    756    765    960   1150   1317
SR2     000020R          370#
SR3     000022R          370#
SR4     000024R          370#
```



```
START   002054R          370    486#
STAT    000026R          370#
STORE   000716R          432#   998*  1012   1053*  1054*  1055   1067*  1068*  1069
SVR0    000062R          370#
SVR1    000064R          370#
SVR2    000066R          370#
SVR3    000070R          370#
SVR4    000072R          370#
SVR5    000074R          370#
SVR6    000076R          370#
SYSCNT  000052R          370#
TABLE   001722R          436#  1007   1009   1201
TIMER   000704R          427#
TOUT    011154R         1448#
TRACK   000647R          408#   722*   726*   733    736    739*   770    772*   785    806
TRERR   011056R          991   1409#
TRKNO   007510R         1158  1350#
TRPDFD= 000022           370#
TRY     000714R          431#   529*   647*   648
UNITNO  000676R          423#   506*   522*   594    600    614    632    656*   665*   666    670    864    928
                         934   1156   1212
UNKNWN  011140R         1440#
UNSER   011310R         1079  1522#
UPER    011234R         1026  1489#
VECTOR  000010R          370#  1326
WASADR  000104R          370#  1244#
WBUFEA  000136R          370#   572    582
WBUFPA  000134R          370#   569    579
WBUFRQ  000140R          370#
WBUFSZ  000142R          370#   566    576    832
WCEERR  011220R          987  1480#
WCF     011370R         1097  1555#
WDFR    000116R          370#   487#
WDTO    000114R          370#   486#
WLER    011324R         1088  1530#
WRITCK  002454R          543    574#   575
WRITE   002360R          542    564#   565
WRTLIM  004062R          528    822#
XFLAG   000005R          370#
XMEM    000630R          397#   572*   582*   592*   618*   619*   620*   621*   627*   628
ZERO    000706R          428#
.     = 011404R          434#   545   1348#  1350#  1352#  1407#  1566#  1568#
```

```
BKMOD      95#
BREAK     185#    596    1184    1185    1256    1262
BTOD      204#
CKDATA    240#    545
DATACK    249#
DATERR    138#
DFSEVN    272#    370
DSEVNT    262#    370
END       175#    535    1267
ENDIT     166#    536
ENDMOD    171#
EQUATS    288#    370
EXIT      120#    630
GETPA     231#    520
GWBUFF    219#    523
HRDER     128#   1009    1201    1266
IOMOD      91#
IOMODP    115#
IOMODR    111#
IOMODX    107#    370
LINEUP    551#    572     582     592
MAP22     235#    622
MODULE      8#    370
MSG       154#
MSGN      158#    901     954     963     967     979     983     987     991     995    1016    1023    1026    1029    1032
         1035    1038    1042    1045    1049    1052    1063    1066    1079    1082    1085    1088    1091    1094    1097
         1100    1104    1107    1110    1113    1160    1188    1205
MSGS      162#
NBKMOD    103#
OTOA      190#    934    1156    1157    1158    1159
PIRQ      179#    638
RAND      124#    705
SBKMOD     99#
SOFER     144#   1007


. ABS.  000000      000
        011404      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XRPDD0.OBJ,XRPDD0.LST/CRF=DDXCOM.P11,XRPDD0.P11
 RUN-TIME: 6 12 1 SECONDS
 RUN-TIME RATIO: 45/22=2.0
 CORE USED:  8K  (15 PAGES)
```

```
1                               .NLIST TTM
          .NLIST SEQ,LD,BIN
          .REPT 0
```

IDENTIFICATION

PRODUCT CODE: AC-F098C-MC

PRODUCT NAME: CXRXBC0 RX02 MODULE

PRODUCT DATE: APRIL 1979

MAINTAINER: TAPE DIAGNOSTICS

1. ABSTRACT

RXB IS AN IOMODX THAT EXERCISES TWO RX02 FLOPPY DISKS ON THE UNIBUS. IT EXERCISES BOTH DRIVES BY WRITING AND READING ALL AVAILABLE DRIVES.

ERRORS ARE CHECKED FOR BUFFER FILL, WRITE, READ, AND DATA COMPARE. TWO RETRIES ARE DONE FOR EACH WRITE OR READ STATUS ERROR. ALL ERRORS ARE REPORTED ON THE CONSOLE TTY.

2. REQUIREMENTS

HARDWARE: 1 OR 2 DISKETTES WITH AN RX02 CONTROLLER

STORAGE:: RXB REQUIRES:
1. DECIMAL WORDS: 1661
2. OCTAL WORDS: 03175
3. OCTAL BYTES: 6372

3. PASS IDENTIFICATION

ONE PASS OF THE RXB MODULE CONSISTS OF 130. WRITE AND READ PASSES ON AVAILABLE DRIVES. THE TEST SEQUENCE WRITES THEN READS EVERY THIRD SECTOR OF EVERY TENTH TRACK STARTING AT TRACK 1 SECTOR 1.

THE ENTIRE DISKETTE IS DONE, EACH PASS OF THE DISKETTE STARTS AS SHOWN:

1. STARTS AT SECTOR #1/TRACK #1
   END OF PASS
2. STARTS AT SECTOR #2/TRACK #1
   END OF PASS
3. STARTS AT SECTOR #3/TRACK #1
   END OF PASS
4. STARTS AT SECTOR #1/TRACK #2
   ...ETC.

4. EXECUTION TIME

ONE PASS OF RXB RUNNING ALONE ON THE PDP-11/05 AVERAGES .75 MINUTES FOR 2 DRIVES.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS: DEVADR:177170
VECTOR:264
BR1:5
DEVCNT:2

REQUIRED PARAMETERS: NONE

6. DEVICE/OPTION SETUP

ASSURE ALL DRIVES ARE POWERED UP, DISKETTES INSTALLED,
AND READY.

7. MODULE OPERATION

A. SETUP DRIVE REGISTER ADDRESSES AND MODULE VARIABLES

B. SELECT DRIVES FOR TEST - IF NONE AVAILABLE, DROP MODULE

C. INITIALIZE DRIVES

D. SET AVAILABLE DRIVES TO SINGLE DENSITY

E. WRITE AND READ ALL AVAILABLE DRIVES - IF ERROR, REPORT AND
RETRY FAILING FUNCTION UP TO RETRY LIMIT.

F. DO DATA COMPARE FOR ALL READS - IF ERROR, REPORT

G. UPDATE TRACK + SECTOR - DO E. + F. UNTIL ALL SECTORS
OF AVAILABLE DRIVES ARE DONE.

H. CHANGE DENSITY OF AVAILABLE DRIVES

I. DO E., F. & G. FOR NEW DENSITY

J. UPDATE STARTING ADDRESS, GO TO D

8. OPERATION OPTIONS

SR1 BIT 0 SET(1):
IF RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, REPORT
HARD ERROR AND DROP THE MODULE.

SR1 BIT 0 CLEAR(0):
IF RETRY LIMIT IS EXCEEDED, CONTINUE WITH NEXT TEST.

9. NON-STANDARD PRINTOUTS

A. ALL PRINTOUTS HAVE THE STANDARD FORMAT DESCRIBED IN THE DEC/X11 DOCUMENT.

B. ERROR MESSAGES DUMP THE CONTENTS OF THE RX02 REGISTERS IN THE FOLLOWING ORDER.

```
RXCS  (COMMAND REGISTER)
RXES  (ERROR REGISTER)
RXTA  (TRACK ADDRESS)
RXSA  (SECTOR ADDRESS)
RXSB  (WORD COUNT REG.)            / (DEFINITIVE ERROR CODE)->----
RXSB1 (DRV1 TRACK ADR)             / (DRV0 TRACK ADR)             ¦
RXSB2 (TARGET SECTOR)              / (TARGET TRACK)               ¦
RXSB3 (TRACK ADR SELECTED DRV*)    / (MICROCODE STATUS)->--       ¦
                                                          ¦       ¦
      (UPPER BYTE-ODD)             / (LOWER BYTE-EVEN)    ¦       ¦
                                                          ¦       ¦
    * = ONLY MEANINGFUL ON A CODE 150 ERROR, SEEK ERROR.  ¦       ¦
                                                          ¦       ¦
                                                          ¦       ¦
MICROCODE STATUS <----------------------------------------       ¦
---------------                                                  ¦
                                                                 ¦
BIT #7 = UNIT SELECT                                             ¦
BIT #6 = DENSITY DRIVE #1                                        ¦
BIT #5 = HEAD LOAD                                               ¦
BIT #4 = DENSITY DRIVE #0                                        ¦
                                                                 ¦
                                                                 ¦
    TABLE OF DEFINITIVE ERROR CODES <----------------------------
    -------------------------------

KNXDV0=  10     /DRIVE 0 FAILED TO SEE HOME ON INITIALIZE.  NO ERROR BIT
KNXDV1=  20     /DRIVE 1 FAILED TO SEE HOME ON INITIALIZE.  NO ERROR BIT
KERTRK=  40     /TRIED TO ACCESS A TRACK GREATER THAN 76.
KHOMERR= 50     /HOME WAS FOUND BEFORE DESIRED TRACK WAS REACHED.
KSELFER= 60     /SELF DIAGNOSTIC ERROR.
KNXHDR=  70     /DESIRED SECTOR COULD NOT BE FOUND AFTER LOOKING AT 52 HEADERS.
KWPROT= 100     /WRITE FUNCTION ATTEMPTED ON A WRITE PROTECTED DISK.
KTIMERR=110     /MORE THAN 40 MICROSECONDS AND NO SEPCLOCK SEEN.
KNXPRAM=120     /A PREAMBLE COULD NOT BE FOUND.
KNXIDAM=130     /PREAMBLE FOUND BUT NO ID MARK FOUND WITHIN ALLOWABLE TIME.
KHCHCER=140     /CRC ERROR ON WHAT APPEARED TO BE A HEADER.  ERROR IS NOT ASSERTED.
KTKSKER=150     /TRACK ADDRESS OF GOOD HEADER DOES NOT COMPARE WITH DESIRED TRACK.
KXSTRYS=160     /TOO MANY TRIES FOR AN IDAM.
KNODAM= 170     /DATA AM NOT FOUND IN ALLOTTED TIME.
KDCRCER=200     /CRC ERROR ON READING THE SECTOR FROM THE DISK.
KMANER= 220     /R/W ELECTRONICS FAILED MAINTENANCE MODE TEST.
KWCNOV= 230     /WORD COUNT OVERFLOW.
KSTDER= 240     /WRONG KEY WORD FOR SET MEDIA DENSITY COMMAND.
```

C. RETRIES: EACH WRITE OR READ STATUS ERROR IS ACCOMPANIED BY
A RETRY NUMBER:

RETRY 0: IS THE ORIGINAL ERROR
RETRY 1: IS THE FIRST RETRY OF THAT ERROR (SAME ADDRESS)
RETRY 2: IS THE SECOND RETRY OF THAT ERROR (SAME ADDRESS)
NOW DROP THE MODULE IF SR1=1 OR
CONTINUE TO NEXT ADDRESS IF SR1=0

10. DEVICE REGISTERS

CODE	FUNCTION
0	= FILL BUFFER
1	= EMPTY BUFFER
2	= WRITE SECTOR
3	= READ SECTOR
4	= SET DENSITY ** TAKES 15 SECONDS **
5	= READ MAINTENANCE STATUS
6	= WRITE SECTOR WITH DELETED DATA
7	= READ ERROR CODE

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
RXCS:	ERR	INT	XM	XM	RX2		SID	DEN	TR	IE	DON	DRV	FUN	FUN	FUN	GO	
WC:	X	X	X	X	X	X	X	X	WORD COUNT								
BA:	BASE ADDRESS																
RXES:	X	X	X	X	NXM	WC OVF	SID #1	DRV #1	DRV RDY	DEL DAT	DSK DEN	DEN ERR	AC LOW	INT DON	SID RDY	CRC	
RXTA:	X	X	X	X	X	X	X	X	0	TRACK ADDRESS							[0 -> 76, LEGAL]
RXSA:	X	X	X	X	X	X	X	X	0	0	0	SECTOR ADDRESS					[1 -> 26, LEGAL]

.ENDR

```
                        IOMODX  <RXBC >,177170,264,5,0,2,202,137,RBUF,64.,66.
                        MODULE  150000,RXBC ,177170,264,5,0,2,202,137,RBUF,64.,66.
                        .TITLE  RXBC DEC/X11 SYSTEM EXERCISER MODULE
                    ;   DDXCOM  VERSION  6      23-MAY-78
                                .LIST   BIN
                                ;*********************************************************************
000000'                         BEGIN:
000000' 054122  041502      040 MODNAM: .ASCII  /RXBC / ;MODULE NAME.
000005'    000                  XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 177170                  ADDR:   177170+0                ;1ST DEVICE ADDR.
000010' 000264                  VECTOR: 264+0                   ;1ST DEVICE VECTOR.
000012'    245                  BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013'    000                  BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014' 000003                  DVID1:  2+1                     ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN                    ;SWITCH REGISTER 4
                                ;*********************************************************************
000026' 150000                  STAT:   150000                  ;STATUS WORD.
000030' 000376'                 INIT:   START                   ;MODULE START ADDR.
000032' 000252'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                       ;PASS COUNTER.
000036' 000202                  ICONT:  202                     ;# OF ITERATIONS PER PASS=202
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'                         SRADR:                          ;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' 000666'                 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' 000137                  IDNUM:  137                     ;MODULE IDENTIFICATION NUMBER=137
000124' 005770'                 RBUFVA: RBUF                    ;READ BUFFER VIRTUAL ADDRESS
000126' 000000                  RBUFPA: OPEN                    ;READ BUFFER PHYSICAL ADDRESS
```



```
000130' 000000                  RBUFEA: OPEN                    ;READ BUFFER EA BITS
000132' 000100                  RBUFSZ: 64.                     ;SIZE OF THE READ BUFFER
000134' 000000                  WBUFPA: OPEN                    ;WRITE BUFFER PHYSICAL ADDRESS
000136' 000000                  WBUFEA: OPEN                    ;WRITE BUFFER EA BITS
000140' 000102                  WBUFRQ: 66.                     ;WRITE BUFFER SIZE REQUESTED
000142' 000000                  WBUFSZ: OPEN                    ;WRITE BUFFER SIZE AVAILABLE
000144' 000000                  CDERCT: OPEN                    ;CDATA/DATCK ERROR COUNT
000146' 000000                  CDWDCT: OPEN                    ;CDATA/DATCK WORD COUNT
000150' 000000                  FREE:   OPEN                    ;RESERVED FOR FUTURE USE
        000040                          .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                        .PLIST
                                        .WORD   0
                                        .LIST
                                        .ENDR
000252'                         MODSP:
                                ;*********************************************************************
```

```
                        ;----FLAGS AND COUNTERS ----------------------------------------------
                        DVID1X: 0                       ;HOLDS WHICH DRIVES TO TEST
000252' 000000          UNIT0:  0                       ;UNIT 0 FLAG
000254' 000000          UNIT1:  0                       ;UNIT 1 FLAG
000256' 000000          DRVN:   0                       ;DRIVE NUMBER
000260' 000000          CMD:    0                       ;COMMAND SAVE
000262' 000000          UTT:    0                       ;UNIT UNDER TEST
000264' 000000          UT:     0                       ;UNIT UNDER COMMAND
000266' 000000          WTF:    0                       ;WRITE FLAG
000270' 000000          TA:     0                       ;CURRENT TRACK ADDRESS
000272' 000000          SA:     0                       ;CURRENT SECTOR ADDRESS
000274' 000000          TASAV:  0                       ;STARTING TRACK ADDRESS SAVE
000276' 000000          SASAV:  0                       ;STARTING SECTOR ADDRESS SAVE
000300' 000000          WDCNT:  100                     ;WORD CCOUNTER
000302' 000100          DENSTY: 0                       ;DENSITY
000304' 000000          DEN:    0                       ;DENSITY FLAG
000306' 000000          TOMLT:  0                       ;TIME OUT MULTIPLIER
000310' 000000          TOCNT:  0                       ;TIME OUT COUNTER
000312' 000000          INT:    0                       ;INTIALIZE FLAG
000314' 000000          UTDONE: 0                       ;UNIT DONE FLAG
000316' 000000          TSSFLG: 0                       ;TSS DONE WITH UNIT FLAG
000320' 000000          INLOOP: 0                       ;IN LOOP FLAG
000322' 000000          ITBYPS: 0                       ;RESTART INIT BYPASS FLAG
000324' 000000          RKBYPS: 0                       ;BREAK BYPASS FLAG
000326' 000000          RTRYFL: 0                       ;RETRY FLAG
000330' 000000          NXERCT: 0                       ;NON EXISTENT MEMORY ERROR COUNTER
000332' 000000          SERFL:  0                       ;STATUS ERROR FLAG
000334' 000000          FIN:    0                       ;FINI FLAG
000336' 000000          ;-------------------------------------------------------------------------


                        ;----TEMPORARY STORAGE REGISTERS -------------------------------------

000340' 000000          SRXCS:  0
000342' 000000          SRXES:  0
000344' 000000          SRXTA:  0
000346' 000000          SRXSA:  0
000350' 000000          SRXSB:  0
000352' 000000          SRXSB1: 0
000354' 000000          SRXSB2: 0
000356' 000000          SRXSB3: 0
000360' 000350'         DECVA:  SRXSB                   ;DEFINITIVE ERROR CODE VIRTUAL ADR
000362' 000364'                 .+2                     ;DEFINITIVE ERROR CODE PHYSICAL ADR
000364' 000366'                 .+2                     ;DEFINITIVE ERROR CODE EXTENDED ADR BITS
                        ;-------------------------------------------------------------------------
                        ;----CONSTANTS -------------------------------------------------------

                        RXCS:   177170                  ;RX02 COMMAND REGISTER - ADDRESS
                        RXDB:   177172                  ;RX02 DATA BUFFER - ADDRESS
                        VEC:    264                     ;VECTOR
                        VARIFY: 111                     ;VARIFY WD FOR SET DENSITY=ASCII "I"
                        ;-------------------------------------------------------------------------
```



```
                        ;BEGINROUTINE (MODULE 0,0 -- CONTROL)
                        ;  IF START
                        ;  ;  INITIALIZE
                        ;  ;  CALL MOD 1,0 - INITIALIZE DEVICE
                        ;  ;  CALL MOD 6,0 - CHECK STATUS
                        ;  RESTART
                        ;  BEGINDO
                        ;  ;  IF INLOOP NOT SET
                        ;  ;  ;  THEN
                        ;  ;  ;     IF INITIALIZE_BYPASS NOT SET
                        ;  ;  ;     ;  THEN-INITIALIZE-TRACK & SECTOR
                        ;  ;  ;     ENDIF
                        ;  ;  ;     IF FINI_FLAG NOT SET
                        ;  ;  ;     ;  THEN
                        ;  ;  ;     ;     CALL MOD 2,0 - OUTPUT (WRITE)
                        ;  ;  ;     ;     IF FINI_FLAG NOT SET
                        ;  ;  ;     ;     ;  THEN
                        ;  ;  ;     ;     ;     CALL MOD 3,0 - INPUT (READ)
                        ;  ;  ;     ;     ;     IF FINI_FLAG NOT SET
                        ;  ;  ;     ;     ;     ;  THEN
                        ;  ;  ;     ;     ;     ;     CALL MOD 4,0 - PROCESS SECTOR
                        ;  ;  ;     ;     ;     ;     SET INLOOP_FLAG & CALL ITERATION
                        ;  ;  ;     ;     ;     ;  ENDIF
                        ;  ;  ;     ;     ;  ENDIF
                        ;  ;  ;     ;  ENDIF
                        ;  ;  ;     ENDIF
                        ;  ;  ;  ENDIF
                        ;  ;  CLEAR INLOOP
                        ;  ;  SET INITIALIZE_BYPASS FLAG
                        ;  ;  IF SECTOR_DONE_FLAG SET
                        ;  ;  ;  THEN
                        ;  ;  ;     CLEAR SECTOR_DONE_FLAG
                        ;  ;  ;     CALL MOD 8,0-UNITS DONE CHECK
                        ;  ;  ;     IF UNIT_DONE SET
                        ;  ;  ;     ;  THEN
                        ;  ;  ;     ;     SET TRACK_ADR=TRACK_ADR+10.
                        ;  ;  ;     ;     IF TRACK ADR > 76
                        ;  ;  ;     ;     ;  THEN-SET TRACK_ADR=TRACK_ADR SAVE
                        ;  ;  ;     ;     ;        INCREMENT SECTOR_ADDRESS_SAVE & CLEAR INITIALIZE_BYPASS_FLAG
                        ;  ;  ;     ;     ;        IF SECTOR_ADDRESS_SAVE=4
                        ;  ;  ;     ;     ;        ;  THEN
                        ;  ;  ;     ;     ;        ;     SET SECTOR_ADDRESS_SAVE=1
                        ;  ;  ;     ;     ;        ;     INCREMENT TRACK_ADDRESS_SAVE
                        ;  ;  ;     ;     ;        ;     IF TRACK_ADDRESS_SAVE=11.
                        ;  ;  ;     ;     ;        ;     ;  THEN
                        ;  ;  ;     ;     ;        ;     ;     SET TRACK_ADDRESS_SAVE=1
                        ;  ;  ;     ;     ;        ;     ;     COMMENT DENSITY_FLAG
                        ;  ;  ;     ;     ;        ;     ;     CALL MOD 7,0 - SET DENSITY
                        ;  ;  ;     ;     ;        ;     ENDIF
                        ;  ;  ;     ;     ;        ENDIF
                        ;  ;  ;     ;     ENDIF
                        ;  ;  ;     ENDIF
                        ;  ;  ENDIF
                        ;  ;  CALL MOD 5,0-SELECT UNIT FOR TEST
                        ;  DOUNTIL FINI=1
                        ;  FINI-->END
                        ;ENDROUTINE
```

```
                                        ;MOD 0.0 CONTROL -- PROCESS SECTORS ----------------------------------
000376' 000240                  START:  NOP
000400' 016706  177426                  MOV     SPOINT,SP       ;INITIALIZE STACK
000404' 012767  000001  177664          MOV     #1,TASAV        ;PRESET TRACK ADDRESS SAVE
000412' 012767  000001  177660          MOV     #1,SASAV        ;PRESET SECTOR ADDRESS
000420' 012767  000100  177654          MOV     #100,WDCNT      ;SET WORD COUNT FOR SINGLE DENSITY
000426' 005067  177704                  CLR     FIN             ;CLEAR FINI FLAG
000432' 005067  177650                  CLR     DEN             ;CLEAR DENSITY FLAG
000436' 005067  177642                  CLR     DENSTY          ;CLEAR DENSITY
000442' 005067  177606                  CLR     UNIT0           ;CLEAR UNIT#0 AVAIL FLAG
000446' 005067  177604                  CLR     UNIT1           ;CLEAR UNIT#1 AVAIL FLAG
000452' 005067  177606                  CLR     UTT             ;CLEAR UNIT UNDER TEST
000456' 005067  177576                  CLR     DRVN            ;CLEAR DRIVE #
000462' 005067  177646                  CLR     SFRFL           ;CLEAR STATUS ERR FLAG
000466' 005067  177640                  CLR     NXERCT          ;CLEAR NON-EXISTENT MEMORY ERROR COUNT
000472' 016767  177316  177552          MOV     DVID1,DVID1X    ;GET  DRIVE COUNT
000500' 122737  000015  000041          CMPB    #15,@#41        ;IF RX02 FLOPPY WAS LOAD MEDIUM
000506' 001033                          BNE     FIRST           ;THEN
000510' 105737  000040                  TSTB    @#40            ;IF DRIVE 0
000514' 001013                          BNE     D1              ;THEN
000516' 042767  000001  177526          BIC     #1,DVID1X       ;KILL TEST FOR DRIVE 0
000524' 032767  000001  177262          BIT     #1,DVID1        ;WAS IT TO BE TESTED?
000532' 001421                          BEQ     FIRST           ;BR IF NO
000534' 012767  000260  004662          MOV     #260,DNUM       ;SET DRIVE # FOR PRINT
000542' 000412                          BR      NUAV            ;BR IF NOT AVAIL MSG
000544' 042767  000002  177500  D1:     BIC     #2,DVID1X       ;KILL TEST FOR DRIVE 1
000552' 032767  000002  177234          BIT     #2,DVID1        ;IF WAS TO BE TESTED
000560' 001406                          BEQ     FIRST           ;THEN
000562' 012767  000261  004634          MOV     #261,DNUM       ;SET DRIVE #1 FOR PRINT
                        NUAV:
000570'                                 MSGN$,BEGIN,NUNT        ;ASCII MESSAGE CALL WITH COMMON HEADER
000570' 104403  000F00' 005510'
000576' 012767  000001  177510  FIRST:  MOV     #1,INT          ;SET INIT FLAG
000604' 004767  002470                  JSR     PC,VSET         ;GO SETUP ADDRESS & VECTOR
000610' 032777  004000  177550          BIT     #4000,@RXCS     ;IF DEVICE IS
000616' 001005                          BNE     GO              ;RX01 THEN
000620' 104403  000000' 005476'         MSGN$,BEGIN,DRP4        ;ASCII MESSAGE CALL WITH COMMON HEADER
000626' 000167  000324                  JMP     FINI            ;GO DROP MODULE
000632' 004767  000324          GO:     JSR     PC,INTIAL       ;INITIALIZE DEVICE---DO MOD 1.0
000636' 005767  177474                  TST     FIN             ;IF FINI FLAG
000642' 001145                          BNE     FINI            ;NOT SET THEN
000644' 004767  001364                  JSR     PC,STATUS       ;CHECK STATUS--------DO MOD 6.0
000650' 005767  177462                  TST     FIN             ;IF FINI FLAG
000654' 001140                          BNE     FINI            ;NOT SET, THEN
000656' 005067  177432                  CLR     INT             ;RESET INIT FLAG
000662' 004767  001256                  CALL    SFLD            ;GO SELECT UNIT---------DO MOD 5.0
000666' 000240          RESTRT: NOP
000670' 005767  177426          AGNDO:  TST     INLOOP          ;IF INLOOP
000674' 001042                          BNE     BKLOOP          ;NOT SET, THEN
000676' 005767  177422                  TST     ITBYPS          ;IF INITIALIZE BYPASS FLAG
000702' 001006                          BNE     EDBYPS          ;NOT SET, THEN
000704' 016767  177370  177362          MOV     SASAV,SA        ;PRESET SECTOR STARTING ADDRESS
000712' 016767  177360  177352          MOV     TASAV,TA        ;PRESET TRACK STARTING ADDRESS
000720' 005767  177412  EDBYFS: TST     FIN             ;IF FINI FLAG
000724' 001026                          BNE     BKLOOP          ;NOT SET, THEN
000726' 104415  000000' 000124'         GETPA$,BEGIN, RBUFVA            ;GET PHYSICAL ADDRESS FROM 18-BIT RBUFVA
000734' 104414  000000'                 GWBUF$, BEGIN           ;GET WRITE BUFFER INFORMATION
```



```
000740' 004767  000266          WRITE:  JSR     PC,OUTPUT       ;GO WRITE-------------DO MOD 2.0
000744' 005767  177366                  TST     FIN             ;TEST FINI FLAG
000750' 001014                          BNE     BKLOOP          ;IF SET THEN BR
000752' 004767  000602          READ:   JSR     PC,INPUT        ;GO READ--------------DO MOD 3.0
000756' 005767  177354                  TST     FIN             ;TEST FINI FLAG
000762' 001007                          BNE     BKLOOP          ;IF SET THEN BR
000764' 004767  001116                  CALL    TSS             ;GO UPDATE SECTOR - DO MOD 4.0
000770' 012767  000001  177324          MOV     #1,INLOOP       ;SET INLOOP FLAG
000776' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION,
                                                                ;MONITOR SHALL TEST END OF PASS
001002' 005067  177314          BKLOOP: CLR     INLOOP          ;CLEAR INLOOP FLAG
001006' 012767  000001  177310          MOV     #1,ITBYPS       ;SET INITIALIZE BYPASS FLAG
001014' 005767  177300                  TST     TSSFLG          ;IF UNIT TSS
001020' 001453                          BEQ     DOUNTL          ;IS DONE, THEN
001022' 005067  177272                  CLR     TSSFLG          ;CLEAR FLAG FOR TSS DONE
001026' 004767  002214                  CALL    UDNCK           ;CALL UNIT DONE CHECK - DO MOD 8.0
001032' 005767  177260                  TST     UTDONE          ;IF UNIT DONE
001036' 001442                          BEQ     1$              ;NOT SET, THEN
001040' 062767  000012  177224          ADD     #12,TA          ;INCREMENT TRACK ADDRESS
001046' 022767  000114  177216          CMP     #114,TA         ;SEE IF DONE TRACKS
001054' 103033                          BHIS    1$              ;IF NOT; BR
001056' 016767  177214  177206          MOV     TASAV,TA        ;RESET TRACK ADDRESS
001064' 005067  177234                  CLR     ITBYPS          ;CLEAR INITIALIZE BYPASS FLAG
001070' 005267  177204                  INC     SASAV           ;BUMP STARTING SECTOR ADDRESS
001074' 022767  000004  177176          CMP     #4,SASAV        ;IF SECTORS = DONE (THREE)
001102' 001020                          BNE     1$              ;THEN
001104' 012767  000001  177166          MOV     #1,SASAV        ;RESET STARTING SECTOR
001112' 005267  177160                  INC     TASAV           ;BUMP STARTING TRACK ADDRESS
001116' 022767  000013  177152          CMP     #11.,TASAV      ;IF TRACKS=DONE (TEN)
001124' 001007                          BNE     1$              ;THEN
001126' 012767  000001  177142          MOV     #1,TASAV        ;RESET TRACK STARTING ADDRESS
001134' 005167  177146                  COM     DEN             ;CHANGE DENSITY
001140' 004767  001570                  JSR     PC,DENCH        ;SET DEVICE DENSITY
001144' 004767  000774          1$:     CALL    SFLD            ;SELECT UNIT------DO MOD 5.0
001150' 005767  177162          DOUNTL: TST     FIN             ;DO UNTIL FINI FLAG
001154' 001644                          BEQ     RESTRT          ;SET
001156'                         FINI:
001156' 104410  000000'                 END$,BEGIN              ;DROP THE MODULE
                                        ;-------------------------------------------------------------
```

```
                                ;MOD 1.0  INTIALIZE DEVICE ----------------------------------
                                ;
001162' 012767  040000  177072  INTIAL: MOV     #40000,CMD      ;SET INT COMMAND
001170' 016777  177066  177170          MOV     CMD,@RXCS       ;INIT UNIT 0
001176' 004767  003142                  JSR     PC,AWDN         ;GO AWAIT DONE
001202' 005767  177130                  TST     FIN             ;IF FINI FLAG
001206' 001010                          BNE     ENDINT          ;EQUALS ZERO THEN
001210' 005777  177152                  TST     @RXCS           ;SEE IF ERROR ON INIT
001214' 100005                          BPL     ENDINT          ;IF NOT; BR
001216' 104403  000000' 005450'         MSGN8,BEGIN,DRP1        ;ASCII MESSAGE CALL WITH COMMON HEADER
001224' 004767  002564                  JSR     PC,TDRP         ;DO TEST DROP CLEAN-UP
001230' 000207                  ENDINT: RTS     PC              ;RETURN
                                ;-----------------------------------------------------------------
                                ;
                                ;MOD 2.0 WRITE SUBROUTINE -----------------------------------
                                ;
001232' 012767  000001  177030  OUTPUT: MOV     #1,WTF          ;SET WRITE FLAG
001240' 004767  000134                  JSR     PC,OUBUF2       ;THEN O/P RX02 BUFFER  - DO MOD 2.1
001244' 005767  177066                  TST     FIN             ;IF FINI FLAG
001250' 001058                          BNE     ENDOUT          ;EQUALS ZERO THEN
001252' 012767  000260  004150          MOV     #260,RTYN       ;RESET RETRY COUNTER-ASCII "0"
001260' 016767  177000  176774  18:     MOV     UTT,CMD         ;SELECT DRIVE
001266' 052767  000105  176766          BIS     #105,CMD        ;SET TO WRITE SECTOR + INT ENABLE
001274' 056767  177004  176760          BIS     DENSTY,CMD      ;SET DENSITY
001302' 016777  176754  177056          MOV     CMD,@RXCS       ;LOAD COMMAND
001310' 004767  003140                  JSR     PC,AWTR         ;GO AWAIT TRANSFER READY
001314' 005767  177016                  TST     FIN             ;IF FINI FLAG
001320' 001024                          BNE     ENDOUT          ;EQUALS ZERO THEN
001322' 016777  176746  177040          MOV     SA,@RXDB        ;LOAD SECTOR ADDRESS
001330' 004767  003120                  JSR     PC,AWTR         ;GO AWAIT TRANSFER READY
001334' 005767  176776                  TST     FIN             ;IF FINI FLAG
001340' 001014                          BNE     ENDOUT          ;EQUALS ZERO THEN
001342' 016777  176724  177020          MOV     TA,@RXDB        ;LOAD TRACK ADDRESS
001350' 004767  001764                  JSR     PC,INTERP       ;WAIT FOR INTERRUPT
001354' 005767  176756                  TST     FIN             ;IF FINI FLAG
001360' 001004                          BNE     ENDOUT          ;NOT SET, THEN
001362' 022767  000003  176740          CMP     #3,PTRYFL       ;IF RETRY FLAG
001370' 001333                          BNE     18              ;EQUALS 3 THEN
001372' 005067  176672          ENDOUT: CLR     WTF             ;CLEAR WRITE FLAG
001376' 000207                          RTS     PC              ;RETURN
                                ;-----------------------------------------------------------------
```



```
                                ;MOD 2.1  OUTPUT RX02 BUFFER ---------------------------------------
                                ;
001400' 012767  000001  176654  OUBUF2: MOV     #1,CMD          ;SET FILL BUFFER COMMAND
001406' 056767  176672  176646          BIS     DENSTY,CMD      ;SET DENSITY
001414' 016701  176516                  MOV     WBUFEA,R1       ;GET EXT. ADR. BITS
001420' 000301                          SWAB    R1              ;
001422' 050167  176634                  BIS     R1,CMD          ;SET EXT. ADR. BITS
001426' 016777  176630  176732          MOV     CMD,@RXCS       ;LOAD COMMAND
001434' 004767  003014                  JSR     PC,AWTR         ;WAIT FOR "TR"
001440' 005767  176672                  TST     FIN             ;IF FINI FLAG
001444' 001044                          BNE     ENDOU2          ;EQUALS ZERO THEN
001446' 016777  176630  176714          MOV     WDCNT,@RXDB     ;LOAD WORD COUNT FOR OUTPUT BUFFER
001454' 005267  176646                  INC     BKBYPS          ;SET BREAK BYPASS FLAG
001460' 004767  002770                  JSR     PC,AWTR         ;WAIT FOR "TR"
001464' 005067  176636                  CLR     BKBYPS          ;CLEAR BREAK BYPASS FLAG
001470' 005767  176642                  TST     FIN             ;IF FINI FLAG
001474' 001030                          BNE     ENDOU2          ;EQUALS ZERO THEN
001476' 016777  176432  176664          MOV     WBUFPA,@RXDB    ;LOAD BASE ADDRESS FOR OUTPUT BUFFER
001504' 004767  002634                  JSR     PC,AWDN         ;WAIT FOR "DONE"
001510' 005767  176622                  TST     FIN             ;IF FINI FLAG
001514' 001020                          BNE     ENDOU2          ;EQUALS ZERO THEN
001516' 004767  002442                  CALL    CKNXEP          ;CALL CHECK NON-EXISTENT MEMORY ERRORS
001522' 005777  176640                  TST     @RXCS           ;IF DEVICE ERROR BIT
001526' 100013                          BPL     ENDOU2          ;IS SET THEN
001530' 004767  002526                  CALL    SUPTRG          ;CALL SETUP PRINT REGISTERS
001534' 012767  000035  176344          MOV     #35,ERRTYP      ;SETUP FILL BUFFER ERROR
                                        ;*****************************************************************
001542' 104405  000000' 000000          HRDER8,BEGIN,NULL       ;BUFFER FILL ERROR
                                        ;*****************************************************************
001550' 104402  000000' 005442'         MSGS8,BEGIN,BFER2       ; ASCII MESSAGE CALL WITH NO HEADER
001556' 000207                  ENDOU2: RTS     PC              ;RETURN
                                ;-----------------------------------------------------------------
```

```
                                        ;MOD 3.0 READ SUBROUTINE ------------------------------------
                                        ;
001560' 012767  000260  003642  INPUT:  MOV     #260,RTYN       ;RESET RETRY COUNTER=ASCII "0"
001566' 016767  176472  176466  1$:     MOV     UTT,CMD         ;SELECT DRIVE
001574' 052767  000107  176460          BIS     #107,CMD        ;SET READ COMMAND + INT ENB
001602' 056767  176476  176452          BIS     DENSTY,CMD      ;SET DENSITY
001610' 016777  176446  176550          MOV     CMD,@RXCS       ;LOAD COMMAND
001616' 004767  002632                  JSR     PC,AWTR         ;GO AWAIT TRANSFER READY
001622' 005767  176510                  TST     FIN             ;IF FINI FLAG
001626' 001023                          BNE     ENDIN           ;NOT SET, THEN
001630' 016777  176446  176532          MOV     SA,@RXDB        ;LOAD SECTOR ADDRESS
001636' 004767  002612                  JSR     PC,AWTR         ;GO AWAIT TRANSFER READY
001642' 005767  176470                  TST     FIN             ;IF FINI FLAG
001646' 001013                          BNE     ENDIN           ;NOT SET, THEN
001650' 016777  176416  176512          MOV     TA,@RXDB        ;LOAD TRACK ADDRESS
001656' 004767  001456                  JSR     PC,INTER        ;WAIT FOR INTERRUPT
001662' 022767  000003  176440          CMP     #3,RTRYFL       ;IF RETRY FLAG
001670' 001336                          BNE     1$              ;EQUALS 3, THEN
001672' 004767  000002                  JSR     PC,INBUF2       ;THEN GET RX02 BUFFER  - DO MOD 3.1
001676' 000207          ENDIN:  RTS     PC              ;RETURN
                                        ;-----------------------------------------------------------
```



```
                                        ;MOD 3.1 INPUT RX02 BUFFER ----------------------------------
                                        ;
001700' 005767  176430  INBUF2: TST     SERFL           ;IF STATUS ERROR NOT SET
001704' 001077                  BNE     ENDIN2          ;THEN
001706' 012767  000003  176346          MOV     #3,CMD          ;SET EMPTY BUFFER COMMAND
001714' 056767  176364  176340          BIS     DENSTY,CMD      ;SET DENSITY
001722' 016701  176202                  MOV     RBUFEA,R1       ;GET EXT. ADR. BITS
001726' 000301                          SWAB    R1              ;
001730' 050167  176326                  BIS     R1,CMD          ;SET EXT. ADR. BITS
001734' 016777  176322  176424          MOV     CMD,@RXCS       ;ELSE LOAD COMMAND
001742' 004767  002506                  JSR     PC,AWTR         ;WAIT FOR "TR"     DO MOD U.TR
001746' 005767  176364                  TST     FIN             ;IF FINI FLAG
001752' 001054                          BNE     ENDIN2          ;EQUALS ZERO
001754' 016777  176322  176406          MOV     WDCNT,@RXDB     ;THEN LOAD WORD COUNT FOR INPUT BUFFER
001762' 005267  176340                  INC     BKBYPS          ;SET BREAK BYPASS FLAG
001766' 004767  002462                  JSR     PC,AWTP         ;WAIT FOR "TR"     DO MOD U.TR
001772' 005067  176334                  CLR     BKBYPS          ;CLEAR BREAK BYPASS FLAG
001776' 005767  176334                  TST     FIN             ;IF FINI FLAG
002002' 001040                          BNE     ENDIN2          ;EQUALS ZERO
002004' 016777  176116  176356          MOV     RBUFPA,@RXDB    ;THEN LOAD BASE ADDRESS FOR INPUT BUFFER
002012' 004767  002326                  JSR     PC,AWDN         ;WAIT FOR  "DONE"
002016' 005767  176314                  TST     FIN             ;IF FINI FLAG
002022' 001030                          BNE     ENDIN2          ;EQUALS ZERO THEN
002024' 005777  176336                  TST     @RXCS           ;IF DEVICE ERROR BIT
002030' 100021                          BPL     1$              ;SET, THEN
002032' 004767  002126                  CALL    CKNXER          ;CALL CHECK NON-EXISTENT MEMORY ERRORS
002036' 005777  176324                  TST     @RXCS           ;IF DEVICE ERROR BIT
002042' 100020                          BPL     ENDIN2          ;IS STILL SET, THEN
002044' 004767  002212                  CALL    SUPTRG          ;CALL SETUP PRINT REGISTERS
002050' 012767  000030  176030          MOV     #30,ERRTYP      ;SETUP DATA TRANSFER ERROR
                                        ;***********************************************************
002056' 104405  000000' 000000          HDR$,BEGIN,NULL         ;EMPTY BUFFER ERROR
                                        ;***********************************************************
002064' 104402  000000' 005434'         MSG$,BEGIN,BFER1        ; ASCII MESSAGE CALL WITH NO HEADER
002072' 000404                          BR      ENDIN2          ;BR TO EXIT
002074'                         1$:
002074' 104412  000000' 000126'         CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
002102' 002104'                         .+2                     ; IF ERROR, CONTINUE
002104' 000207          ENDIN2: RTS     PC              ;RETURN
                                        ;-----------------------------------------------------------
```

```
                                ;MOD 4.0 SECTOR UPDATE ------------------------------------------------
                                ;
002106' 062767  000003  176160  TSS:    ADD     #3,SA           ;BUMP SECTOR ADDRESS
002114' 022767  000033  176152          CMP     #33,SA          ;SEE IF DONE SECTORS
002122' 101401                          BLOS    1$              ;IF SO: BR
002124' 000406                          BR      ENDTSS          ;RETURN
002126' 016767  176146  176140  1$:     MOV     SASAV,SA        ;RESET SA
002134' 012767  000001  176156          MOV     #1,TSSFLG       ;SET TSS FLAG= (UNIT DONE WITH TSS)
002142' 000207                  ENDTSS: RTS     PC
                                ;---------------------------------------------------------------------
                                ;MOD 5.0 SELECT UNIT FOR TEST ----------------------------------
                                ;
002144' 005767  176110          SELD:   TST     DRVN            ;SEE IF DRIVE 0
002150' 001415                          BEQ     2$              ;IF SO: BR
002152' 005067  176102                  CLR     DRVN            ;ELSE SET TO 0
002156' 005767  176074                  TST     UNIT1           ;SEE IF UNIT 1 AVAILABLE
002162' 001401                          BEQ     1$              ;IF SO: BR
002164' 000767                          BR      SELD            ;GO SEE IF UNIT IS AVAILABLE
002166' 012767  000020  176070  1$:     MOV     #20,UTT         ;SET UNIT UNDER TEST
002174' 012767  000261  003222          MOV     #261,DNUM       ;SET DRIVE NUMBER FOR PRINTS
002202' 000413                          BR      ENDSEL          ;RETURN
002204' 005167  176050  2$:     COM     DRVN            ;SWITCH UNITS
002210' 005767  176040                  TST     UNIT0           ;SEE IF UNIT 0 AVAILABLE
002214' 001401                          BEQ     3$              ;IF SO: BR
002216' 000752                          BR      SELD            ;SELECT NEXT
002220' 005067  176040  3$:     CLR     UTT             ;SELECT DRIVE 0
002224' 012767  000260  003172          MOV     #260,DNUM       ;SET DRIVE NUMBER FOR PRINTS
002232' 000207                  ENDSEL: RTS     PC              ;RETURN
                                ;---------------------------------------------------------------------
```



```
                                ;MOD 6.0   DEVICE  STATUS ---------------------------------------
                                ;
002234' 000240                  STATUS: NOP
002236' 004767  000120          U0:     JSR     PC,UNT0         ;CHECK STATUS UNIT 0    - DO MOD 6.1
002242' 005767  176006                  TST     UNIT0           ;IF UNIT 0 IS
002246' 001012                          BNE     U1              ;AVAIL , THEN
002250' 005767  176040                  TST     INT             ;IF INIT FLAG
002254' 001407                          BEQ     U1              ;IS SET, THEN
002256' 005067  176002                  CLR     UTT             ;SET UNIT=0
002262' 012767  000260  003134          MOV     #260,DNUM       ;SET DRIVE # FOR PRINT
002270' 004767  000322                  JSR     PC,DNSET        ;SETUP DENSITY UNIT 0   - DO MOD 6.3
002274' 004767  000200          U1:     JSR     PC,UNT1         ;CHECK STATUS UNIT 1    - DO MOD 6.2
002300' 005767  175752                  TST     UNIT1           ;IF UNIT 1 IS
002304' 001014                          BNE     1$              ;AVAIL, THEN
002306' 005767  176002                  TST     INT             ;IF INIT FLAG
002312' 001422                          BEQ     ENDST           ;IS SET, THEN
002314' 012767  000020  175742          MOV     #20,UTT         ;SET UNIT=1
002322' 012767  000261  003074          MOV     #261,DNUM       ;SET DRIVE # FOR PRINT
002330' 004767  000262                  JSR     PC,DNSET        ;SETUP DENSITY UNIT 1   - DO MOD 6.3
002334' 000411                          BR      ENDST           ;BR TO END STATUS
002336' 005767  175712          1$:     TST     UNIT0           ;IF UNIT 0
002342' 001406                          BEQ     ENDST           ;NOT AVAIL, THEN
002344' 104403  000000' 005460'         MSGN$,BEGIN,DRP2        ;ASCII MESSAGE CALL WITH COMMON HEADER
002352' 012767  000001  175756          MOV     #1,FIN          ;GO DROP MODULE
002360' 000207                  ENDST:  RTS     PC              ;RETURN
                                ;---------------------------------------------------------------
                                ;MOD 6.1   UNIT 0 STATUS ---------------------------------------
                                ;
002362' 032767  000001  175662  UNT0:   BIT     #1,DVID1X       ;IF UNIT 0
002370' 001437                          BEQ     1$              ;SELECTED, THEN
002372' 012767  000013  175662          MOV     #13,CMD         ;READ STATUS UNIT 0
002400' 056767  175700  175654          BIS     DENSITY,CMD     ;SET DENSITY
002406' 016777  175650  175752          MOV     CMD,@RXCS       ;SEND COMMAND
002414' 004767  001724                  JSR     PC,AWDN         ;GO AWAIT DONE
002420' 005767  175712                  TST     FIN             ;IF FINI FLAG
002424' 001024                          BNE     ENDUT0          ;EQUALS ZERO THEN
002426' 017700  175736                  MOV     @RXDB,R0        ;GET RXES
002432' 032700  000200                  BIT     #200,R0         ;IF DRIVE RDY
002436' 001017                          BNE     ENDUT0          ;NOT SET, THEN
002440' 012767  000260  002756          MOV     #260,DNUM       ;SET DRIVE # FOR PRINT
002446' 012767  000006  175432          MOV     #6,ERRTYP
                                ;*********************************************************************
002454' 104405  000000' 000000          HDER$,BEGIN,NULL        ;UNIT 0 NOT AVAILABLE
                                ;*********************************************************************
002462' 104402  000000' 005510'         MSG$,BEGIN,NUNT         ; ASCII MESSAGE CALL WITH NO HEADER
002470' 012767  177777  175556  1$:     MOV     #-1,UNIT0       ;DESELECT UNIT 0
002476' 000207                  ENDUT0: RTS     PC              ;RETURN
                                ;---------------------------------------------------------------
```

```
740                                     ;MOD 6.2   UNIT 1 STATUS ----------------------------------------
741
742  002500' 032767  000002  175544 UNT1$:  BIT     #2,DVID1X       ;IF UNIT 1
743  002506' 001437                 BEQ     1$              ;SELECTED, THEN
744  002510' 012767  000033  175544         MOV     #33,CMD         ;READ STATUS UNIT 1
745  002516' 056767  175562  175536         BIS     DENSTY,CMD      ;SET DENSITY
746  002524' 016777  175532  175634         MOV     CMD,@RXCS       ;SEND COMMAND
747  002532' 004767  001606                 JSR     PC,AWDN         ;GO AWAIT DONE
748  002536' 005767  175574                 TST     FIN             ;IF FINI FLAG
749  002542' 001024                 BNE     ENDUT1          ;EQUALS ZERO THEN
750  002544' 017700  175620                 MOV     @RXDB,R0        ;READ RXES
751  002550' 032700  000200                 BIT     #200,R0         ;IF DRIVE RDY
752  002554' 001017                 BNE     ENDUT1          ;NOT SET, THEN
753  002556' 012767  000261  002640         MOV     #261,DNUM       ;SET DRIVE # FOR PRINT
754  002564' 012767  000006  175314         MOV     #6,ERRTYP
755                                     ;*****************************************************************
756  002572' 104405  000000' 000000         HDDER$,BEGIN,NULL       ;UNIT 1 NOT AVAILABLE
757                                     ;*****************************************************************
758  002600' 104402  000000' 005510'        MSGS$,BEGIN,NUNT        ; ASCII MESSAGE CALL WITH NO HEADER
759  002606' 012767  177777  175442 1$:     MOV     #-1,UNIT1       ;DESELECT UNIT 1
760  002614' 000207                 ENDUT1: RTS     PC              ;RETURN
761                                     ;---------------------------------------------------------------
762
763
764                                     ;MOD 6.3   DEVICE DENSITY SETUP ---------------------------------
765
766  002616' 012767  000013  175436 DNSET:  MOV     #13,CMD         ;SET READ STATUS
767  002624' 056767  175434  175430         BIS     UTT,CMD         ;SET UNIT
768  002632' 016777  175424  175526         MOV     CMD,@RXCS       ;SEND COMMAND
769  002640' 004767  001500                 JSR     PC,AWDN         ;GO AWAIT DONE
770  002644' 005767  175466                 TST     FIN             ;IF FINI FLAG
771  002650' 001030                 BNE     XDNSET          ;IS ZERO, THEN
772  002652' 012767  000011  175402         MOV     #11,CMD         ;SET SINGLE DENSITY
773  002660' 056767  175400  175374         BIS     UTT,CMD         ;SET UNIT
774  002666' 016777  175372  175472         MOV     CMD,@RXCS       ;SEND COMMAND
775  002674' 004767  001554                 JSR     PC,AWTR         ;GO AWAIT  "TR"
776  002700' 005767  175432                 TST     FIN             ;IF FINI FLAG IS
777  002704' 001012                 BNE     XDNSET          ;ZERO
778  002706' 016777  175462  175454         MOV     VARIFY,@RXDB    ;SEND VARIFY WORD
779  002714' 104403  000000' 005716'        MSGN$,BEGIN,SETSDN      ;ASCII MESSAGE CALL WITH COMMON HEADER
780  002722' 004767  001416                 JSR     PC,AWDN         ;WAIT FOR "DONE"
781  002726' 004767  000440                 JSR     PC,SECK         ;GO CHECK FOR ERROR
782  002732' 000207                 XDNSET: RTS     PC              ;RETURN
783                                     ;---------------------------------------------------------------
```



```
784                                     ;MOD 7.0   DENSITY CHANGE ---------------------------------------
785
786  002734' 000240                 DENCH:  NOP
787  002736' 005767  175314                 TST     UNIT1           ;IF UNIT 1
788  002742' 001007                 BNE     1$              ;IIS AVAIL
789  002744' 012767  000020  175314         MOV     #20,UT          ;SET TO DO UNIT 1
790  002752' 012767  000261  002444         MOV     #261,DNUM       ;SET DRIVE # FOR PRINT
791  002760' 000405                 BR      2$              ;CONTINUE
792  002762' 005067  175300         1$:     CLR     UT              ;ELSE SET TO DO UNIT 0
793  002766' 012767  000260  002430         MOV     #260,DNUM       ;SET DRIVE # FOR PRINT
794  002774' 005767  175306         2$:     TST     DEN             ;IF LENSITY FLAG
795  003000' 001051                 BNE     5$              ;EQUALS ZERO (SINGLE DENSITY) THEN
796  003002' 012767  000111  175252 3$:     MOV     #111,CMD        ;SETUP SET DENSITY = SINGLE COMMAND
797  003010' 056767  175252  175244         BIS     UT,CMD          ;SETUP UNIT UNDER TEST
798  003016' 016777  175240  175342         MOV     CMD,@RXCS       ;SEND COMMAND
799  003024' 004767  001424                 JSR     PC,AWTR         ;WAIT FOR "TR"
800  003030' 005767  175302                 TST     FIN             ;IF FINI FLAG
801  003034' 001103                 BNE     ENDDNC          ;EQUALS ZERO THEN
802  003036' 016777  175332  175324         MOV     VARIFY,@RXDB    ;SEND VARIFY COMMAND
803  003044' 104403  000000' 005734'        MSGN$,BEGIN,SSGLDN      ;ASCII MESSAGE CALL WITH COMMON HEADER
804  003052' 004767  000262                 JSR     PC,INTER        ;WAIT FOR "DONE" & INTERRUPT
805  003056' 005767  175254                 TST     FIN             ;IF FINI FLAG
806  003062' 001070                 BNE     ENDDNC          ;EQUALS ZERO THEN SET
807  003064' 005767  175176                 TST     UT              ;IF
808  003070' 001411                 BEQ     4$              ;UNIT 1 WAS DONE
809  003072' 005067  175170                 CLR     UT              ;SET TO DO UNIT 0
810  003076' 012767  000260  002320         MOV     #260,DNUM       ;SET DRIVE # FOR PRINT
811  003104' 005767  175144                 TST     UNIT0           ;IF UNIT 0
812  003110' 001061                 BNE     4$              ;IS AVAIL
813  003112' 000733                 BR      3$              ;SET UNIT 0 DENSITY
814  003114' 012767  000000  175162 4$:     MOV     #0,DENSTY       ;SET DENSITY=SINGLE
815  003122' 000450                 BR      ENDDNC          ;BRANCH TO END
816  003124' 012767  000511  175130 5$:     MOV     #511,CMD        ;ELSE SET DENSITY=DOUBLE COMMAND
817  003132' 056767  175130  175122         BIS     UT,CMD          ;SETUP UNIT UNDER TEST
818  003140' 016777  175116  175220         MOV     CMD,@RXCS       ;SEND COMMAND
819  003146' 004767  001302                 JSR     PC,AWTR         ;WAIT FOR "TR"
820  003152' 005767  175160                 TST     FIN             ;IF FINI FLAG
821  003156' 001032                 BNE     ENDDNC          ;EQUALS ZERO THEN
822  003160' 016777  175210  175202         MOV     VARIFY,@RXDB    ;SEND VARIFY COMMAND
823  003166' 104403  000000' 005750'        MSGN$,BEGIN,SDBLDN      ;ASCII MESSAGE CALL WITH COMMON HEADER
824  003174' 004767  000140                 JSR     PC,INTER        ;WAIT FOR "DONE" & INTERRUPT
825  003200' 005767  175132                 TST     FIN             ;IF FINI FLAG
826  003204' 001017                 BNE     ENDDNC          ;EQUALS ZERO THEN
827  003206' 005767  175054                 TST     UT              ;IF
828  003212' 001411                 BEQ     6$              ;UNIT 1 WAS DONE
829  003214' 005067  175046                 CLR     UT              ;SET TO DO UNIT 0
830  003220' 012767  000260  002176         MOV     #260,DNUM       ;SET DRIVE # FOR PRINT
831  003226' 005767  175022                 TST     UNIT0           ;IF UNIT 0
832  003232' 001001                 BNE     6$              ;AVIAL
833  003234' 000733                 BR      5$              ;CONTINUE
834  003236' 012767  000400  175040 6$:     MOV     #400,DENSTY     ;SET DENSITY = DOUBLE
835  003244' 000207                 ENDDNC: RTS     PC              ;RETURN
836                                     ;---------------------------------------------------------------
```

```
                                ;MOD 8.0 - UNIT DONE CHECK ----------------------------------------------

                        UDNCK:  TST     UTT             ;IF UNIT 0
                                BNE     1$              ;UNDER TEST, THEN
                                TST     UNIT1           ;IF UNIT 1
                                BNE     1$              ;AVAILABLE, THEN
                                CLR     UTDONE          ;CLEAR UNITS DONE FLAG
                                BR      XUDNCK          ;BR TO MOD EXIT
003270' 012767  000001  175020  1$:     MOV     #1,UTDONE       ;SET UNITS DONE FLAG
003276' 000207                  XUDNCK: RETURN                  ;RETURN
                                ;-----------------------------------------------------------------------

                                ;MOD U.VS VECTOR SET UP -------------------------------------------------

003300' 016700  174504          VSET:   MOV     VECTOR,R0
003304' 012720  003344'                 MOV     #NTRUPT,(R0)+   ;SET UP INTERRUPT HANDLER ADDRESS
003310' 116710  174476                  MOVB    BR1,(R0)        ;SET BR LEVEL
003314' 016767  174466  175044          MOV     ADDR,RXCS       ;SET ADDRESS OF RXCS
003322' 016767  174460  175040          MOV     ADDR,RXDB       ;GET ADDRESS OF RXCS
003330' 062767  000002  175032          ADD     #2,RXDB         ;SET ADDRESS OF DATA BUFFER
003336' 000207                          RTS     PC
                                ;-----------------------------------------------------------------------

                                ;MOD U.INTR  INTERRUPT HANDLER ------------------------------------------

003340'                         INTER:
003340' 104400  000000'                 EXIT$,REGIN                     ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.
003344'                         NTRUPT:
                                        ;---------------------------------------------------------------
003344' 000004  000000' 003352'         FIPQ$,REGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;---------------------------------------------------------------
003352' 004767  000014          1$:     JSR     PC,SECK         ;GO CHECK FOP ERROR
003356' 016700  174426                  MOV     VECTOR,R0       ;GET VECTOR ADDRESS
003362' 005720                          TST     (R0)+           ;BUMP POINTER
003364' 016710  174422                  MOV     BR1,(R0)        ;ASSURE RESET TO BR LEVEL 5
003370' 000207                          RTS     PC              ;RETURN
                                ;-----------------------------------------------------------------------
```



```
                                ;MOD U.EC       STATUS ERROR CHECK SUBROUTINE -------------------------

003372' 005067  174736          SECK:   CLR     SERFL           ;CLEAR STATUS ERROR FLAG
003376' 057767  174764  174656          BIS     @RXCS,CMD       ;GET RXCS
003404' 005767  174652                  TST     CMD             ;SEE IF ERROR
003410' 100404                          BMI     1$              ;IF SO: BR
003412' 012767  000003  174710          MOV     #3,RTRYFL       ;SET RETRY FLAG, SO NO RETRY
003420' 000423                          BR      ENDEC           ;ELSE RETURN
003422' 004767  000634          1$:     CALL    SUPTRG          ;CALL SETUP PRINT REGISTERS
003426' 016767  174632  174626          MOV     UTT,CMD         ;SELECT DRIVE
003434' 004767  000422                  CALL    DVEC            ;GET DEVICE ERROR CODE
003440' 004767  000026                  CALL    CKDNER          ;CHECK FOR DENSITY ERROR
003444' 005767  174620                  TST     WTF             ;SEE IF WRITE ERROR
003450' 001003                          BNE     2$              ;IF SO: BR
003452' 004767  000066                  CALL    CKRDER          ;CHECK READ ERROR
003456' 000402                          BR      CL              ;GO CLEAR ERROR
003460' 004767  000122          2$:     CALL    CKWTER          ;CALL CHECK WRITE ERROR
003464' 004767  000160          CL:     JSR     PC,CLEAR        ;CLEAR ERRORS
003470' 000207                  ENDEC:  RTS     PC              ;RETURN
                                ;-----------------------------------------------------------------------

                                ;MOD U,CKDNER - CHECK DENSITY ERRORS ------------------------------------

003472' 016705  174642          CKDNER: MOV     SRXCS,R5        ;SETUP RXCS FOR TEST
003476' 042705  177761                  BIC     #177761,R5      ;CLEAR ALL BUT COMMAND
003502' 022705  000010                  CMP     #10,R5          ;IF COMMAND WAS
003506' 001015                          BNE     XDNER           ;SET DENSITY, THEN
003510' 012767  000001  174620          MOV     #1,FIN          ;SET FINI FLAG
003516' 005767  174564                  TST     DEN             ;IF
003522' 001004                          BNE     1$              ;SINGLE DENSITY, THEN
003524' 104403  000000' 005520'         MSGN$,BEGIN,SETSDE      ;ASCII MESSAGE CALL WITH COMMON HEADER
003532' 000403                          BR      XDNER           ;CONTINUE
003534'                         1$:
003534' 104403  000000' 005536'         MSGN$,BEGIN,SETDDE      ;ASCII MESSAGE CALL WITH COMMON HEADER
003542' 000207                  XDNER:  RETURN                  ;RETURN
                                ;-----------------------------------------------------------------------

                                ;MOD U,CKRDER - CHECK READ ERRORS ---------------------------------------

003544' 012767  000000  174334  CKRDER: MOV     #0,ERRTYP
                                        ;***************************************************************
003552' 104406  000000' 004626'         SOFER$,BEGIN,TABLE      ;READ ERROR
                                        ;***************************************************************
003560' 005767  174522                  TST     DEN             ;IF DENSITY FLAG IS
003564' 001004                          BNE     1$              ;EQUAL TO ZERO (SINGLE) THEN
003566' 104402  000000' 005674'         MSGS$,BEGIN,RTERSD      ; ASCII MESSAGE CALL WITH NO HEADER
003574' 000403                          BR      2$              ;GO PRINT ERRORS
003576'                         1$:
003576' 104402  000000' 005652'         MSGS$,BEGIN,RTERDD      ; ASCII MESSAGE CALL WITH NO HEADER
003604' 000207                  2$:     RETURN                  ;RETURN
                                ;-----------------------------------------------------------------------
```

```
                                ;MOD U,CKWTER - CHECK WRITE ERRORS ----------------------------------

003606' 012767  000000  174272  CKWTER: MOV     #0,ERRTYP
                                ;*****************************************************************
003614' 104406  000000' 004626'         SOFERR$,BEGIN,TABLE     ;WRITE ERROR
                                ;*****************************************************************
003622' 005767  174460          TST     DEN             ;IF DENSITY FLAG IS
003626' 001004                  BNE     1$              ;EQUAL TO ZERO (SINGLE) THEN
003630' 104402  000000' 005630'         MSGS$,BEGIN,WTERSD      ; ASCII MESSAGE CALL WITH NO HEADER
003636' 000403                  BR      2$              ;GO PRINT ERRORS
003640'                         1$:
003640' 104402  000000' 005606'         MSGS$,BEGIN,WTERDD      ; ASCII MESSAGE CALL WITH NO HEADER
003646' 000207                  2$:     RETURN          ;RETURN
                                ;-----------------------------------------------------------------

                                ;MOD U,CLR   CLEAR ERRORS -------------------------------------------

003650' 005267  174460          CLEAR:  INC     SERFL           ;SET STATUS ERROR FLAG
003654' 105767  174470          TSTB    SRXSB           ;SEE IF R,E,C, ERROR
003660' 001421                  BEQ     2$              ;IF NOT: BR
003662' 012767  040000  174372  MOV     #40000,CMD      ;SET COMMAND= INIT
003670' 016777  174366  174470  MOV     CMD,@RXCS       ;REINITIALIZE
003676' 004767  000442          1$:     JSR     PC,AWDN         ;GO AWAIT DONE
003702' 005777  174460          TST     @RXCS           ;SEE IF ERROR ON INIT
003706' 100006                  BPL     2$              ;IF NOT: BR
003710' 104403  000000' 005466'         MSGN$,BEGIN,DRP3        ;ASCII MESSAGE CALL WITH COMMON HEADER
003716' 004767  000072          JSR     PC,TDRP         ;GO DROP MODULE
003722' 000431                  BR      ENDCLR
003724' 005267  001500          2$:     INC     RTYN            ;BUMP RETRY COUNTER
003730' 016703  001474          MOV     RTYN,R3         ;GET COUNTER
003734' 042703  177770          BIC     #177770,R3      ;MARK ASCII
003740' 010367  174364          MOV     R3,RTRYFL       ;SET RETRY FLAG
003744' 022703  000003          CMP     #3,R3           ;SEE IF DONE RETRIES
003750' 001016                  BNE     ENDCLR          ;IF NOT: BR
                                ;*****************************************************************
003752' 104405  000000' 004626'         HRDER$,BEGIN,TABLE      ;RECLASSIFICATION OF SOFT ERR
                                ;*****************************************************************
003760' 032767  000001  174030          BIT     #1,SR1          ;SEE IF SHOULD DROP MODULE
003766' 001404                  BEQ     3$              ;IF NOT: BR
003770' 012767  000001  174340          MOV     #1,FIN          ;GO DROP MODULE
003776' 000403                  BR      ENDCLR
004000' 012767  000260  001422  3$:     MOV     #260,RTYN       ;RESET RETRY COUNTER
004006' 005067  174336          ENDCLR: CLR     SRXSB           ;CLEAR ERROR
004012' 000207                  RTS     PC              ;RETURN
                                ;-----------------------------------------------------------------
```



```
                                ;MOD U,TDRP   TEST DROP CLEAN-UP  ----------------------------------

004014' 004767  000242          TDRP:   CALL    SUPTRG          ;CALL SETUP PRINT REGISTERS
004020' 005067  174236                  CLR     CMD             ;CLEAR COMMAND WORD
004024' 004767  000032                  CALL    DVEC            ;GET DEVICE ERROR CODE
004030' 005767  174302                  TST     FIN             ;IF FINI FLAG
004034' 001011                  BNE     ENDTDR          ;EQUALS ZERO THEN
004036' 012767  000034  174042  2$:     MOV     #34,ERRTYP
                                ;*****************************************************************
004044' 104405  000000' 004626'         HRDER$,BEGIN,TABLE      ;INITIALIZE ERROR
                                ;*****************************************************************
004052' 012767  000001  174256          MOV     #1,FIN          ;GO DROP MODULE
004060' 000207                  ENDTDR: RTS     PC
                                ;-----------------------------------------------------------------

                                ;MOD U,DVEC ----- GET DEVICE ERROR CODE -----------------------------

004062' 000240                  DVEC:   NOP                     ;
004064' 052767  000017  174170          BIS     #17,CMD         ;OR ERROR CODE COMMAND
004072' 104415  000000' 000360'         GFTPA$,BEGIN, DECVA             ;GET PHYSICAL ADDRESS FROM 16-BIT DECVA
004100' 016701  174256                  MOV     DECVA+2,R1      ;GET DEF E.C. PHYS ADR
004104' 016702  174254                  MOV     DECVA+4,R2      ;GET DEF E.C. XTEND ADR BITS
004110' 000302                  SWAB    R2              ;
004112' 050267  174144                  BIS     R2,CMD          ;SET EXTEND ADR BITS
004116' 016777  174140  174242          MOV     CMD,@RXCS       ;SEND COMMAND
004124' 005267  174176                  INC     BKBYPS          ;SET BREAK BYPASS FLAG
004130' 004767  000320                  JSR     PC,AWTR         ;THEN GO AWAIT IR
004134' 005067  174166                  CLR     BKBYPS          ;CLEAR BREAK BYPASS FLAG
004140' 005767  174172                  TST     FIN             ;IF FINI FLAG
004144' 001006                  BNE     EDDVEC          ;EQUALS ZERO THEN
004146' 010177  174216                  MOV     R1,@RXDB        ;SET BASE ADDR FOR LOAD EXT ERR REG.
004152' 004767  000166                  JSR     PC,AWDN         ;GO AWAIT DONE
004156' 004767  000002                  CALL    CKNXER          ;CALL CHECK NON-EXIIST MEMORY ERROR
004162' 000207                  EDDVEC: RTS     PC              ;RETURN
                                ;-----------------------------------------------------------------
```

```
1005
1006                                    ;MOD U,CKNXER - CHECK NON-EXISTENT MEMORY ERRORS ----------------------
1007
1008  004164' 032777  004000  174176  CKNXER: BIT     #4000,@PXDB     ;IF NONEXISTENT MEMORY
1009  004172' 001432                          BEQ     XNXER           ;ERROR, THEN
1010  004174' 004767  000062                  CALL    SUPTRG          ;CALL SETUP PRINT REGISTERS
1011  004200' 012777  040000  174160          MOV     #40000,@RXCS    ;INITIALIZE DEVICE (TO CLEAR ERROR)
1012  004206' 012767  000010  173672          MOV     #10,ERRTYP      ;SET NON-EXISTENT MEMORY ERROR
1013                                    ;*****************************************************************
1014  004214' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;NON-EXISTENT MEMORY ERROR
1015                                    ;*****************************************************************
1016  004222' 104402  005764' 000000'         MSGS$,NXERMS,BEGIN
1017  004230' 005267  174076                  INC     NXERCT          ;INCREMENT NONEXISTENT ERROR COUNT MEMORY
1018  004234' 022767  000004  174070          CMP     #4,NXERCT       ;IF NON-EXISTENT MEMORY
1019  004242' 101006                          BHI     XNXER           ;ERROR COUNT=3, THEN
1020  004244' 012767  000001  174064  1$:     MOV     #1,FIN          ;SET FINI FLAG
1021  004252' 104402  000000' 005504'        MSGS$,BEGIN,DPP5        ; ASCII MESSAGE CALL WITH NO HEADER
1022  004260' 000207                  XNXER:  RETURN                  ;RETURN
1023                                    ;-----------------------------------------------------------------
1024
1025                                    ;MOD U,SUPG - SETUP PRINT REGISTERS -----------------------------
1026
1027  004262' 017767  174100  174050  SUPTRG: MOV     @RXCS,SRXCS     ;GET RXCS
1028  004270' 056767  173766  174042          BIS     CMD,SRXCS       ;SET SRXCS TO SHOW COMMAND
1029  004276' 017767  174066  174036          MOV     @RXDB,SRXES     ;GET RXES
1030  004304' 016767  173764  174034          MOV     SA,SRXSA        ;GET SECTOR ADDRESS
1031  004312' 016767  173754  174024          MOV     TA,SRXTA        ;SET TRACK ADDRESS
1032  004320' 016767  174014  173554          MOV     SRXCS,ACSR      ;SET CONTENTS OF RXCS
1033  004326' 016767  174034  173544          MOV     RXCS,CSRA       ;SET ADDRESS OF RXCS
1034  004334' 016767  174002  173542          MOV     SRXES,ASTAT     ;GET RXES
1035  004342' 000207                          RETURN                  ;RETURN
1036                                    ;-----------------------------------------------------------------
```



```
1037                                    ;MOD U,WDN AWAIT DONE BIT SUBROUTINE ----------------------------
1038
1039  004344' 005067  173740          AWDN:   CLR     TOMLT           ;RESET TIME OUT MULTIPLIER
1040  004350' 005067  173736          1$:     CLR     TOCNT           ;PRESET TIME OUT COUNTER
1041  004354' 032777  000040  174004  2$:     BIT     #40,@RXCS       ;SEE IF DONE SET
1042  004362' 001033                          BNE     3$              ;IF SO: BR
1043  004364' 104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
1044  004370' 104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
1045  004374' 005267  173712                  INC     TOCNT           ;BUMP TIME OUT COUNTER
1046  004400' 001365                          BNE     2$              ;IF NOT TIMED OUT: BR
1047  004402' 005267  173702                  INC     TOMLT           ;INCREMENT TIMEOUT MULTIPLIER
1048  004406' 022767  000002  173674          CMP     #2,TOMLT        ;IF ON 2ND
1049  004414' 101355                          BHI     1$              ;TIMEOUT PASS, THEN
1050  004416' 004767  177640                  CALL    SUPTRG          ;CALL SETUP PRINT REGISTERS
1051  004422' 012767  000011  173456          MOV     #11,ERRTYP
1052                                    ;*****************************************************************
1053  004430' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;DONE BIT TIME OUT
1054                                    ;*****************************************************************
1055  004436' 104402  000000' 005554'        MSGS$,BEGIN,DMTO        ; ASCII MESSAGE CALL WITH NO HEADER
1056  004444' 012767  000001  173664          MOV     #1,FIN          ;DROP MODULE
1057  004452' 000207                  3$:     RTS     PC              ;EXIT
1058                                    ;-----------------------------------------------------------------
1059
1060                                    ;MOD U,ATR AWAIT TRANSFER READY SUBROUTINE ----------------------
1061
1062  004454' 005067  173630          AWTR:   CLR     TOMLT           ;PRESET TIMEOUT MULTIPLIER
1063  004460' 005067  173626          1$:     CLR     TOCNT           ;PRESET TIME OUT COUNTER
1064  004464' 032777  000040  173674  2$:     BIT     #40,@RXCS       ;IF DONE BIT NOT
1065  004472' 001026                          BNE     4$              ;SET, THEN
1066  004474' 032777  000200  173664          BIT     #200,@RXCS      ;SEE IF TRANSFER READY SET
1067  004502' 001046                          BNE     6$              ;IF SO: BR
1068  004504' 005767  173616                  TST     BKBYPS          ;IF BREAK BYPASS FLAG
1069  004510' 001004                          BNE     3$              ;NOT SET, THEN
1070  004512' 104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
1071  004516' 104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
1072  004522' 005267  173564          3$:     INC     TOCNT           ;BUMP TIME OUT COUNTER
1073  004526' 001356                          BNE     2$              ;IF NOT TIMED OUT: BR
1074  004530' 005267  173554                  INC     TOMLT           ;INCREMENT TIMEOUT MULTIPLIER
1075  004534' 022767  000002  173546          CMP     #2,TOMLT        ;IF ON 2ND
1076  004542' 101346                          BHI     1$              ;TIMEOUT PASS, THEN
1077  004544' 004767  177512                  CALL    SUPTRG          ;CALL SETUP PRINT REGISTERS
1078  004550' 012767  000040  173330  4$:     MOV     #40,ERRTYP
1079                                    ;*****************************************************************
1080  004556' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;TRANSFER READY TIME OUT
1081                                    ;*****************************************************************
1082  004564' 012767  000001  173544          MOV     #1,FIN          ;GO DROP MODULE
1083  004572' 032777  000040  173566          BIT     #40,@RXCS       ;IF DONE BIT
1084  004600' 001404                          BEQ     5$              ;SET, THEN
1085  004602' 104402  000000' 005574'        MSGS$,BEGIN,TRDNER      ; ASCII MESSAGE CALL WITH NO HEADER
1086  004610' 000403                          BR      6$              ;EXIT
1087  004612'                         5$:
1088  004612' 104402  000000' 005564'        MSGS$,BEGIN,TRTO        ; ASCII MESSAGE CALL WITH NO HEADER
1089  004620' 005067  173562          6$:     CLR     BKBYPS          ;CLEAR BREAK BYPASS FLAG
1090  004624' 000207                          RTS     PC              ;RETURN
1091                                    ;-----------------------------------------------------------------
```

```
1092                                    ;---- MESSAGE TABLE ----------
1093
1094  004626'                           TABLE:
1095  004626' 000340'                   ARXCS:  SRXCS
1096  004630' 000342'                   ARXES:  SRXES
1097  004632' 000344'                   ARXTA:  SRXTA
1098  004634' 000346'                   ARXSA:  SRXSA
1099  004636' 000350'                   ARXSB:  SRXSB
1100  004640' 000352'                   ARXSB1: SRXSB1
1101  004642' 000354'                   ARXSB2: SRXSB2
1102  004644' 000356'                   ARXSB3: SRXSB3
1103  004646' 177777                           -1
1104                                    ;-----------------------------------------------------------------
1105
1106                                    ;---- MESSAGES ---------------------------------------------------
1107
1108  004650' 054122  030460  051440    MSG1:   .ASCIZ  'RX01 SUBSYSTEM'
1109  004656' 041125  054523  052123
1110  004664' 046505      000
1111  004667'     040  044524  042515    MSG2:   .ASCIZ  ' TIME OUT'
1112  004674' 047440  052125      000
1113  004701'     040  051104  053111    MSG3:   .ASCIZ  ' DRIVE '
1114  004706' 020105      000
1115  004711'     040  037055  051104    MSG4:   .ASCIZ  ' ->DROP MODULE%'
1116  004716' 050117  046440  042117
1117  004724' 046125  022505      000
1118  004731'     040  047111  052111    MSG5:   .ASCIZ  ' INITIALIZE'
1119  004736' 040511  044514  042532
1120  004744'     000
1121  004745'     040  047516  052440    MSG6:   .ASCIZ  ' NO UNITS TO TEST'
1122  004752' 044516  051524  052040
1123  004760' 020117  042524  052123
1124  004766'     000
1125  004767'     045  051040  044505    MSG7:   .ASCIZ  '% REINITIALIZE'
1126  004774' 044516  044524  046101
1127  005002' 055111  000105
1128  005006' 051440  052105  026440    MSG8:   .ASCIZ  ' SET => WAITING ON'
1129  005014' 020076  040527  052111
1130  005022' 047111  020107  047117
1131  005030'     000
1132  005031'     040  044506  046114    MSG9:   .ASCIZ  ' FILL BUFFER'
1133  005036' 041040  043125  042506
1134  005044' 000122
1135  005046' 051440  047111  046107    MSG11:  .ASCIZ  ' SINGLE'
1136  005054' 000105
1137  005056' 042040  052517  046102    MSG12:  .ASCIZ  ' DOUBLE'
1138  005064' 000105
1139  005066' 042040  047117  020105    MSG13:  .ASCIZ  ' DONE BIT'
1140  005074' 044502  000124
1141  005100' 052040  040522  051516    MSG14:  .ASCIZ  ' TRANSFER READY'
1142  005106' 042506  020122  042522
1143  005114' 042101  000131
1144  005120' 042040  047105  044523    MSG15:  .ASCIZ  ' DENSITY'
1145  005126' 054524      000
1146  005131'     040  051105  047522    MSG17:  .ASCIZ  ' ERROR= RETRY:'
1147  005136' 026522  051040  052105
```



```
1148  005144' 054522  000072
1149  005150' 020045  042523  000124    MSG18:  .ASCIZ  '% SET'
1150  005156' 020045  051127  052111    MSG19:  .ASCIZ  '% WRITE'
1151  005164' 000105
1152  005166' 020045  042522  042101    MSG20:  .ASCIZ  '% READ'
1153  005174'     000
1154  005175'     040  046505  052120    MSG21:  .ASCIZ  ' EMPTY BUFFER'
1155  005202' 020131  052502  043106
1156  005210' 051105      000
1157  005213'     040  047516  020124    MSG22:  .ASCIZ  ' NOT AVIALABLE'
1158  005220' 053101  040511  040514
1159  005226' 046102  000105
1160  005232' 042440  051122  051117    MSG23:  .ASCIZ  ' ERROR'
1161  005240'     000
1162  005241'     045  026440  051476    MSG24:  .ASCIZ  '% ->SETTING DISKETTE TO'
1163  005246' 052105  044524  043516
1164  005254' 042040  051511  042513
1165  005262' 052124  020105  047524
1166  005270'     000
1167  005271'     045  020040  044440    MSG25:  .ASCIZ  '%   IF INTERRUPTED-> MAY NEED REFORMAT%'
1168  005276' 020106  047111  042524
1169  005304' 051122  050125  042524
1170  005312' 026504  020076  030515
1171  005320' 020131  044516  042105
1172  005326' 051040  043105  051117
1173  005334' 040515  022524      000
1174  005341'     045      000          MSG26:  .ASCIZ  '%'
1175  005343'     055  020076  047516    MSG27:  .ASCIZ  '-> NONEXISTENT MEMORY ERROR - INITIALIZE DEVICE'
1176  005350' 042516  044530  052123
1177  005356' 047105  020124  042515
1178  005364' 047515  054522  042440
1179  005372' 051122  051117  026440
1180  005400' 044440  044516  044524
1181  005406' 046101  055111  020105
1182  005414' 042504  044526  042523
1183  005422'     000
1184          005424'                   .EVEN
1185  005424' 030040      000          DNUM:   .ASCIZ  ' 0'
1186          005430'                   .EVEN
1187  005430' 030040      000          RTYN:   .ASCIZ  ' 0'
1188          005434'                   .EVEN
1189                                    ;-----------------------------------------------------------------
```

```
1190                            ;---- FORMATTED MESSAGES ----------------------------------------------
1191  005434' 005175'  BFER1:  MSG21                   ;"EMPTY BUFFER"
1192  005436' 005232'          MSG23                   ;"ERROR"
1193  005440' 177777           -1
1194  005142' 005031'  BFER2:  MSG9                    ;"FILL BUFFER"
1195  005444' 005232'          MSG23                   ;"ERROR"
1196  005446' 177777           -1
1197  005450' 004731'  DRP1:   MSG5                    ;"INITIALIZE"
1198  005452' 005232'          MSG23                   ;"ERROR"
1199  005454' 004711'          MSG4                    ;"DROP MODULE"
1200  005456' 177777           -1
1201  005460' 004745'  DRP2:   MSG6                    ;"NO UNITS TO TEST"
1202  005462' 004711'          MSG4                    ;"DROP MODULE"
1203  005464' 177777           -1
1204  005466' 004767'  DRP3:   MSG7                    ;"REINITIALIZE"
1205  005470' 005232'          MSG23                   ;"ERROR"
1206  005472' 004711'          MSG4                    ;"DROP MODULE"
1207  005474' 177777           -1
1208  005476' 004650'  DRP4:   MSG1                    ;"RX01 SUBSYSTEM"
1209  005500' 004711'          MSG4                    ;"DROP MODULE
1210  005502' 177777           -1
1211  005504' 004711'  DRP5:   MSG4                    ;"DROP MODULE"
1212  005506' 177777           -1
1213  005510' 004701'  MUNT:   MSG3                    ;"DRIVE"
1214  005512' 005424'          DNUM                    ;"#"
1215  005514' 005213'          MSG22                   ;"NOT AVAILABLE"
1216  005516' 177777           -1
1217  005520' 004701'  SETSDE: MSG3                    ;"DRIVE"
1218  005522' 005424'          DNUM                    ;"#"
1219  005524' 005150'          MSG18                   ;"SET"
1220  005526' 005046'          MSG11                   ;"SINGLE"
1221  005530' 005120'          MSG15                   ;"DENSITY"
1222  005532' 005232'          MSG23                   ;"ERROR"
1223  005534' 177777           -1
1224  005536' 004701'  SETDDE: MSG3                    ;"DRIVE"
1225  005540' 005424'          DNUM                    ;"#"
1226  005542' 005150'          MSG18                   ;"SET"
1227  005544' 005056'          MSG12                   ;"DOUBLE"
1228  005546' 005120'          MSG15                   ;"DENSITY"
1229  005550' 005232'          MSG23                   ;"ERROR"
1230  005552' 177777           -1
1231  005554' 005066'  DNTO:   MSG13                   ;"DONE BIT"
1232  005556' 004667'          MSG2                    ;"TIME OUT"
1233  005560' 004711'          MSG4                    ;"DROP MODULE"
1234  005562' 177777           -1
1235  005564' 005100'  TRTO:   MSG14                   ;"TRANSFER READY"
1236  005566' 004667'          MSG2                    ;"TIME OUT"
1237  005570' 004711'          MSG4                    ;"DROP MODULE"
1238  005572' 177777           -1
1239  005574' 005066'  TPDNER: MSG13                   ;;"DONE BIT"
1240  005576' 005006'          MSG8                    ;"SET => WAITING ON"
1241  005600' 005100'          MSG14                   ;"TRANSFER READY"
1242  005602' 004711'          MSG4                    ;"=>DROP MODULE"
1243  005604' 177777           -1
1244  005606' 004701'  WTERDD: MSG3                    ;"DRIVE"
1245  005610' 005424'          DNUM                    ;"#"
```



```
1246  005612' 005156'          MSG19                   ;"WRITE"
1247  005614' 005056'          MSG12                   ;"DOUBLE"
1248  005616' 005120'          MSG15                   ;"DENSITY"
1249  005620' 005131'          MSG17                   ;"ERROR=RETRY:"
1250  005622' 005430'          RTYN                    ;"#"
1251  005624' 005341'          MSG26                   ;"CRLF"
1252  005626' 177777           -1
1253  005630' 004701'  WTERSD: MSG3                    ;"DRIVE"
1254  005632' 005424'          DNUM                    ;"#"
1255  005634' 005156'          MSG19                   ;"WRITE"
1256  005636' 005046'          MSG11                   ;"SINGLE"
1257  005640' 005120'          MSG15                   ;"DENSITY"
1258  005642' 005131'          MSG17                   ;"ERROR=RETRY:"
1259  005644' 005430'          RTYN                    ;"#"
1260  005646' 005341'          MSG26                   ;"CRLF"
1261  005650' 177777           -1
1262  005652' 004701'  RTERDD: MSG3                    ;"DRIVE"
1263  005654' 005424'          DNUM                    ;"#"
1264  005656' 005166'          MSG20                   ;"READ"
1265  005660' 005056'          MSG12                   ;"DOUBLE"
1266  005662' 005120'          MSG15                   ;"DENSITY"
1267  005664' 005131'          MSG17                   ;"ERROR=RETRY:"
1268  005666' 005430'          RTYN                    ;"#"
1269  005670' 005341'          MSG26                   ;"CRLF"
1270  005672' 177777           -1
1271  005674' 004701'  RTERSD: MSG3                    ;"DRIVE"
1272  005676' 005424'          DNUM                    ;"#"
1273  005700' 005166'          MSG20                   ;"READ"
1274  005702' 005046'          MSG11                   ;"SINGLE"
1275  005704' 005120'          MSG15                   ;"DENSITY"
1276  005706' 005131'          MSG17                   ;"ERROR=RETRY:"
1277  005710' 005430'          RTYN                    ;"#"
1278  005712' 005341'          MSG26                   ;"CRLF"
1279  005714' 177777           -1
1280  005716' 004701'  SETSDN: MSG3                    ;"DRIVE"
1281  005720' 005424'          DNUM                    ;"#"
1282  005722' 005241'          MSG24                   ;"SETTING DISKETTE TO"
1283  005724' 005046'          MSG11                   ;"SINGLE"
1284  005726' 005120'          MSG15                   ;"DENSITY"
1285  005730' 005271'          MSG25                   ;"IF INTERRUPTED=MAY NEED REFORMAT"
1286  005732' 177777           -1
1287  005734' 004701'  SSGLDN: MSG3                    ;"DRIVE"
1288  005736' 005424'          DNUM                    ;"#"
1289  005740' 005241'          MSG24                   ;"SETTING DISKETTE TO"
1290  005742' 005046'          MSG11                   ;"SINGLE"
1291  005744' 005120'          MSG15                   ;"DENSITY"
1292  005746' 177777           -1
1293  005750' 004701'  SDBLDN: MSG3                    ;"DRIVE"
1294  005752' 005424'          DNUM                    ;"#"
1295  005754' 005241'          MSG24                   ;"SETTING DISKETTE TO"
1296  005756' 005056'          MSG12                   ;"DOUBLE"
1297  005760' 005120'          MSG15                   ;"DENSITY"
1298  005762' 177777           -1
1299  005764' 005343'  NXERMS: MSG27                   ;"=>NON EXISTENT MEMORY ERROR"
1300  005766' 177777           -1
1301                            ;---------------------------------------------------------------------
```

```
 1302                                ;---- DATA BUFFERS ----------------------------------------------------
 1303
 1304                                        .EVEN
 1305  005770' 000000       RBUF:   0                       ;READ BUFFER
 1306          006372'               .=.+400
 1307                                ;-------------------------------------------------------------------
 1308          000001                .END
```



```
ACSR    000102R          293#  1032*
ADDR    000006R          259#   854    855
ADDR22= 001000           322#
ARXCS   004626R         1095#
ARXES   004630R         1096#
ARXSA   004634R         1098#
ARXSB   004636R         1099#
ARXSB1  004640R         1100#
ARXSB2  004642R         1101#
ARXSB3  004644R         1102#
ARXTA   004632R         1097#
ASB     000106R          297#
ASTAT   000104R          295#  1034*
AWAS    000110R          298#
AWDN    004344R          530    585    640    725    747    769    780    947   1001   1039#
AWTP    004454R          551    555    575    580    606    610    630    635    775    799    819    996   1062#
BEGIN   000000R          256#   461    466    485    486    495    524    535    594    596    651    653    656
                         712    734    736    756    758    779    803    823    863    866    904    907    915
                         919    922    929    933    936    950    960    979    989   1014   1016   1021   1043
                        1044   1053   1055   1070   1071   1080   1085   1088
BFER1   005434R          653   1191#
BFER2   005442R          596   1194#
BGNDO   004670R          477#
BIT0  = 000001           322#
BIT1  = 000002           322#
BIT10 = 002000           322#
BIT11 = 004000           322#
BIT12 = 010000           322#
BIT13 = 020000           322#
BIT14 = 040000           322#
BIT15 = 100000           322#
BIT2  = 000004           322#
BIT3  = 000010           322#
BIT4  = 000020           322#
BIT5  = 000040           322#
BIT6  = 000100           322#
BIT7  = 000200           322#
BIT8  = 000400           322#
BIT9  = 001000           322#
BKBYPS  000326R          346#   579*   581*   634*   636*   995*   997*  1068   1089*
BKLOOP  001002R          478    484    489    492    497#
BREAK$= 104407           322#  1043   1044   1070   1071
BR1     000012R          261#   853    871
BR2     000013R          262#
BTOD$ = 104421           322#
CDATA$= 104412           322#   656
CDERCT  000144R          312#
CDWDCT  000146R          313#
CKDNEP  003472R          885*   897#
CKNXEF  004164R          588*   645*  1002*  1008#
CKRDER  003544R          888*   913#
CKWTEP  003606R          890*   927#
CL      003464R          889    891#
CLEAR   003650R          891    942#
CMD     000262R          328#   528*   529    547*   548*   549*   550    569*   570*   573*   574    602*   603*
                         604*   605    624*   625*   628*   629    722*   723*   724    744*   745*   746    766*
```

```
                        767*   768    772*   773*   774    796*   797*   798    816*   817*   818    877*   878
                        883*   945*   946    973*   988*   993*   994    1028
CONFIG  000056R         281#
CSRA    000100R         291#   1033*
DATCK$= 104411          322#
DATER$= 104404          322#
DECVA   000360R         364#   989    990    991
DEN     000306R         338#   438*   518*   794    902    917    931
DENCH   002734R         519    786#
DENSTY  000304R         337#   439*   549    570    604    625    723    745    814*   834*
DNSET   002616R         700    708    766#
DNTO    005554R        1055   1231#
DNUM    005424R         454*   459*   680*   687*   699*   707*   731*   753*   790*   793*   810*   830*   1185#
                       1214   1218   1225   1245   1254   1263   1272   1281   1288   1294
DOUNTL  001150R         500    521#
DRP1    005454R         535    1197#
DRP2    005460R         712    1201#
DRP3    005466R         950    1204#
DRP4    005476R         466    1208#
DRP5    005504R        1021   1211#
DRVN    000260R         327#   443*   673    675*   682*
DVEC    004062R         884*   974*   987#
DVID1   000014R         263#   446    452    457
DVID1X  000252R         324#   446*   451*   456*   720    742
D1      000544R         450    456#
EDBYPS  000720R         480    483#
EDDVEC  004162R         999    1003#
ENDCLR  003006R         952    958    965    967#
ENDDNC  003244R         801    806    815    821    826    835#
ENDEC   003470R         881    892#
ENDIN   001676R         608    612    618#
ENDINT  001230R         532    534    537#
ENDIN2  002104R         623    632    638    642    647    654    658#
ENDIT$= 104413          322#   495
ENDOUT  001372R         545    553    557    561    564#
ENDOU2  001556R         577    583    587    590    597#
ENDSEL  002232R         681    688#
ENDST   002360R         705    709    711    714#
ENDTDR  004060R         976    982#
ENDTSS  002142R         665    668#
ENDUT0  002476R         727    730    738#
ENDUT1  002614R         749    752    760#
END$  = 104410          322#   524
ERRTYP  000106R         296#   592*   649*   732*   754*   913*   927*   977*   1012*  1051*  1078*
EXIT$ = 104400          322#   863
FIN     000336R         350#   437*   469    472    483    488    491    521    531    544    552    556    560
                        576    582    586    607    611    631    637    641    713*   726    748    770    776
                        800    805    820    825    901*   964*   975    981*   998    1020*  1056*  1082*
FINI    001156R         467    470    473    523#
FIRST   000576R         448    453    458    462#
FREE    000150R         314#
GETPAS= 104415          322#   485    989
GO      000632R         465    468#
GWBUFS= 104414          322#   486
HRDCNT  000044R         276#
HRDERS= 104405          322#   594    651    734    756    960    979    1014   1053   1080
```



```
HRDPAS  000050R         278#
ICONT   000036R         273#
ICOUNT  000040R         274#
IDNUM   000122R         303#
IMODX.= 000000          315*   487
INBUF2  001700R         617    622#
INIT    000030R         270#
INLOOP  000322R         344#   477    494*   497*
INPUT   001560R         490    601#
INT     000314R         341#   462*   474*   696    704
INTEP   003340R         559    614    804    824    862#
INTIAL  001162R         468    528#
INTR    000120R         302#
ITBYPS  000324R         345#   479    498*   509*
MAP22$= 104416          322#
MODNAM  000000R         257#
MODSP   000252R         271    320#
MSGN$ = 104403          322#   461    466    535    712    779    803    823    904    907    950
MSGS$ = 104402          322#   596    653    736    758    919    922    933    936    1016   1021   1055   1085
                       1088
MSG$  = 104401          322#
MSG1    004650R        1108#  1208
MSG11   005046R        1135#  1220   1256   1274   1283   1290
MSG12   005056R        1137#  1227   1247   1265   1296
MSG13   005066R        1139#  1231   1239
MSG14   005100R        1141#  1235   1241
MSG15   005120R        1144#  1221   1228   1248   1257   1266   1275   1284   1291   1297
MSG17   005131R        1146#  1249   1258   1267   1276
MSG18   005150R        1149#  1219   1226
MSG19   005156R        1150#  1246   1255
MSG2    004667R        1111#  1232   1236
MSG20   005166R        1152#  1264   1273
MSG21   005175R        1154#  1191
MSG22   005213R        1157#  1215
MSG23   005232R        1160#  1192   1195   1198   1205   1222   1229
MSG24   005241R        1162#  1282   1289   1295
MSG25   005271R        1167#  1285
MSG26   005341R        1174#  1251   1260   1269   1278
MSG27   005343R        1175#  1299
MSG3    004701R        1113#  1213   1217   1224   1244   1253   1262   1271   1280   1287   1293
MSG4    004711R        1115#  1199   1202   1206   1209   1211   1233   1237   1242
MSG5    004731R        1118#  1197
MSG6    004745R        1121#  1201
MSG7    004767R        1125#  1204
MSG8    005006R        1128#  1240
MSG9    005031R        1132#  1194
NTRUPT  003344R         852    864#
NUAV    000570R         455    460#
NULL  = 000000          322#   594    651    734    756    1014   1053   1080
NUNT    005510R         461    736    758    1213#
NXERCT  000332R         348#   445*   1017*  1018
NXERMS  005764R        1016   1299#
OPEN  = 000000          258    264    265    266    267    284    285    286    287    288    289    290    291
                        293    295    297    298    300    301    302    305    306    308    309    311    312
                        313    314    322#
OTOA$ = 104420          322#
```

```
OUBUF2  001100R          543     569#
OUTPUT  001232R          487     542#
PASCNT  000034R          272#
PIRQ$ = 000004           322#    866
POPSP = 005726           322#
POPSP2= 022626           322#
PPTY  = 000000           322#
PRTY0 = 000000           262     322#
PRTY1 = 000040           322#
PRTY2 = 000100           322#
PRTY3 = 000140           322#
PRTY4 = 010200           322#
PRTY5 = 000240           261     322#
PRTY6 = 000300           322#
PRTY7 = 000340           322#
PS    = 177776           322#
PS%   = 177776           322#
PUSH  = 005746           322#
PUSH2 = 024646           322#
RAND$ = 104417           322#
RANNUM  000054R          290#
RBUF    005770R          304     1305#
RBUFEA  000130R          306#    626
RBUFPA  000126R          305#    639     656
RBUFSZ  000132R          307#
RBUFVA  000124R          304#    485
READ    002752R          490#
RESTRT  000666R          299     476#    522
RES1    000056R          282#
RES2    000060R          283#
RSTRT   000112R          299#
RTERDD  005652R          922     1262#
RTERSD  005674R          919     1271#
RTRYFL  000330R          347#    562     615     880*    956*
RTYN    005430R          546*    601#    953*    954     966*    1187#   1250    1259    1268    1277
RXCS    000366R          370#    464     529*    533     550*    574*    589     605*    629*    643     646     724*    746*
                         768#    774*    798*    818*    854*    877     946*    948     994*    1011*   1027    1033    1041
                        1064    1066    1083
RXDB    000370R          371#    554*    558*    578*    584*    609*    613*    633*    639*    728     750     778*    802*
                         822#    855*    856*    1000*   1008    1029
SA      000274R          333#    481*    554     609     662*    663     666*    1030
SASAV   000300R          335#    435*    481     510*    511     513*    666
SBADR   000102R          292#
SDBLDN  005750R          823     1293#
SECK    003372R          781     868     876#
SELD    002144R          475*    520*    673#    678     685
SERFL   000334R          349#    444*    622     876*    942*
SETDDE  005536R          907     1224#
SETSDE  005520R          904     1217#
SETSDN  005716R          779     1280#
SOFCNT  000042R          275#
SOFER$= 104406           322#    915     929
SOFPAS  000046R          277#
SPOINT  000032R          271#    433
SPSIZ = 000040             1#    315
SRXCS   000340R          356#    897     1027*   1028*   1032    1095
```



```
SRXES   000342R          357#    1029*   1034    1096
SRXSA   000346R          359#    1030*   1098
SRXSB   000350R          360#    364     943     967*    1099
SRXSB1  000352R          361#    1100
SRXSB2  000354R          362#    1101
SRXSB3  000356R          363#    1102
SRXTA   000344R          358#    1031*   1097
SR1     000016R          264#    962
SR2     000020R          265#
SR3     000022R          266#
SR4     000024R          267#
SSGLDN  005734R          803     1287#
START   000376R          270     432#
STAT    000026R          269#
STATUS  002234R          471     692#
SUPTRG  004262R          591*    648*    882*    972*    1010*   1027*   1050*   1077*
SVR0    000062R          284#
SVR1    000064R          285#
SVR2    000066R          286#
SVR3    000070R          287#
SVR4    000072R          288#
SVR5    000074R          289#
SVR6    000076R          290#
SYSCNT  000052R          279#
TA      000272R          332#    482*    505*    506     508*    558     613     1031
TABLE   004626R          915     929     960     979     1094#
TASAV   000276R          334#    434*    482     508     514*    515     517*
TDRP    004014R          536     951     972#
TOCNT   000312R          340#    1040*   1045*   1063*   1072*
TOMLI   000310R          339#    1039*   1047*   1048    1062*   1074*   1075
TRDNEP  005574R          1085    1239#
TRPDFD= 000022           322#
TRTO    005564R          1088    1235#
TSS     002106R          493*    662#
TSSFLG  000320R          343#    499     501*    667#
UDNCK   003246R          502*    839#
UNIT0   000254R          325#    440*    683     694     710     737*    811     831
UNIT1   000256R          326#    441*    676     702     759*    787     841
UNT0    002362R          693     720#
UNT1    002500R          701     742#
UT      000266R          330#    789*    797     807     809*    817     827     829*
UTDONE  000316R          342#    503     843*    845*
UTT     000264R          329#    442*    547     602     679*    686*    698*    706*    767     773     839     883
U0      002236R          693#
U1      002274R          695     697     721#
VARIFY  000374R          373#    778     802     822
VEC     000372R          372#
VECTOR  000010R          260#    851     869
VSET    003300R          463     851#
WASADR  000104R          294#
WBUFEA  000136R          309#    571
WBUFPA  000134R          308#    584
WBUFRQ  000140R          310#
WBUFSZ  000142R          311#
WDCNT   000302R          336#    436*    578     633
WDFR    000116R          301#
```

```
WDTO    000114R          300#
WRITF   000740R          487#
WTERDD  005600R          936    1244#
WTERSD  005630R          933    1253#
WTF     000270R          331#    542*    564*    886
XDNER   003542R          900     905     908#
XDNSET  002732R          771     777     782#
XFLAG   000005R          258#
XNXER   004260R         1009    1019    1022#
XUDNCK  003276R          844     846#
.     = 006372R          365     366     657    1184#   1186#   1188#   1306#

. ABS.  000000      000
        006372      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XRXBC0,XRXBC0/SOL/CRF:SYM=DDXCOM,XRXBC0
 RUN-TIME:  7 12 1 SECONDS
 RUN-TIME RATIO: 90/21=4.2
 CORE USED:  9K  (17 PAGES)
```

```
                .REPT   0


                                IDENTIFICATION
                                --------------

PRODUCT CODE:   AC-5234B-MC

PRODUCT NAME:   CXVSC80 VSV11 CSS DEC/X MOD

DATE CREATED:   OCT 1,1980

MAINTAINER:     CSS PRODUCTS GROUP

AUTHOR(S):      GUS PASQUANTONIO
                PAUL TAYLOR
                DICK GORDON

REVISION HISTORY:

        PAUL TAYLOR     18-AUG-81       <VERSION A1>
        DICK GORDON     20-DEC-81       <<VER B>>
```



```
ABSTRACT:
--------
        "XVSCB" IS A DEC/X11 SOFTWARE MODULE WHICH EXERCISES A
        VSV11/VS11 GRAPHIC SYSTEM.
        THE "XVSCB" DEC/X11 MODULE CONSISTS OF

        *** VSV-11 INSTRUCTION SET.
        *** SPECIAL MACROS AND OPDEFS.
        *** DEVICE REGISTER ASSIGNMENTS
        *** JOY-STICK AND TIME-OUT INTERRUPT SERVICE ROUTINES.
        *** LOOK-UP-TABLE SERVICE ROUTINES.
        *** PROGRAM START RESTART
        *** PART 1 -- THE STANDARD VISUAL REFERENCE FRAME
        *** PART 2 -- THE OLD BOUNCING BALLS
        ***     SR1 BIT0 = 1 = ENABLE PART  2.
        ***     SR1 BIT1 = 1 = USE TWO MEMORY DYNAMIC MODE.
        ***     SR1 BIT2 = 1 = DISABLE BELL ON BOUNCE.
        ***     SR2 BIT0 = 1 = ENABLE COLOR LOOKUP TABLE        <VERSION A1>
        ***     NOTE:

                THE EXERCISER WILL WORK WITH A SINGLE MEMORY CHANNEL,
        EXCEPT FOR THE COMBINATION OF SR1 BIT0=1 AND SR1 BIT1=1.THE
        VSV11/VS11 CAN HAVE UP TO FOUR SEPARATE MEMORY CHANNELS (0 THRO 3)
        THOSE ARE JUMPER SELECTABLE ON THE MODULE.TO RUN THE TEST WITH
        BOTH SR1 BIT0=1 AND SR1 BIT1=1,TWO MEMORY CHANNELS ARE NEEDED.
        REFER THE VSV TECHNICAL MANUAL (FIGURE 2-4) FOR DYNAMIC
        CONFIGURATION.

                ALSO IF THE COLOR LOOKUP IS SPECIFIED ONLY ONE VSV IS
        EXERCISED. THIS IS DONE BY SETTING BIT 0 OF SR2.
                SR1 = 0         ;TEST PATTERN ONLY
                SR1 = 1         ;DISPLAY+BOUNCING BALL+BELL
                SR1 = 2         ;TEST PATTERN ONLY
                SR1 = 3         ;TEST PATTERN+BOUNCING BALL+BELL(TWO CHANNELS)
                SR1 = 4         ;TEST PATTERN ONLY
                SR1 = 5         ;TEST PATTERN+BOUNCING BALL
                SR1 = 6         ;TEST PATTERN ONLY
                SR1 = 7         ;TEST PATTERN+BOUNCING BALL (TWO CHANNELS)
***

        *** PART 1 -- DISPLAY CODE
        *** PART 2 -- DISPLAY CODE
        *** ASCII CHARACTER ADDRESS TABLE AND SUBPIX
        *** 256 LEVEL LUMINANCE (GREY-SCALE) TABLE
        *** PATCH AREA AND END
```

```
                                REQUIREMENTS:
                                -------------
                                        PDP-11/LSI-11 PROCESSOR WITH 16K MEMORY OR MORE MEMORY
                                        CONSOLE DEVICE (LA30,LA36,VT50,VT100,ETC.)
                                        XXDP + LOAD DEVICE (RX,RP,RL,TM,DT,ETC.)
                                        1 TO 8 VSV11 SYSTEMS, EACH CONSISTS OF:

                                                M7064  IMAGE PROCESSOR
                                                M7062  IMAGE MEMORY (1 TO 4 CHANNELS)
                                                M7069  COLOR LOOK-UP RAM (OPTIONAL)
                                                VCG04  SYNC GENERATOR WITH CURSOR CONTROL
                                                DISPLAY MONITOR, COLOR OR MONOCHROME (OPTIONAL)
                                PASS DEFINATION:
                                ----------------
                                                ONE PASS OF THE "XVSCB" CONSISTS OF ONE ITERATION OF TEST
                                                SEQUENCE.



                                EXECUTION TIME:
                                ---------------
                                                ONE PASS OF "XVSCB" ALONE ON PDP11/34 TAKES ABOUT 10SEC.



                                CONFIGURATION REQUIREMENTS:
                                ---------------------------

                                                DEFAULT PARAMETERS:
                                                DEVADR:172010,VECTOR:320,BR1:4,ICONST:1.

                                        .ENDR
```



```
                                        IDMOD   <VSCB >,172010,320,4,0,0,1
000000                                  MODULE  140000,/SCB ,172010,320,4,0,0,1,
                                        .TITLE  VSCB DEC/X11 SYSTEM EXERCISER MODULE
                                ;       ODACOM  VERSION  6       23-MAY-78
                                        .LIST   BIN
                                ;*****************************************************************
000000                          BEGIN:
000000  126     123     103     MODNAM: .ASCII  /VSCB / ;MODULE NAME.
000003  102     040
000005  000                     XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF #BUFF USAGE
000006  172010                  ADDR:   172010+0                ;1ST DEVICE ADDR.
000010  000320                  VECTOR: 320+0                   ;1ST DEVICE VECTOR.
000012  200                     BR1:    .BYTE   PRTY4+0         ;1ST BR LEVEL.
000013  000                     BR2:    .BYTE   PRTY0+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  001000'                 INIT:   START                   ;MODULE START ADDR.
000032  000222'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034  000000                  PASCNT: 0                       ;PASS COUNTER.
000036  000001                  ICONT:  1                       ;# OF ITERATIONS PER PASS=1
000040  000000                  ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042  000000                  SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044  000000                  HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046  000000                  SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050  000000                  HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052  000000                  SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054  000000                  RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056                          CONFIG:
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                          #ASADR:                         ;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                  A#AS:   OPEN                    ;ACTUAL DATA.
000112  001026'                 RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114  000000                  #DTO:   OPEN                    ;#WORDS TO MEMORY PER ITERATION
000116  000000                  #DFR:   OPEN                    ;#WORDS FROM MEMORY PER ITERATION
000120  000000                  INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122                          IDNUM:                          ;MODULE IDENTIFICATION NUMBER=
        000040                          .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
```

```
                                .NLIST
                                .WORD   0
                                .LIST
                                .ENDM
     000222                     MODSP:
                                ;*****************************************************************************
  137                           ;
  138                           ; DECX MODULE FOR THE VSV-11 GRAPHICS IMAGE PROCESSER.
  139                           ;
  140                           ; NOTE THAT THIS PROGRAM REQUIRES THE MACRO PACKAGE DDXCOM.C6.
  141                           ;
  142                           .SBTTL
  143                           .SBTTL
  144                           .SBTTL  *** DECX MODULE FOR VSV-11
  145                           .SBTTL
  146                           .SBTTL
  147                           ;
  148                           ; BUILD A RUN TIME MODULE UNDER XXDP+ AS FOLLOWS:
  149                           ;
  150                           ;       .R DXCL<CR>
  151                           ;           ::
  152                           ;       BLAH-BLAH-BLAH          ; GREETINGS.
  153                           ;           ::
  154                           ;       *CNF/NP<CR>             ; CONFIGURE, NO PROMPT.
  155                           ;       *MONITOR: C<CR>         ; USE C MONITOR.
  156                           ;       *MDL VSCA<CR>           ; THIS MODULE NAME...
  157                           ;       *MDL NAM2<CR>           ;...AND ANY OTHERS.
  158                           ;           ::
  159                           ;           ::
  160                           ;       *EX<CR>                 ; LEAVE CONFIGURE MODE.
  161                           ;
  162                           ;       *LINK ODEV:NAME.BIN=IDEV:LIBNAM.LIB<CR>
  163                           ;       SYS SIZE: 1B0000<CR>
  164                           ;           ::
  165                           ;       BLAH-BLAH-BLAH          ; LINKER PROGRESS REPORTS.
  166                           ;           ::
  167                           ;       LINK DONE
  168                           ;       *
  169                           ;
  170                           ;PRGSIZ=        ^H<LASTAD>+1    ; PROGRAM SIZE IN 1/8K UNITS.
  171                           .NLIST  ME
  172                           .LIST   MEB
```



```
  174                           .SBTTL  *** VSV-11 INSTRUCTION SET.
  175                           ;
  176                           ;PLOTTING INSTRUCTIONS.
  177                           ;
  178     100000                CHAR=   100000          ;CHARACTER MODE.
  179     104000                SVEC=   104000          ;SHORT VECTOR MODE.
  180     110000                LVEC=   110000          ;LONG VECTOR MODE.
  181     114000                APNT=   114000          ;ABSOLUTE POINT MODE.
  182     120000                GHX=    120000          ;GRAPH/HISTOGRAM X MODE.
  183     124000                GHY=    124000          ;GRAPH/HISTOGRAM Y MODE.
  184     130000                RPNT=   130000          ;RELATIVE POINT MODE.
  185                           ;
  186                           ;PIXEL DATA FIELD <10:2> FOR THE ABOVE OPCODES.
  187                           ;
  188     002000                L0=     ^B10000000000           ;INTENSITY LEVEL 0 (BLACK)
  189     002000                NONE=   L0
  190     003774                ALL=    L0!<255.*4>     ; MAX IS 256. GREY SCALE LEVELS.
  191     003774                WHITE=  ALL
  192     002170                BLU=    L0!<36*4>       ; ON THE GREY-SCALE, BLUE = L36...
  193     002500                RED=    L0!<120*4>      ;...RED = L120...
  194     003140                GRN=    L0!<230*4>      ;...AND GREEN = L230.
  195                           ;
  196                           ; BIT MAP MODE DEFINITIONS.
  197                           ;
  198     136000                BM14=   136000          ; MODE 1 WITH 4 BIT PIXELS.
  199     137000                BM18=   137000          ; MODE 1 WITH 8 BIT PIXELS.
  200                                                   ; PIXEL COUNT IN <8:0>.
  201
  202     134000                BM04=   134000          ; MODE 0 WITH 4 BIT PIXELS.
  203     135000                BM08=   135000          ; MODE 0 WITH 8 BIT PIXELS.
  204     000000                M32=    0               ;  32 SQUARE PIXEL ARRAY.
  205     000001                M64=    1               ;  64 SQUARE.
  206     000002                M128=   2               ; 128 SQUARE.
  207     000003                M256=   3               ; 256 SQUARE.
  208     000010                EX2=    10              ; EXPAND BY 2, NO SMOOTHING.
  209     000020                EX4=    20              ; EXPAND BY 4, NO SMOOTHING.
  210     000050                EXSM2=  50              ; EXPAND AND SMOOTH BY 2.
  211     000060                EXSM4=  60              ; EXPAND AND SMOOTH BY 4.
  212     000100                R256=   100             ; 256 X 256 RESOLUTION.
```

```
                        ;
                        ;CONTROL AND STATUS INSTRUCTIONS.
                        ;
146040                  CUOFF=  146040          ;CURSOR (JOY-STICK) VIDEO OFF.
146060                  CUON=   146060          ;CURSOR VIDEO ON.
146100                  CURD=   146100          ;READ JSX,JSY => DXR,DYR.
146016                  CUIM=   146016          ;INTERRUPT ON MATCH (SWITCH DISABLED).
146013                  CUIS=   146013          ;INTERRUPT ON SWITCH (MATCH DISABLED).
146012                  CUIOFF= 146012          ;INTERRUPTS (BOTH) OFF.

150000                  SETHB=  150000          ;SET GRAPH/HISTO BASE LINE.
152000                  SETCB=  152000          ;SET CHARACTER BASE ADDRESS.

160000                  DJMP=   160000          ;DISPLAY JUMP.
164000                  DNOP=   164000          ;DISPLAY NO-OP.
165000                  DPOP=   165000          ;DPOP = RTS FROM CHARACTER SUB-PIX.
164001                  SYNC=   DNOP+1          ;PROCEED IN SYNC = DNOP + N <8:0>.
                                                ;WAITS FOR N OCCURRENCES OF VERTICAL...
                                                ;...SYNC START, THEN PROCEEDS...
                                                ;...(16.6MS @ 60HZ OR 20MS @ 50HZ).

000010                  ADDBAS= 10              ; DIFFERENCE BETWEEN VSV11 BASE ADDRESSES <VERSION A1>
000020                  VECBAS= 20              ; DIFFERENCE BETWEEN VECTORS            <VERSION A1>
170200                  SWTCH= 170200           ;SWITCH READ/WRITE STATUS.
170140                  CLRMEM= 170140          ;CLEAR IMAGE MEMORY TO 0'S.
170100                  SETMEM= 170100          ;SET IMAGE MEMORY TO CURRENT PIX DATA.

172000                  STOP=   172000          ;DISPLAY STOP.
171000                  SIOFF=  171000          ;STOP INTERRUPT DISABLE.
171400                  SION=   171400          ;STOP INTERRUPT ENABLE.
173000                  STOPN=  STOP!SIOFF      ;STOP, DO NOT INTERRUPT.
173400                  STOPI=  STOP!SION       ;STOP AND INTERRUPT.

174100                  GXI=    174100          ;SET X INCREMENT <5:0> FOR GRAPH/HST Y.
174100                  GYI=    GXI             ;SET Y INCREMENT <5:0> FOR GRAPH/HST X.

176000                  PROTEC= 176000          ;PROTECT MEMORY (OFF).
176050                  READ=   176040!SWE      ;READ MEM (DISPLAY), SWITCH ENABLED.
176034                  WRT=    176024!SWE      ;WRITE MEM (1'S AND 0'S), SWITCH ENABLED
176036                  WRT1=   176026!SWE      ;WRITE MEM (1'S ONLY), SWITCH ENABLED.
000010                  SWE=    10              ; ENABLE SWITCH READ/WRITE.
176074                  RDWRT=  READ!WRT        ;READ, WRITE ALL DURING RETRACE.
176076                  RDWRT1= READ!WRT1       ;READ, WRITE 1'S DURING RETRACE.

000000                  CH0=    0               ;CHANNEL SELECT BITS...
000400                  CH1=    400             ;...FOR MEMORY CONTROL...
001000                  CH2=    1000            ;...INSTRUCTIONS (176XXX).
001400                  CH3=    1400

000000                  JS0=    CH0             ;JOY-STICK UNIT SELECT BITS...
000400                  JS1=    CH1             ;...FOR CURSOR CONTROL...
001000                  JS2=    CH2             ;...INSTRUCTIONS (146XXX).
001400                  JS3=    CH3
```



```
                        ;
                        ; COLOR LOOK-UP-RAM EQUATES
                        ;
100000                  LDUN=   100000          ; CSR REGISTER -- FUNCTION DONE BIT.
010000                  LREAD=  ^B1000000000000         ; READ DATA FROM ADDRESS <7:0>.
002000                  LDI=    ^B10000000000           ; INTERRUPT ON READ/WRITE DONE.

010000                  LSET=   ^B1000000000000         ; DATA REGISTER -- SET ALL ADDRESSES...
                                                ;...WITH DATA IN <11:0>...
                                                ;...BLUE<11:8>, GREEN<7:4>, RED<3:0>.

000100                  GCOFF=  ^B1000000               ; MAINTENANCE REGISTER...
                                                ;...GAMMA CORRECTION OFF.
                        ;
                        ;OTHER USEFUL DEFINITIONS.
                        ;
040000                  I=      ^B1000000000000000              ;INTENSIFY VECTOR OR POINT.
020000                  MSX=    ^B100000000000000               ;MINUS SIGNS FOR...
000100                  MSY=    ^B1000000               ;...SHORT FORM VECTOR DATA.
020000                  MXY=    ^B100000000000000               ;MINUS SIGN FOR LONG FORM DATA.
040000                  HST=    ^B1000000000000000              ;USE GHX/GHY DATA AS HISTOGRAM.

001776                  MAXX=   511.*2          ;MAXIMUM X ADDRESS (512 X 512 X 2).
001776                  MAXY=   MAXX            ;SAME FOR Y.
000776                  HAFX=   <MAXX/2>-1      ;HALF MAX X (CENTER SCREEN).
000776                  HAFY=   HAFX            ;SAME FOR Y.

001776                  MAXY50= 511.*2          ;MAX VISIBLE Y ON 50HZ SYSTEM.
000776                  HAFY50= <MAXY50/2>-1    ;HALF VISIBLE Y.

001676                  MAXY60= <511.-32.>*2    ;MAX VISIBLE Y ON 60HZ SYSTEM.
000736                  HAFY60= <MAXY60/2>-1    ;HALF VISIBLE Y.

000002                  DX=     2               ;DELTA X (1 PIXEL UNIT).
000002                  DYI=    DX              ;DELTA Y, INTERLACED MODE.
000004                  DYNI=   DX*2            ;DELTA Y, NON-INTERLACED MODE.

000014                  SETRR= 0+14             ; SET RELOCATE REGISTER WITH <15:4>...
                                                ;...DSR<15:4> BECOMES RR<17:6>...
                                                ;...VSV RELOCATES ON 32. WORD BOUNDARY.
```

```
                                .SBTTL  *** SPECIAL MACROS AND OPDEFS.
                                ;
                                ; MACROS FOR ASSEMBLING VECTOR DATA.
                                ; CALLING ARGUMENTS ARE:
                                ;       A = INTENSIFY BIT, "I" OR "U".
                                ;       B = DELTA X, +/-0 THRU 1776 (76 IF SHORT TYPE).
                                ;       C = DELTA Y,            DITTO
                                ;
                                        .MACRO  SXY A,B,C       ;ASSEMBLE SHORT (1 WORD) DATA.
                                        $I = ^B100000000000000
                                        $X = <B*200>
                                        $Y = C
                                        .IIF IDN <U> <A>, $I = 0
                                        .IIF LT B, $X = <-B*200>!<^B100000000000000>
                                        .IIF LT C, $Y = <-C>!<^B1000000>
                                        .WORD   $I+$X+$Y
                                        .ENDM   SXY

                                        .MACRO  LXY A,B,C       ;ASSEMBLE LONG (2 WORDS) DATA.
                                        $I = ^B100000000000000
                                        $X = B
                                        $Y = C
                                        .IIF IDN <U> <A>, $I = 0
                                        .IIF LT B, $X = <-B>!<^B100000000000000>
                                        .IIF LT C, $Y = <-C>!<^B100000000000000>
                                        .WORD   $I+$X, $Y
                                        .ENDM   LXY
                                ;
                                ;SOME HANDY -11 OPDEF'S.
                                ;
000401                          SKP1=   BR+1                    ;SKIP NEXT WORD.
000402                          SKP2=   BR+2                    ;SKIP NEXT 2 WORDS.
000403                          SKP3=   BR+3                    ;SKIP NEXT 3 WORDS.
```



```
                                .SBTTL  *** DEVICE REGISTER ASSIGNMENTS
                                ;
                                ; DEVICE REGISTER ASSIGNMENTS.
                                ;
000222  172010                 DPC:    172010          ;DISPLAY PC
000224                          DRR:
000224  172012                 DSR:    172012          ;DISPLAY STATUS REGISTER (ALSO REG REG).
000226  172014                 DXR:    172014          ;DISPLAY X POSITION REGISTER.
000230  172016                 DYR:    172016          ;DISPLAY Y POSITION REGISTER.

000232  172020                 LSR:    172020          ; LOOK-UP-RAM CSR.
000234  172022                 LDR:    172022          ; LOOK-UP-RAM DATA REGISTER.
000236  172024                 LMR:    172024          ; LOOK-UP-RAM MAINTENANCE REGISTER.

000240  000320                 STPV:   320             ;STOP VECTOR
000242  000324                 JSMV:   324             ;JOY-STICK MATCH VECTOR.
000244  000330                 TOTV:   330             ;TIME-OUT VECTOR
000246  000334                 JSSV:   334             ;JOY-STICK SWITCH (DEMAND) VECTOR.
000250  000340                 LUTV:   340             ; LUT READ/WRITE DONE VECTOR.
                                ; MISCELLANEOUS STORAGE REGISTERS.
                                ;
000252  000000                 SDPC:   0               ; SAVED DPC ON INTERRUPT.
000254  000000                 SDSR:   0               ;       DITTO DSR
000256  000000                 SDXR:   0               ;       DITTO DXR
000260  000000                 SDYR:   0               ;       DITTO DYR

000262  000000                 LUTFLG: 0               ; +1 IF LUT (A025) INSTALLED.

000264  000000                 SLSR:   0               ; SAVED LUT STATUS.
000266  000000                 SLDR:   0               ;       DITTO DATA.
000270  000000                 SLMR:   0               ;       DITTO MAINT.

000272  000000                 FIDX:   0               ; FRAME INDEX COUNTER (FOR PART 1).
000274  000010                 FCD:    8.              ; FRAME CHANGE DELAY COUNT...           <VERSION A1>
000276  000010                 FCD1:   8.              ;...IS DECREMENTED HERE.                <VERSION A1>
000300  000000                 ADDRX:  0               ; CURRENT BASE ADDRESS                  <VERSION A1>
000302  000000                 VECIXX: 0               ; CURRENT VECTOR ADDRESS                <VERSION A1>
000304  000000                 DEVCN:  0               ; CURRENT DEVICE CHECK MASK             <VERSION A1>
000306  000000                 DIVCHK: 0               ; USER SPECIFIED DEVICE MASK            <VERSION A1>

000310  000000                 T0:     0               ; SOME TEMP REGISTERS.
000312  000000                 T1:     0
000314  000000                 T2:     0
000316  000000                 T3:     0

000320  003726'                VADDR:  DBEGIN          ; DISPLAY CODE VIRTUAL ADDRESS.
000322  000000                 PADDR:  0               ; CONVERTED TO PHYSICAL HERE...
000324  000000                 EABITS: 0               ;...WITH EA BITS HERE <5:4>.
000326  000000                 DSTRT:  0               ; DISPLAY START (CLEAR) DISPATCHER.
000330  000000                 DLOOP:  0               ; DISPLAY LOOPING DISPATCHER.
000332  000014                 RELOC:  SETRR           ; PLUS RELOCATION BITS IN <15:4>.
000334  000000                 RELOK:  0               ; NON-ZERO (CAN'T RUN) IF RELOCATION...
                                                       ;...BOUNDARY IS NOT A MULTIPLE...
                                                       ;...OF 100B (32. WORDS).
000336  000000                 BADPAD: 0               ; PHYSICAL ADDRESS WHICH CAUSED...
                                                       ;..."RELOK" TO BE NON-ZERO.
```

```
400 000340  000000                  TEMPS:  0               ;STORE TEMP PS
401 000342  000000                  TEMPC:  0               ;STORE TEMP PC
402         000004                  MEMERR=4                ;NON EXIST MEM TRAP
```



```
404                                 .SBTTL  *** JOY-STICK AND TIME-OUT INTERRUPT SERVICE ROUTINES.
405                                 ;
406                                 ; JOY-STICK INTERRUPT SERVICE.
407                                 ;
408 000344                          MATCH:
409 000344                          SWITCH:
    000344  000004  000000' 000352'        PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
410 000352  004767  000074          1$:    JSR     PC,SAVDPU       ; SAVE REGISTERS...
411 000356  000240  000240                 240,240                 ;...AND THATS ALL (FOR NOW).
412                                        ;
413                                        ;
414                                        ;
415                                        ;
416 000362  000167  001424                 JMP     RESM            ; RESUME AND EXIT.
417                                 ;
418                                 ; DISPLAY TIME-OUT MEANS TROUBLE -- QUIT.
419                                 ;
420 000366                          TIMOUT:
    000366  000004  000000' 000374'        PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
421 000374  004767  000052          1$:    JSR     PC,SAVDPU       ; SAVE REGISTERS.
422 000400  104401  000000' 000420'        MSG$,BEGIN,2$   ;ASCII MESSAGE CALL
423 000406  046767  177672  177702        BIC     DEVON,F3        ; CLR THIS ONE FROM MASK        <<VER B>>
424 000414  000167  000406                 JMP     RESINT          ; GET THE NEXT ONE
425
426 000420      045     040     040 2$:    .ASCIZ  '%  VSCA0 -- DPU TIME-OUT%'
    000423      126     123     103
    000426      101     060     040
    000431      055     055     040
    000434      104     120     125
    000437      040     124     111
    000442      115     105     055
    000445      117     125     124
    000450      045     000
427                                        .EVEN
428                                 ;
429                                 ; COMMON ROUTINE TO SAVE DPU REGISTERS.
430                                 ;
431 000452  017767  177544  177572  SAVDPU: MOV     @#DPC,SDPC      ; SAVE ALL DPU REGISTERS.
432 000460  017767  177540  177566          MOV     @DSR,SDSR
433 000466  017767  177534  177562          MOV     @DXR,SDXR
434 000474  017767  177530  177556          MOV     @DYR,SDYR
435 000502  000240                          240
436 000504  000240                          240             ; AND ANYTHING ELSE WE MIGHT NEED.
437 000506  000240                          240
438 000510  000207                          RTS     PC
```

```
                                        .SBTTL  *** LOOK-UP-TABLE SERVICE ROUTINES.
                                        ;
                                        ; NORMALLY, WE DON'T EXPECT ANY LUT INTERRUPTS.
                                        ; REPORT AND RESTART IF WE GET ONE.
                                        ;
000512                                  UNXLI:
000512  000004  000000' 000520'             PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
000520  017767  177506  177536  1$:       MOV     @LSR,SLSR       ; SAVE LUT REGISTERS.
000526  017767  177502  177532            MOV     @LDR,SLDR
000534  017767  177476  177526            MOV     @LMR,SLMR
000542  000240  000240                     240,240                 ; IN CASE WE NEED SOMETHING ELSE
000546  104401  000000' 000560'             MSG$,BEGIN,2$   ;ASCII MESSAGE CALL
000554  000167  000246                     JMP     RESTRT          ;...AND RESTART.

000560      045     040     040  2$:       .ASCIZ  '% VSCA0 -- UNEXPECTED LUT INTERRUPT%'
000563      126     123     103
000566      101     060     040
000571      055     055     040
000574      125     116     105
000577      130     120     105
000602      103     124     105
000605      104     040     114
000610      125     124     040
000613      111     116     124
000616      105     122     122
000621      125     120     124
000624      045     000
                                        .EVEN
                                        ;
                                        ; BLAST A 256 LEVEL GREY-SCALE FROM THE DATA IN "LUMTBL".
                                        ;
000626  012777  000656' 177414  GRASCL: MOV     #2$,@LUTV       ; SET DONE VECTOR.
000634  012777  002000  177370          MOV     #LD1!0,@LSR     ; INTERRUPT AND ADDR 0.
000642  012700  015112'                 MOV     #LUMTBL,R0      ; TABLE ADDRESS.
000646  012077  177362          1$:     MOV     (R0)+,@LDR      ; FILL THE LUT RAM.
000652  104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
000656                                  2$:
000656  000004  000000' 000664'             PIRQ$,BEGIN,3$          ; QUEUE UP TO CONTINUE AT 3$ AND RTI
000664  105777  177342          3$:     TSTB    @LSR            ; LUT RAM FULL ??
000670  001366                          BNE     1$              ; NO, LOOP.
000672  000431                          BR      BLXIT           ; YES, RETURN.
                                        ;
                                        ; BLAST 16 SHADES OF THE BASIC COLOR IN (R0).
                                        ;
000674  012777  000732' 177346  SHADES: MOV     #3$,@LUTV       ; SET WRITE DONE VECTOR.
000702  012777  002000  177322          MOV     #LD1!0,@LSR     ; ADDR 0, INTERRUPT WHEN DONE.
000710  042700  177356                  BIC     #^C421,R0       ; SET DIMMEST SHADE THIS COLOR.
000714  005001                          CLR     R1              ; 1ST SHADE IS ALWAYS BLACK.
000716  012702  000020          1$:     MOV     #16.,R2         ; 16 WORDS / SHADE.
000722  010177  177306          2$:     MOV     R1,@LDR         ; WRITE A WORD.
000726  104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
000732                                  3$:
000732  000004  000000' 000740'             PIRQ$,BEGIN,4$          ; QUEUE UP TO CONTINUE AT 4$ AND RTI
000740  105777  177266          4$:     TSTB    @LSR            ; LUT RAM FULL ??
000744  001404                          BEQ     5$              ; EXIT IF SO.
000746  005302                          DEC     R2              ; THIS SHADE DONE ??
000750  001364                          BNE     2$              ; NOT YET, LOOP.
```



```
000752  060001                          ADD     R0,R1           ; YES, INCREMENT THE SHADE...
000754  000760                          BR      1$              ;...AND LOOP.
000756                                  5$:
                                        ;
                                        ; COMMON EXIT FROM EITHER PATTERN.
                                        ;
000756  005077  177250          BLXIT:  CLR     @LSR            ; CLEAR STATUS...
000762  012777  000100  177246          MOV     #GCOFF,@LMR     ;...GAMMA CORRECT OFF...
000770  012777  000512' 177252          MOV     #UNXLI,@LUTV    ;...RESET VECTOR...
000776  000207                          RTS     PC              ;...AND RETURN.
```

```
                                        .SBTTL  *** PROGRAM START-RESTART
                                        ;
                                        ; INITIALIZE DISPLAY ADDRESSES, AND VECTORS.
                                        ; BLAST THE LUT (IF IT'S THERE).
                                        ;
001000                                  START:
001000  032767  000001  177012                  BIT     #1,SR2          ; USING COLOUR LOOKUP                   <VERSION A1>
001006  001404                                  BEQ     1$              ; NO                                    <VERSION A1>
001010  012767  000001  177270                  MOV     #1,DIVCHK       ; YES- FORCE ONE DEVICE                 <VERSION A1>
001016  000411                                  BR      ONE             ;                                       <VERSION A1>
001020  016767  176770  177270  1$:     MOV     DVID1,T3        ; GET MASK AND SAVE IT                  <<VER B>>
                                        ;
001026                                  RESTRT:
001026  016767  177264  177252                  MOV     T3,DIVCHK       ; PUT MASK INTO WORK LOCATION           <VERSION A1>
001034  016767  177256  177252                  MOV     T3,T2           ; ALSO INTO T2 COUNTER                  <<VER B>>
001042                                  ONE:                            ;
001042  012767  000001  177234                  MOV     #1,DEVON        ; START CHECK AT DEVICE 1               <VERSION A1>
001050  016767  176732  177222                  MOV     ADDR,ADDRX      ; SET DEVICE START ADDRESS              <VERSION A1>
001056  016767  176726  177216                  MOV     VECTOR,VECTXX   ; AND VECTOR START                      <VERSION A1>
                                        ;
001064  005767  177216          RESTRA: TST     DIVCHK          ; ANY DEVICE TO TEST                    <VERSION A1>
001070  001002                                  BNE     77$             ; YES                                   <VERSION A1>
001072  104410  000000'                        ENDS,BEGIN              ;
001076  005767  177202          77$:    TST     DEVON           ; ALL MODULES TESTED                    <VERSION A1>
001102  001736                                  BEQ     START           ; YES RESTART FROM THE TOP              <VERSION A1>
001104  036767  177174  177174                  BIT     DEVON,DIVCHK    ; IS THIS MODULE ON                     <VERSION A1>
001112  001012                                  BNE     76$             ; YES USE IT                            <VERSION A1>
001114  062767  000010  177156                  ADD     #ADDRAS,ADDRX   ; ELSE STEP TO NEXT ONE                 <VERSION A1>
001122  062767  000020  177152                  ADD     #VECBAS,VECTXX  ; AND VECTOR                            <VERSION A1>
001130  006367  177150                          ASL     DEVON           ;                                       <VERSION A1>
001134  001721                                  BEQ     START           ; IF ALL DONE RESET TO TOP              <VERSION A1>
001136  000757                                  BR      77$             ;                                       <VERSION A1>
001140                                  76$:                            ;
001140  016700  177134                          MOV     ADDRX,R0        ; GET BASE DEVICE ADDRESS.              <VERSION A1>
001144  012701  000222'                        MOV     #DPC,R1         ;
001150  010021                          1$:     MOV     R0,(R1)+        ; SET REGISTER ADDRESSES.
001152  062700  000002                          ADD     #2,R0           ;
001156  020127  000240'                        CMP     R1,#STPV        ; END OF ADDRESS BLOCK ??
001162  001372                                  BNE     1$              ; BR IF NOT.
                                        ;
001164  016700  177112                          MOV     VECTXX,R0       ; YES, SET VECTOR ADDRESSES.            <VERSION A1>
001170  010021                          2$:     MOV     R0,(R1)+        ;
001172  062700  000004                          ADD     #4,R0           ;
001176  020127  000252'                        CMP     R1,#LUTV+2      ;
001202  001372                                  BNE     2$              ; NO
                                        ;
001204  016700  177030                          MOV     STPV,R0         ; YES, SET DPU VECTORS...
001210  116701  176576                          MOVB    BR1,R1          ;...AND PRIORITIES.
001214  042701  177437                          BIC     #^C340,R1       ;
001220  012720  002022'                        MOV     #STOP1,(R0)+    ; SET STOP...
001224  010120                                  MOV     R1,(R0)+        ;
001226  012720  000344'                        MOV     #MATCH,(R0)+    ;...MATCH...
001232  010120                                  MOV     R1,(R0)+        ;
001234  012720  000366'                        MOV     #TIMOUT,(R0)+   ;...TIME-OUT...
001240  010120                                  MOV     R1,(R0)+        ;
001242  012720  000344'                        MOV     #SWITCH,(R0)+   ;...AND SWITCH VECTORS.
001246  010120                                  MOV     R1,(R0)+        ;
```



```
001250  012720  000512'                        MOV     #UNXLI,(R0)+    ; SET LUT DONE VECTOR.
001254  010120                                  MOV     R1,(R0)+        ;
                                        ;
001256  005227                          LUTIN:  INC     (PC)+           ; AUTO-SIZE FOR LUT INSTALLED.
001260  177777                                  -1                      ; ONCE-ONLY SWITCH.
001262  001040                                  BNE     1$
001264  013767  000004  177050                  MOV     @#MEMERR,TEMPC  ;STORE ADDRESS AT NON EXISTANCE
                                                                        ;MEMORY TRAP (TEMPARARY)
001272  013767  000006  177040                  MOV     @#MEMERR+2,TEMPS        ;STORE PS
001300  012737  003362' 000004                  MOV     #TRP1,@#MEMERR  ;NON EXIST MEM TRAP HERE
001306  013737  177776  000006                  MOV     @#PS,   @#MEMERR+2      ;GET CURRENT PS
001314  042737  177437  000006                  BIC     #177437,@#MEMERR+2      ;CLEAR UNWANTED BITS
001322  005067  176734                          CLR     LUTFLG          ; ASSUME NO LUT INSTALLED.
001326  032767  000001  176464                  BIT     #1,SR2          ; USING COLOUR LOOKUP                   <VERSION A1>
001334  001405                                  BEQ     3$              ;                                       <VERSION A1>
001336  005777  176670                          TST     @LSR            ; TRAP AND RESTART IF NO LUT.
001342  012767  000001  176712                  MOV     #1,LUTFLG       ; OTHERWISE, SET LUT AVAILABLE.
001350                                  3$:                             ;                                       <VERSION A1>
001350  016737  176766  000004                  MOV     TEMPC,@#MEMERR  ;RECOVER ERRVEC PC
001356  016737  176756  000006                  MOV     TEMPS,@#MEMERR+2        ;
                                        ;
001364                                  1$:
                                        ;
```

```
                                        ;
                                        ; NOW COMPUTE ANY RELOCATION OFFSET BASED ON:
                                        ;       OFFSET = PHYSICAL - VIRTUAL
                                        ;
                                        ; VSV RELOCATES ON 32. WORD (100 BYTES OCTAL) BOUNDARIES,
                                        ; IN THE TRADITIONAL KT STYLE.
                                        ; DURING PROGRAM RELOCATION (IF ENABLED), PA<5:0> AND VA<5:0>
                                        ; SHOULD BE IDENTICAL.  IF NOT, THE RELOCATION PROCESS DIDN'T
                                        ; WORK RIGHT.  TELL THE MAN, BUT DON'T TRY TO RUN ANYTHING.
001364                                  SETUP:
001364 012767  004154' 010730                   MOV     #SUB.1,ACAT+2   ; SET SUBROUTINE CALLS...
001372 012767  005132' 010724                   MOV     #SUB.2,ACAT+4   ;...IN VIRTUAL CHAR...
001400 012767  012270' 010720                   MOV     #SUB.3,ACAT+6   ;...ADDRESS TABLE.
001406 012767  003726' 176704                   MOV     #DBEGIN,VADDR   ; CONVERT VIRTUAL TO PHYSICAL.
001414 104415  000000' 000320'                   GETPAS,BEGIN, VADDR             ;GET PHYSICAL ADDRESS FROM 16-BIT VADDR
001422 016701  176674                           MOV     PADDR,R1
001426 042701  000077                           BIC     #77,R1          ; PA<15:6> => R1.
001432 016700  176666                           MOV     BABITS,R0       ; PA<17:16> => R0<5:4>.
001436 000300                                   SWAB    R0
001440 006100                                    ROL    R0
001442 006100                                     ROL   R0              ; PA<17:16> => R0<15:14>.
001444 006001                                   ROR     R1
001446 006001                                    ROR    R1
001450 050001                                     BIS   R0,R1           ; PA<17:6> => R1<15:4>.
001452 016700  176642                           MOV     VADDR,R0
001456 042700  000077                           BIC     #77,R0          ; VA<15:6> => R0.
001462 000241                                   CLC
001464 006000                                    ROR    R0
001466 006000                                     ROR   R0              ; VA<15:6> => R0<13:4>.
001470 160001                                   SUB     R0,R1           ; PHY - VIRT = RELOC.
001472 052701  000014                           BIS     #SETRR,R1       ; MAKE SETRR INSTRUCTION...
001476 010167  176630                           MOV     R1,RELOC        ;...FOR DSR (RR REGISTER)...
001502 016767  176612  176616                   MOV     VADDR,DSTRT     ;...AND VIRTUAL START FOR DPC.

001510 005067  176620                           CLR     RELOK           ; NOW, ASSUME RELOC IS VALID.
001514 016700  176602                           MOV     PADDR,R0
001520 166700  176574                           SUB     VADDR,R0
001524 042700  177700                           BIC     #^C77,R0        ; PA<5:0> = VA<5:0> ???
001530 001414                                   BEQ     1$              ; WE'RE OK IF SO.
001532 005167  17657b                           COM     RELOK           ; NO, INHIBIT START-UP.
001536 026767  176560  176572                   CMP     PADDR,BADPAD    ; SAME PA AS LAST TIME ??
001544 001406                                   BEQ     1$              ; YES, DON'T REPORT AGAIN.
001546 016767  176550  176562                   MOV     PADDR,BADPAD    ; NO, SAVE THE BAD NUMBER...
001554 104401  000000' 001564'                   MSGS,BEGIN,CANT ;ASCII MESSAGE CALL
001562 000423                           1$:     BR      PART1           ;...AND FALL THRU.
                                                                        ; "RELOK" NON-ZERO WILL CAUSE...
                                                                        ;...US TO "ENDIT".  SOONER OR...
                                                                        ;...LATER, WE'LL RESTART AND...
                                                                        ;...GET A NEW RELOCATION VALUE.

001564      045     040     040 CANT:   .ASCIZ  '%  VSCA0 -- BAD RELOCATION BOUNDARY%'
001567      126     123     103
001572      101     060     040
001575      055     055     040
001600      102     101     104
001603      040     122     105
001606      114     117     103
```



```
001611      101     124     111
001614      117     116     040
001617      102     117     125
001622      116     104     101
001625      122     131     045
001630      000
                                        .EVEN
```

```
                                        .SBTTL  *** PART 1 -- THE STANDARD VISUAL REFERENCE FRAME
                                        ;
                                        ; CLEAR SCREEN, BLAST LUT (OPTIONAL) AND START DISPLAY.
                                        ;
001632  005767  176476          PART1:  TST     RELOC
001636  001402                          BEQ     1$              ; *** NOP TO DEBUG ***
001640  000167  000620                  JMP     DONE2           ; RELOC INVALID, CAN'T RUN.
                                        
001644  013767  000004  176470  1$:     MOV     @#MEMERR,TEMPC  ;STORE ADDRESS AT NON EXISTANCE
                                                                ;MEMORY TRAP (TEMPARARY)
001652  013767  000006  176460          MOV     @#MEMERR+2,TEMPS        ;STORE PS
001660  012737  003404' 000004          MOV     #TRP2,@#MEMERR  ;NON EXIST MEM TRAP HERE
001666  016777  176440  176330          MOV     RELOC,@DSR      ; SET RR REGISTER.
001674  016777  176426  176320          MOV     DSTRT,@DPC      ; CLEAR SCREEN RAMS.
001702                          2$:
        001702  104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
        001706  104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001712  005777  176306                  TST     @DSR
001716  100371                          BPL     2$
001720  005767  176336                  TST     LUTFLG
001724  001402                          BEQ     3$
001726  004767  176674                  JSR     PC,GRASCL       ; BLAST THE LUT.
001732  016737  176404  000004  3$:     MOV     TEMPC,@#MEMERR  ;RECOVER ERRVEC PC
001740  016737  176374  000006          MOV     TEMPS,@#MEMERR+2        ;RECOVER PS
001746  012777  002022' 176264          MOV     #STOP1,@STPV    ; SET DPU STOP VECTOR.
001754  016767  176346  176346          MOV     DSTRT,DLOOP
001762  062767  000020  176340          ADD     #LOGO-DBEGIN,DLOOP ; SET 1ST FRAME ADDRESS.
001770  005067  176276                  CLR     FIDX            ; INIT FRAME INDEX.
                                        
001774  016767  176274  176274  NEXT:   MOV     FCD,FCD1        ; SET/RESET FRAME CHANGE DELAY.
002002  016777  176322  176212          MOV     DLOOP,@DPC      ; START/RESTART DPU...
002010  000402                          SKP2                    ;...OR...
002012  005277  176204          RESM:   INC     @DPC            ;...RESUME.
002016  104400  000000'                 EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
                                        ;
                                        ; RETURN HERE ON STOP INTERRUPT.
                                        ; CHANGE FRAMES AFTER A SUITABLE DELAY.
                                        ;
002022                          STOP1:
        002022  000004  000000' 002030'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
002030  004767  176416          1$:     JSR     PC,SAVDPU       ; SAVE DPU REGISTERS.
002034  005367  176236                  DEC     FCD1            ; CHANGE DELAY EXPIRED ??
002040  001360                          BNE     NEXT+6          ; NOT YET, LOOP ON THIS FRAME.
002042  062767  000002  176222          ADD     #2,FIDX         ; YES, BUMP INDEX...
002050  016700  176216                  MOV     FIDX,R0
002054  000170  002060'                 JMP     @2$(R0)         ;...AND CHANGE FRAMES.
002060  001632'                 2$:     PART1                   ; FRAME 1 (CLEAR AND LOGO).
002062  002076'                         FR2                     ;       2 (OCTAGONS).
002064  002106'                         FR3                     ;       3 (GRAPH/HISTO Y).
002066  002116'                         FR4                     ;       4 (GRAPH/HISTO X).
002070  002130'                         FR5                     ;       5 (BIT MAPS).
002072  002224'                         FR6                     ;       6 (COLOR BARS).
002074  002354'                         HUES                    ; CHANGE COLORS (OPTIONAL).
```



```
                                        ;
                                        ; SET UP SEQUENCE FOR EACH FRAME.
                                        ;
002076  062767  000110  176224  FR2:    ADD     #OCTGNS-LOGO,DLOOP ; ADJUST TO FRAME 2.
002104  000403                          SKP3
002106  062767  000140  176214  FR3:    ADD     #GRY-OCTGNS,DLOOP ; ADJUST TO FRAME 3.
002114  000403                          SKP3
002116  062767  000310  176204  FR4:    ADD     #GRX-GRY,DLOOP  ; ADJUST TO FRAME 4.
002124  000167  177644                  JMP     NEXT
                                        
002130  012700  005132'          FR5:    MOV     #OBUFR,R0       ; FRAME 5 REQUIRES THE BUFFER.
002134  012701  000400                  MOV     #256.,R1        ; SET UP TO FILL IT.
002140  012702  177777                  MOV     #-1,R2
002144  000407                          BR      2$
002146  032701  000077          1$:     BIT     #77,R1          ; BREAK EVERY 64 TIMES...
002152  001004                          BNE     2$              ;...SO WE DON'T HOG THE CPU.
002154  104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
        002160  104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002164  010220                  2$:     MOV     R2,(R0)+        ; FILL BIT MAP BUFFER WITH...
002166  005020                          CLR     (R0)+           ;...ALTERNATE 1'S AND 0'S.
002170  005301                          DEC     R1
002172  001365                          BNE     1$              ; LOOP 'TIL DONE.
002174  016767  176074  176074          MOV     FCD,FCD1        ; ADJUST FRAME DELAY...
002202  006267  176070                  ASR     FCD1            ;...'CAUSE BIT MAPS...
002206  006267  176064                  ASR     FCD1            ;...TAKE TOO LONG !!
002212  062767  000310  176110          ADD     #814-GRX,DLOOP  ; ADJUST ADDRESS TO FRAME 5.
002220  000167  177556                  JMP     NEXT+6
                                        
002224  012700  005132'          FR6:    MOV     #OBUFR,R0       ; FRAME 6 REQUIRES THE BUFFER.
002230  012701  112000                  MOV     #LVEC!LO,R1
002234  012702  000500                  MOV     #500,R2         ; +500 Y VALUE.
002240  012703  020500                  MOV     #-#XY!500,R3    ; -500 Y VALUE.
002244  000407                          BR      2$
002246  032701  000077          1$:     BIT     #77,R1          ; BREAK ONCE IN A WHILE...
002252  001004                          BNE     2$              ;...LIKE BEFORE.
002254  104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
        002260  104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002264  010120                  2$:     MOV     R1,(R0)+        ; SET VECTOR!L...
002266  012720  040000                  MOV     #I!0,(R0)+      ;...X...
002272  010220                          MOV     R2,(R0)+        ;...+Y.
002274  012720  000002                  MOV     #2,(R0)+
002300  005020                          CLR     (R0)+           ; MOVE OVER 1...
002302  062701  000004                  ADD     #4,R1           ;...AND INCREMENT THE LEVEL.
002306  010120                          MOV     R1,(R0)+        ; SET VECTOR!L+4
002310  012720  040000                  MOV     #I!0,(R0)+      ;...X...
002314  010320                          MOV     R3,(R0)+        ;...-Y...
002316  012720  000002                  MOV     #2,(R0)+
002322  005020                          CLR     (R0)+           ; MOVE OVER...
002324  062701  000004                  ADD     #4,R1           ;...AND INCREMENT LEVEL AGAIN.
002330  032701  001777                  BIT     #1777,R1
002334  001344                          BNE     1$              ; LOOP 'TIL ALL LEVELS SET.
002336  012720  165000                  MOV     #DPOP,(R0)+     ; THEN, TERMINATE THE BUFFER.
002342  062767  000060  175760          ADD     #BARS-814,DLOOP ; ADJUST TO FRAME 6 ADDRESS.
002350  000167  177420                  JMP     NEXT            ;...AND DO IT.
```

```
                                        ;
                                        ; NOW IF LUT IS THERE, REBLAST IT TO DISPLAY
                                        ; 16 SHADES OF EACH OF THE NTSC BASIC COLORS.
                                        ; DISPLAY IS STATIC (STOPPED) DURING THIS PHASE.
                                        ;
002354  005767  175702          HUES:   TST     LUTFLG
002360  001435                         BEQ     DONE1           ; BYPASS IF NO LUT AVAILABLE.
002362  012705  002434'                 MOV     #3$,R5          ; COLOR TABLE ADDRESS.
002366  012500                  1$:     MOV     (R5)+,R0        ; 1ST/NEXT COLOR.
002370  001431                         BEQ     DONE1           ; BR IF END OF TABLE.
002372  004767  176276                 JSR     PC,SHADES       ; BLAST IT.
002376  016767  175672  175672         MOV     FCD,FCD1
002404  006367  175666                 ASL     FCD1            ; EXTEND THE DELAY TIME.
002410  006367  175662                 ASL     FCD1
002414                          2$:
002414  104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002420  104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002424  005367  175646                 DEC     FCD1
002430  001371                         BNE     2$              ; DELAY...
002432  000755                         BR      1$              ;...AND LOOP THRU ALL COLORS.

002434  007400                  3$:     7400            ; BLUE
002436  000017                          0017            ; RED
002440  007417                          7417            ; MAGENTA
002442  000360                          0360            ; GREEN
002444  007760                          7760            ; CYAN
002446  000377                          0377            ; YELLOW
002450  007777                          7777            ; WHITE
002452  000000                          0               ; BLACK (END)
                                        ;
                                        ; THIS IS A COMMON "END-ITERATION" POINT FOR BOTH PARTS.
                                        ;
002454  032767  000001  175334  DONE1:  BIT     #BIT0,SR1       ; PART 2 ENABLED ??
002462  001031                         BNE     PART2           ; DO IT IF  SO.
002464  046767  175614  175622  DONE2:  BIC     DEVON,T2        ; DONE ALL MODULES?             <<VER B>>
002472  001421                         BEQ     PASS            ; YES SIGNAL EOP                <<VER B>>
002474  006367  175604                 ASL     DEVON           ; SELECT NEXT MODULE            <VERSION A1>
002500  032767  000001  175312         BIT     #1,SR2          ; USING COLOR LOOKUP?           <<VER B>>
002506  001403                         BEQ     2$              ; NO!                           <<VER B>>
002510  062767  000010  175562         ADD     #10,ADDRX       ; FOUR MORE CSR'S THEN          <<VER B>>
002516  062767  000010  175554  2$:     ADD     #ADDBAS,ADDRX   ; GET NEW ADDRESS               <VERSION A1>
002524  062767  000020  175550         ADD     #VECBAS,VECTXX  ; PLUS VECTOR                   <VERSION A1>
002532  000167  176326                 JMP     RESTRA          ; DO NEXT ONE

002536                          PASS:
002536  104413  000000'                 END1T$,BEGIN            ;SIGNAL END OF ITERATION.
002542  000167  176260                 JMP     RESTRT          ; GO FOR MORE                   <<VER B>>
```



```
                                        .SBTTL  *** PART 2 -- THE OLD BOUNCING BALLS
                                        .SBTTL  ***     SR1 BIT0 = 1 = ENABLE PART 2.
                                        .SBTTL  ***     SR1 BIT1 = 1 = USE *PITE-IN-RETRACE MODE.
                                        .SBTTL  ***     SR1 BIT2 = 1 = DISABLE BELL ON BOUNCE.
                                        .SBTTL
                                        ;
                                        ; DISPLAY TWO OCTAGONS FLOATING AROUND AT RANDOM.
                                        ; BOUNCE THEM OFF THE *ALLS AND "CRASH" IF THEY TOUCH.
                                        ;
002546  016777  175554  175446  PART2:  MOV     DSTRT,@DPC      ; CLEAR SCREEN (R/W MODE).
002554                          1$:
002554  104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002560  104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002564  005777  175434                 TST     @DSR
002570  100371                         BPL     1$
002572  005767  175464                 TST     LUTFLG
002576  001402                         BEQ     2$
002600  004767  176022                 JSR     PC,GRASCL
002604  012767  002000  175500  2$:     MOV     #2000,T1        ; SETUP DEAD MAN TIMER          <<VER B>>
002612  012767  000200  000536         MOV     #200,LOLIM
002620  012767  001600  000532         MOV     #1600,HILIM     ;SET LIMITS FOR ORIGINS.
002626  004767  000456                 JSR     PC,RAND
002632  010167  007340                 MOV     R1,RB           ;NEW ORIGIN, RED BALL.
002636  004767  000446                 JSR     PC,RAND
002642  010167  007332                 MOV     R1,RB+2
002646  004767  000436                 JSR     PC,RAND
002652  010167  007332                 MOV     R1,GB           ;NEW ORIGIN, GREEN BALL.
002656  004767  000426                 JSR     PC,RAND
002662  010167  007324                 MOV     R1,GB+2
002666  012767  000004  000462         MOV     #4,LOLIM
002674  012767  000030  000456         MOV     #30,HILIM       ;SET LIMITS FOR DELTAS.
002702  004767  000402                 JSR     PC,RAND
002706  010167  000624                 MOV     R1,DX1          ;BALL 1, DELTA X...
002712  004767  000372                 JSR     PC,RAND
002716  010167  000616                 MOV     R1,DX1+2        ;...AND Y.
002722  004767  000362                 JSR     PC,RAND
002726  010167  000610                 MOV     R1,DX2          ;BALL 2, DITTO.
002732  004767  000352                 JSR     PC,RAND
002736  010167  000602                 MOV     R1,DX2+2

002742  016767  175360  175360         MOV     DSTRT,DLOOP
002750  062767  006210  175352         ADD     #B1-DBEGIN,DLOOP ; INIT STARTING ADDRESS.
002756  032767  000002  175032         BIT     #BIT1,SR1       ; DYNAMIC MODE (DEFAULT) ??
002764  001007                         BNE     3$              ; BR IF NOT SO.
002766  062767  000012  175334         ADD     #B1A-B1,DLOOP   ; NO, ADJUST STARTING ADDRESS...
002774  012767  164002  007256         MOV     #SYNC+1,B4      ;...AND OPCODE FOR R/W MODE;
003002  000414                         BR      4$
003004  016777  175320  175210  3$:     MOV     DLOOP,@DPC      ; DYNAMIC, RECONFIGURE IM'S.
003012  005777  175206                 TST     @DSR            ; WAIT FOR DPU STOP.
003016  100375                         BPL     .-4
003020  062767  000036  175302         ADD     #B2-B1,DLOOP    ; ADJUST STARTING ADDRESS...
003026  012767  170340  007224         MOV     #SWTCH!CLRMEM,B4 ;...AND OPCODE FOR DYNAM MODE.
```

```
828 003034  012767  020040  007160  4$:     MOV     #20040,B3       ; NULL (SPACE) THE CRASH TEXT.
829 003042  016767  007154  007154          MOV     B3,B3+2
830 003050  016767  007146  007150          MOV     B3,B3+4
831 003056  012777  003114' 175154          MOV     #STOP2,@STPV    ; SET STOP VECTOR...
832 003064  016767  175204  175204          MOV     FCD,FCD1
833 003072  006367  175200                   ASL     FCD1            ;...EXTEND THE FRAME DELAY.
834 003076  006367  175174                   ASL     FCD1
835 003102  016777  175222  175112  CNTNU:  MOV     DLOOP,@DPC      ; START/RESTART DISPLAY.
836 003110  104400  000000'                  EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
837                                 ;
838                                 ; CONTINUE HERE ON STOP INTERRUPT.
839                                 ; MOVE THE BALLS AROUND UNTIL THEY CRASH, THEN END-PASS.
840                                 ;
841 003114                          STOP2:
    003114  000004  000000' 003122'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
842 003122  126727  007074  000040  1$:     CMPB    B3,#40          ; ARE WE STILL RUNNING ??
843 003130  001405                          BEQ     2$              ; BR IF SO.
844 003132  005367  175140                  DEC     FCD1            ; NO, WE'RE DEAD -- DISPLAY...
845 003136  001061                          BNE     4$              ;...THE MESSAGE FOR A WHILE...
846 003140  000167  177320                  JMP     DONE2           ;...AND SIGNAL ALL DONE.
847
848 003144  016767  007026  007000  2$:     MOV     RB,ORB          ; OK, COPY THE ORIGINS...
849 003152  016767  007022  006774          MOV     RB+2,ORB+2      ;...SO WE CAN ERASE.
850 003160  016767  007024  006776          MOV     GB,OGB
851 003166  016767  007020  006772          MOV     GB+2,OGB+2
852 003174  016701  006776                  MOV     RB,R1           ;NOW LET'S SEE IF WE'VE...
853 003200  166701  007004                  SUB     GB,R1           ;...HAD A CRASH.
854 003204  100001                          BPL     .+4             ;IF DIFF NEG...
855 003206  005401                          NEG     R1              ;...MAKE POSITIVE.
856 003210  020127  000034                  CMP     R1,#34          ;X DIFF < 34 ??
857 003214  101025                          BHI     3$              ; NO, WE'RE OK !!!
858 003216  016701  006756                  MOV     RB+2,R1         ;YES, CRASH POSSIBLE...
859 003222  166701  006764                  SUB     GB+2,R1         ;...COMPARE Y ORIGINS.
860 003226  100001                          BPL     .+4             ;IF NEG DIFF...
861 003230  005401                          NEG     R1              ;...MAKE POSITIVE.
862 003232  020127  000034                  CMP     R1,#34          ;Y DIFF < 34 ??
863 003236  101014                          BHI     3$              ; BR IF NOT, WE'RE STILL OK.
864
865 003240  012767  051103  006754          MOV     #"CR,B3         ; OTHERWISE...
866 003246  012767  051501  006750          MOV     #"AS,B3+2       ;...SAY "CRASH" !!!!
867 003254  012767  020110  006744          MOV     #"H ,B3+4
868 003262  004767  000416                  JSR     PC,DING
869 003266  000402                          SKP2
870 003270  004767  000252          3$:     JSR     PC,MOVE         ; MOVE THE BALLS AROUND.
871 003274  005367  175012                  DEC     I1              ; DON'T RUN FOREVER,                    <<VER B>>
872 003300  001401                          BEQ     5$              ; GET OUT EVENTUALLY!                   <<VER B>>
873 003302  000677                  4$:     BR      CNTNU
874 003304  000167  177154          5$:     JMP     DONE2
875
876                                 ;
877                                 ; RANDOM NUMBER GETTER.
878                                 ;
879                                 ;
880 003310                          RAND:
    003310  104417  000000'                  RAND$,BEGIN
881 003314  016701  174536                  MOV     RANNUM,R1
882 003320  042701  176000                  BIC     #^C1777,R1
```



```
883 003324  020167  000026          1$:     CMP     R1,LOLIM
884 003330  103003                          BHIS    2$
885 003332  066701  000020                  ADD     LOLIM,R1
886 003336  000772                          BR      1$
887 003340  020167  000014          2$:     CMP     R1,HILIM
888 003344  101403                          BLOS    3$
889 003346  166701  000006                  SUB     HILIM,R1
890 003352  000772                          BR      2$
891 003354  000207                  3$:     RTS     PC
892 003356  000000                  LOLIM:  0
893 003360  000000                  HILIM:  0
```

```
                                ;NON EXISTANCE MEMORY TRAP HANDLER
                                ;
003362                          TRP1:
003362  016737  174754  000004          MOV     TEMPC,@#MEMERR          ;RESTORE NON EXIST MEM ADR
003370  016737  174744  000006          MOV     TEMPS,@#MEMERR+2        ;RESTORE PS
003376  022626                          CMP     (SP)+,(SP)+             ;ADJUST STACK
003400  000167  175760                  JMP     SETUP
003404                          TRP2:                                   ;!!!!TRP2 ADDED AT <<VER B>>!!!!
003404  016737  174732  000004          MOV     TEMPC,@#MEMERR          ;RESTORE NON EXIST MEM ADR
003412  016737  174722  000006          MOV     TEMPS,@#MEMERR+2        ;RESTORE PS
003420  022626                          CMP     (SP)+,(SP)+             ;ADJUST STACK
003422  016700  174652                  MOV     ADDRX,R0                ;GET CSR
003426  010067  174446                  MOV     R0,CSRA                 ;PUT INTO CSRA
003432  012767  000010  174446          MOV     #10,ERRTYP              ;LOAD ERROR CODE
003440  104405  000000' 000000          HRDERS,BEGIN,NULL       ;NXM TIMOUT,MODULE DROPPED
003446  104401  000000' 003466'          MSGS,BEGIN,2$   ;ASCII MESSAGE CALL
003454  046767  174624  174634          BIC     DEVON,T3                ;REMOVE FROM LIST
003462  000167  176776                  JMP     DONE2                   ;NEXT..!
                                ;
                                ;
003466      045     040     040 2$:     .ASCIZ  '%  VSCB0 -- NXM TIMEOUT,MODULE DROPPED%'
003471      126     123     103
003474      102     060     040
003477      055     055     040
003502      116     130     115
003505      040     124     111
003510      115     105     117
003513      125     124     054
003516      115     117     104
003521      125     114     105
003524      040     104     122
003527      117     120     120
003532      105     104     045
003535      000
                                .EVEN
                                ;
                                ;
                                ;
                                ; ROUTINE TO MOVE THE BALLS AROUND ON THE SCREEN.
                                ;
003536  000000  000000          DX1:    0,0                     ;DELTAS FOR BALL 1...
003542  000000  000000          DX2:    0,0                     ;...AND 2.
                                ;
003546  012701  012176'          MOVE:   MOV     #RB,R1          ;RED BALL ORIGIN...
003552  012702  003536'                  MOV     #DX1,R2         ;...AND DELTA POINTERS.
003556  004767  000010                  JSR     PC,1$
003562  012701  012210'                  MOV     #GR,R1          ; GREEN BALL ORIGIN...
003566  012702  003542'                  MOV     #DX2,R2         ;...AND DELTA POINTERS.
                                ;
003572  061211                  1$:     ADD     (R2),(R1)       ;OLD X + DELTA X.
003574  021127  000024                  CMP     (R1),#24        ;HIT LEFT WALL ??
003600  101005                          BHI     2$              ;NO
003602  012711  000024                  MOV     #24,(R1)        ;YES, SET X AT MIN.....
003606  005412                          NEG     (R2)            ;...AND REVERSE DELTA X.
```



```
003610  004767  000070                  JSR     PC,DING         ;"DING"
                                ;
003614  021127  001752          2$:     CMP     (R1),#MAXX-24   ;HIT RIGHT WALL ??
003620  103405                          BLO     3$              ;NO
003622  012711  001752                  MOV     #MAXX-24,(R1)   ;YES, SET X AT MAX...
003626  005412                          NEG     (R2)            ;...AND REVERSE DELTA X.
003630  004767  000050                  JSR     PC,DING
                                ;
003634  005721                  3$:     TST     (R1)+           ;BUMP POINTERS FOR Y'S.
003636  005722                          TST     (R2)+
003640  061211                          ADD     (R2),(R1)       ;OLD Y + DELTA Y.
003642  021127  000024                  CMP     (R1),#24        ;HIT FLOOR ??
003646  101005                          BHI     4$              ;NO
003650  012711  000024                  MOV     #24,(R1)        ;YES, SET Y AT MIN...
003654  005412                          NEG     (R2)            ;...AND REVERSE DELTA Y.
003656  004767  000022                  JSR     PC,DING
                                ;
003662  021127  001652          4$:     CMP     (R1),#MAXY60-24 ;HIT CEILING ??
003666  103405                          BLO     5$              ;NO
003670  012711  001652                  MOV     #MAXY60-24,(R1) ;YES, SET Y AT MAX...
003674  005412                          NEG     (R2)            ;...AND REVERSE DELTA Y.
003676  004767  000002                  JSR     PC,DING
                                ;
003702  000207                  5$:     RTS     PC              ;EXIT.
                                ;
003704  032767  000004  174104  DING:   BIT     #BIT2,SR1       ;IF BIT2=1 BELL OFF
003712  001003                          BNE     1$              ; EXIT IF "BELL" DISABLED.
003714  104401  000000' 003724'          MSGS,BEGIN,BELL ;ASCII MESSAGE CALL
003722  000207                  1$:     RTS     PC
003724      007     000         BELL:   .ASCIZ  <7>
```

```
                                        .SBTTL  *** PART 1 -- DISPLAY CODE
                                        ;
                                        ; SMORGASBORD, ONE FRAME THAT HAS A BIT OF EVERYTHING.
                                        ; FIRST, A DEC LOGO (SORT OF), THEN VIOLATE THE EDGES
                                        ; AT LOWER LEFT AND UPPER RIGHT.
                                        ;
          100000                        DSUB = CHAR             ; USE CHAR MODE TO CALL SUBROUTINES.
                                                                ; *** REMEMBER THAT THEY ***
                                                                ; *** CANNOT BE NESTED   ***
                                                                ; CHAR CODES 1 - 11, AND 16 - 77
                                                                ; ARE AVAILABLE AS CALL ARGUMENTS.
                                                                ; APPROPRIATE DISPATCH POINTERS MUST
                                                                ; BE SET IN TABLE "ACAT".

003726  176074  176474           DBEGIN: RDWRT!CH0, RDWRT!CH1
003732  177074  177474                   RDWRT!CH2, RDWRT!CH3
003736  152000  012320'                  SETCB, ACAT             ; SET CHAR BASE POINTER.
003742  170140  173000                   CLRMEM, STOPN           ; CLEAR SCREEN RAM AND STOP

003746  114000  000540  000100   LOGO:   APNT, 540, 100
003754  103774                   CHAR!ALL
003756     104     040     111           .ASCIZ  /D I G I T A L/
003761     040     107     040
003764     111     040     124
003767     040     101     040
003772     114     000

003774  114000  000140  000140           APNT,140,140            ;EXCEED EDGES AT LOWER LEFT...
004002  110000                           LVEC
004004  060200  000000                   .WORD   $I+$X, $Y
004010  040000  020200                   .WORD   $I+$X, $Y
004014  040200  000000                   .WORD   $I+$X, $Y
004020  040000  000200                   .WORD   $I+$X, $Y
004024  114000  001640  001540           APNT,1640,1540          ;...AND AT UPPER RIGHT.
004032  110000                           LVEC
004034  040200  000000                   .WORD   $I+$X, $Y
004040  040000  000300                   .WORD   $I+$X, $Y
004044  060200  000000                   .WORD   $I+$X, $Y
004050  040000  020300                   .WORD   $I+$X, $Y
004054  173401                           STOPI+1                 ; END OF FRAME 1.
```



```
                                        ;
                                        ; NOW LONG, AND SHORT VECTORS, AND SOME RELATIVE POINTS.
                                        ;
004056  114000  000500  000002   OCTGNS: APNT,500,2
004064  113774                           LVEC!ALL                ; 1 LARGE OCTAGON...
004066  040576  000000                   .WORD   $I+$X, $Y
004072  040436  000436                   .WORD   $I+$X, $Y
004076  040000  000576                   .WORD   $I+$X, $Y
004102  060436  000436                   .WORD   $I+$X, $Y
004106  060576  000000                   .WORD   $I+$X, $Y
004112  060436  020436                   .WORD   $I+$X, $Y
004116  040000  020576                   .WORD   $I+$X, $Y
004122  040436  020436                   .WORD   $I+$X, $Y

004126  114000  000160  000620           APNT,160,620            ;...AND 2 SMALL ONES.
004134  100000  000001                   DSUB, 1                 ; CALL OCTAGON SUBROUTINE.

004140  114000  001520  000620           APNT,1520,620
004146  100000  000001                   DSUB, 1

004152  173402                           STOPI+2                 ; END OF FRAME 2.

004154                           SUB.1:                  ; SUBROUTINE FOR SMALL OCTAGON...
004154  107774                           SVEC!ALL                ;...CALLED VIA CHAR CODE 1.
004156  057400                           .WORD   $I+$X+$Y
004160  053456                           .WORD   $I+$X+$Y
004162  040076                           .WORD   $I+$X+$Y
004164  073456                           .WORD   $I+$X+$Y
004166  077400                           .WORD   $I+$X+$Y
004170  073556                           .WORD   $I+$X+$Y
004172  040176                           .WORD   $I+$X+$Y
004174  053556                           .WORD   $I+$X+$Y
004176  130000                           RPNT                    ;5 REL POINTS IN CENTER.
004200  004050                           .WORD   $I+$X+$Y
004202  044050                           .WORD   $I+$X+$Y
004204  044000                           .WORD   $I+$X+$Y
004206  064020                           .WORD   $I+$X+$Y
004210  064120                           .WORD   $I+$X+$Y
004212  044120                           .WORD   $I+$X+$Y
004214  165000                           DPOP
```

```
1048                                    ;
1049                                    ; NEXT, SHOW GRAPH/HISTO Y IN UPPER LEFT.
1050                                    ;
1051            001240                  YVAL = 1240
1052 004216  114000  000100  001240  GRY:    APNT,100,YVAL
1053 004224  113774                          LVEC!ALL                ;DRAW BASE LINE.
1054 004226  040340  000000                  I!340, 0
1055 004232  020344  000000                  MXY!344, 0
1056 004236  174104                          GXI!4                   ;X INCR = 4 FOR GRAPH.
1057 004240  127774                          GHY!ALL                 ; GRAPH Y MODE.
1058            000071                       .REPT   71
1059                                         0!YVAL
1060                                         YVAL=YVAL+4
1061                                         .ENDM
     004242  001240                         0!YVAL
     004244  001244                         0!YVAL
     004246  001250                         0!YVAL
     004250  001254                         0!YVAL
     004252  001260                         0!YVAL
     004254  001264                         0!YVAL
     004256  001270                         0!YVAL
     004260  001274                         0!YVAL
     004262  001300                         0!YVAL
     004264  001304                         0!YVAL
     004266  001310                         0!YVAL
     004270  001314                         0!YVAL
     004272  001320                         0!YVAL
     004274  001324                         0!YVAL
     004276  001330                         0!YVAL
     004300  001334                         0!YVAL
     004302  001340                         0!YVAL
     004304  001344                         0!YVAL
     004306  001350                         0!YVAL
     004310  001354                         0!YVAL
     004312  001360                         0!YVAL
     004314  001364                         0!YVAL
     004316  001370                         0!YVAL
     004320  001374                         0!YVAL
     004322  001400                         0!YVAL
     004324  001404                         0!YVAL
     004326  001410                         0!YVAL
     004330  001414                         0!YVAL
     004332  001420                         0!YVAL
     004334  001424                         0!YVAL
     004336  001430                         0!YVAL
     004340  001434                         0!YVAL
     004342  001440                         0!YVAL
     004344  001444                         0!YVAL
     004346  001450                         0!YVAL
     004350  001454                         0!YVAL
     004352  001460                         0!YVAL
     004354  001464                         0!YVAL
     004356  001470                         0!YVAL
     004360  001474                         0!YVAL
     004362  001500                         0!YVAL
     004364  001504                         0!YVAL
     004366  001510                         0!YVAL
```



```
     004370  001514                         0!YVAL
     004372  001520                         0!YVAL
     004374  001524                         0!YVAL
     004376  001530                         0!YVAL
     004400  001534                         0!YVAL
     004402  001540                         0!YVAL
     004404  001544                         0!YVAL
     004406  001550                         0!YVAL
     004410  001554                         0!YVAL
     004412  001560                         0!YVAL
     004414  001564                         0!YVAL
     004416  001570                         0!YVAL
     004420  001574                         0!YVAL
     004422  001600                         0!YVAL
1062
1063            001240                  YVAL = 1240
1064 004424  150000  001240          HSY:    SETHB,YVAL              ; SET HISTO BASE LINE.
1065 004430  114000  000130  001240          APNT,130,YVAL
1066 004436  174110                          GXI!10                  ;X INCR = 10 FOR HISTO.
1067 004440  127774                          GHY!ALL
1068            000031                       .REPT   31
1069                                         HST!YVAL
1070                                         YVAL=YVAL+10
1071                                         .ENDR
     004442  041240                         HST!YVAL
     004444  041250                         HST!YVAL
     004446  041260                         HST!YVAL
     004450  041270                         HST!YVAL
     004452  041300                         HST!YVAL
     004454  041310                         HST!YVAL
     004456  041320                         HST!YVAL
     004460  041330                         HST!YVAL
     004462  041340                         HST!YVAL
     004464  041350                         HST!YVAL
     004466  041360                         HST!YVAL
     004470  041370                         HST!YVAL
     004472  041400                         HST!YVAL
     004474  041410                         HST!YVAL
     004476  041420                         HST!YVAL
     004500  041430                         HST!YVAL
     004502  041440                         HST!YVAL
     004504  041450                         HST!YVAL
     004506  041460                         HST!YVAL
     004510  041470                         HST!YVAL
     004512  041500                         HST!YVAL
     004514  041510                         HST!YVAL
     004516  041520                         HST!YVAL
     004520  041530                         HST!YVAL
     004522  041540                         HST!YVAL
1072 004524  173403                         STOPI+3                 ; END OF FRAME 3.
```

```
                                        ;
                                        ; NEXT, GRAPH/HISTO X IN LOWER RIGHT.
                                        ;
                 001340                 XVAL = 1340
004526  114000  001340  000100  GRX:    APNT,XVAL,100
004534  113774                          LVEC!ALL                ;DRAW BASE LINE.
004536  040000  000340                   I!0, 340
004542  000000  020344                   0, MXY!344
004546  174104                          GYI!4                   ;Y INCR = 4 FOR GRAPH.
004550  123774                          GHX!ALL
                 000071                 .REPT   71
                                        0!XVAL
                                        XVAL=XVAL+4
                                        .ENDR
     004552  001340                     0!XVAL
     004554  001344                     0!XVAL
     004556  001350                     0!XVAL
     004560  001354                     0!XVAL
     004562  001360                     0!XVAL
     004564  001364                     0!XVAL
     004566  001370                     0!XVAL
     004570  001374                     0!XVAL
     004572  001400                     0!XVAL
     004574  001404                     0!XVAL
     004576  001410                     0!XVAL
     004600  001414                     0!XVAL
     004602  001420                     0!XVAL
     004604  001424                     0!XVAL
     004606  001430                     0!XVAL
     004610  001434                     0!XVAL
     004612  001440                     0!XVAL
     004614  001444                     0!XVAL
     004616  001450                     0!XVAL
     004620  001454                     0!XVAL
     004622  001460                     0!XVAL
     004624  001464                     0!XVAL
     004626  001470                     0!XVAL
     004630  001474                     0!XVAL
     004632  001500                     0!XVAL
     004634  001504                     0!XVAL
     004636  001510                     0!XVAL
     004640  001514                     0!XVAL
     004642  001520                     0!XVAL
     004644  001524                     0!XVAL
     004646  001530                     0!XVAL
     004650  001534                     0!XVAL
     004652  001540                     0!XVAL
     004654  001544                     0!XVAL
     004656  001550                     0!XVAL
     004660  001554                     0!XVAL
     004662  001560                     0!XVAL
     004664  001564                     0!XVAL
     004666  001570                     0!XVAL
     004670  001574                     0!XVAL
     004672  001600                     0!XVAL
     004674  001604                     0!XVAL
     004676  001610                     0!XVAL
```



```
     004700  001614                     0!XVAL
     004702  001620                     0!XVAL
     004704  001624                     0!XVAL
     004706  001630                     0!XVAL
     004710  001634                     0!XVAL
     004712  001640                     0!XVAL
     004714  001644                     0!XVAL
     004716  001650                     0!XVAL
     004720  001654                     0!XVAL
     004722  001660                     0!XVAL
     004724  001664                     0!XVAL
     004726  001670                     0!XVAL
     004730  001674                     0!XVAL
     004732  001700                     0!XVAL

                 001340                 XVAL = 1340
004734  150000  001340          HSX:    SETHB,XVAL              ; SET HISTO BASE LINE.
004740  114000  001340  000130          APNT,XVAL,130
004746  174110                          GYI!10                  ;Y INCR = 10 FOR HISTO.
004750  123774                          GHX!ALL
                 000031                 .REPT   31
                                        HST!XVAL
                                        XVAL=XVAL+10
                                        .ENDR
     004752  041340                     HST!XVAL
     004754  041350                     HST!XVAL
     004756  041360                     HST!XVAL
     004760  041370                     HST!XVAL
     004762  041400                     HST!XVAL
     004764  041410                     HST!XVAL
     004766  041420                     HST!XVAL
     004770  041430                     HST!XVAL
     004772  041440                     HST!XVAL
     004774  041450                     HST!XVAL
     004776  041460                     HST!XVAL
     005000  041470                     HST!XVAL
     005002  041500                     HST!XVAL
     005004  041510                     HST!XVAL
     005006  041520                     HST!XVAL
     005010  041530                     HST!XVAL
     005012  041540                     HST!XVAL
     005014  041550                     HST!XVAL
     005016  041560                     HST!XVAL
     005020  041570                     HST!XVAL
     005022  041600                     HST!XVAL
     005024  041610                     HST!XVAL
     005026  041620                     HST!XVAL
     005030  041630                     HST!XVAL
     005032  041640                     HST!XVAL
005034  173404                          STOPI+4                 ; END OF FRAME 4.
```

```
                                        ;
                                        ; NEXT SHOW BIT MAP MODE 1 AT BOTTOM CENTER.
                                        ; AND BIT MAP MODE 0 AT TOP CENTER.
                                        ;
                                        ; DON'T FORGET TO FILL THE BUFFER FIRST.
                                        ;
005036  114000  000400  000240  B14:    APNT,400,240            ; 4 ON 4 OFF FOR 340 PIXELS.
005044  136340                          BM14!340
005046  005132'                         DBUFR
005050  164000                          DNOP

005052  114000  000300  000340  B18:    APNT,300,340            ; 2 ON 2 OFF FOR 400 PIXELS.
005060  137400                          BM18!400
005062  005132'                         DBUFR
005064  164000                          DNOP

005066  114000  000540  001340  B04:    APNT, 540, 1340         ; 4 BIT X 32 SQUARE...
005074  134010                          BM04!M32!EX2            ;...EXPANDED X 2.
005076  005132'                         DBUFR
005100  164000                          DNOP

005102  114000  001040  001340  B08:    APNT, 1040, 1340        ; 8 BIT X 32 SQUARE...
005110  135010                          BM08!M32!EX2            ;...EXPANDED X 2.
005112  005132'                         DBUFR
005114  173405                          STOP1+5                 ; END OF FRAME 5.

                                        ;
                                        ; FINALLY, AT DEAD-CENTER:
                                        ; VERTICAL COLOR BARS -- NUMBER OF COLORS/SHADES DEPENDANT
                                        ; UPON IMAGE MEMORY SIZE AND PRESENCE (OR ABSENCE) OF THE LUT.
                                        ; IF ALL 256. GREY-SCALE LEVELS ARE ACTIVE, WE'LL HAVE ONLY
                                        ; 1 VECTOR PER LEVEL !!!!!
                                        ; BUFFER REQUIRES <256.*5>+2 WORDS.
                                        ;

005116  114000  000376  000476  BARS:   APNT, MAPX-400, MAPY60-240
005124  100000  000002                  DSUB, 2                 ; EXECUTE CODE IN BUFFER.
005130  173406                          STOP1+6                 ; END OF FRAME 6.
                                        ;
                                        ; DISPLAY CODE BUFFER.
                                        ; SHARED BY THE BIT MAP AND BARS ROUTINES.
                                        ;
005132                                  SUB.2:                  ; SUBROUTINE CALL FOR "BARS".
005132                                  DBUFR:  .BLKW   <256.*5>+2
```



```
                                        .SBTTL  *** PART 2 -- DISPLAY CODE
                                        ;
                                        ; DISPLAY FILES FOR BOUNCING BALLS.
                                        ; IMAGE MEMORIES ARE IN R/W MODE INITIALLY.
                                        ; IF R/W MODE SELECTED, USE B1A AS ENTRY.
                                        ; OTHERWISE, RECONFIGURE IM'S AT B1, AND USE B2 AS THE ENTRY.
                                        ;
012136  176050  176434          B1:     READ!CH0, WRT!CH1       ; DYNAMIC MODE, RESET IM'S.
012142  177050  177434                  READ!CH2, WRT!CH3
012146  173000                          STOPN

012150  116000                  B1A:    APNT!NONE               ; READ/WRITE MODE...
012152  000000  000000          ORB:    0,0                     ;...ERASE OLD IMAGE...
012156  100000  000003                  DSUB, 3
012162  114000                          APNT
012164  000000  000000          OGB:    0,0
012170  100000  000003                  DSUB, 3                 ;...AND FALL THRU.

012174  116500                  B2:     APNT!RED                ; DYNAMIC MODE...
012176  000000  000000          RB:     0,0                     ;...REDRAW NEW IMAGE.
012202  100000  000003                  DSUB, 3
012206  117140                          APNT!GRN
012210  000000  000000          GB:     0,0
012214  100000  000003                  DSUB, 3
012220  103774                          CHAR!WHITE
012222     040     040     040  B3:     .ASCII  /       /         ; "CRASH" IF THAT'S THE CASE.
012225     040     040     040
012230  117774  000000  000000          APNT!WHITE, 0, 0        ; FINALLY, REFRESH THE FRAME.
012236  110000                          LVEC
012240  041776  000000                  1+MAXX, 0
012244  040000  001676                  I+0, MAXY60
012250  061776  000000                  I+MXY+MAXX, 0
012254  040000  021676                  I+0, MXY+MAXY60

012260  164002                  B4:     SYNC+1                  ; "SWITCH!CURMEM" IF DYNAMIC.
012262  164000  164000                  DNOP, DNOP
012266  173400                          STOP1
                                        ;
                                        ; OCTAGON SUBROUTINE.
                                        ;
012270                                  SUB.3:                  ; CALL VIA CHAR CODE 3.
012270  104000                                  SVEC            ;DRAW CCW FROM LO LEFT CORNER.
012272  022124                                  .WORD   $I+$X+$Y
012274  044000                                  .WORD   $I+$X+$Y
012276  043014                                  .WORD   $I+$X+$Y
012300  040020                                  .WORD   $I+$X+$Y
012302  063014                                  .WORD   $I+$X+$Y
012304  064000                                  .WORD   $I+$X+$Y
012306  063114                                  .WORD   $I+$X+$Y
012310  040120                                  .WORD   $I+$X+$Y
012312  043114                                  .WORD   $I+$X+$Y
012314  002024                                  .WORD   $I+$X+$Y
012316  165000                                  DPOP
```

```
                                        .SBTTL  *** ASCII CHARACTER ADDRESS TABLE AND SUBPIX
                                        ;
                                        ;THE FOLLOWING 128. WORDS CONTAIN THE CHARACTER SUBPIX
                                        ;STARTING ADDRESSES.  0-37 AND 140-177 ARE NULL FOR NOW.
                                        ;       (EXCEPT FOR 12 AND 15)
                                        ;
012320  012720'                 ACAT:   ..NULL          ;0-37 ARE NULL (NLISTED).

        012420'                         $SVPC=.         ;SAVE PC, BACK UP AND...
        012344'                         .=ACAT+<12*2>
012344  012722'                         ..LF            ;...INSERT LINE FEED...
        012352'                         .=ACAT+<15*2>
012352  012732'                         ..CR            ;...AND CARRIAGE RETURN ADDRESSES.
        012420'                         .=$SVPC         ;RESTORE PC

012420  012746'                         ..SPC           ;SPACE
012422  012754'                         ..EXC           ;EXCLAM
012424  012772'                         ..QUO           ;DOUBLE QUOTE.
012426  013010'                         ..NUM           ;NUMBER SIGN
012430  013036'                         ..DOL           ;DOLLAR
012432  013070'                         ..PCT           ;PER CENT
012434  013130'                         ..AND           ;AMPERSAND
012436  013170'                         ..QUO1          ;SINGLE QUOTE
012440  013202'                         ..LPAR          ;LFT PAREN
012442  013220'                         ..RPAR          ;RT PAREN
012444  013236'                         ..AST           ;ASTERISK
012446  013260'                         ..PLS           ;PLUS
012450  013276'                         ..CMA           ;COMMA
012452  013310'                         ..MNS           ;MINUS
012454  013322'                         ..DOT           ;PERIOD
012456  013332'                         ..SLSH          ;SLASH
012460  013346'                         ..00
012462  013400'                         ..11
012464  013420'                         ..22
012466  013452'                         ..33
012470  013512'                         ..44
012472  013532'                         ..55
012474  013562'                         ..66
012476  013614'                         ..77
012500  013634'                         ..88
012502  013704'                         ..99
012504  013734'                         ..COLN          ;COLON
012506  013752'                         ..SEMI          ;SEMI COLON
012510  013770'                         ..LANG          ;LFT ANGLE
012512  014004'                         ..EQ            ;EQUALS
012514  014022'                         ..RANG          ;RT ANGLE
012516  014034'                         ..QUES          ;QUESTION
012520  014062'                         ..ATS           ;AT SIGN
012522  014116'                         ..AA
012524  014142'                         ..BB
012526  014176'                         ..CC
012530  014224'                         ..DD
012532  014246'                         ..EE
012534  014254'                         ..FF
012536  014272'                         ..GG
012540  014310'                         ..HH
012542  014330'                         ..II
```



```
012544  014352'                         ..JJ
012546  014372'                         ..KK
012550  014410'                         ..LL
012552  014424'                         ..MM
012554  014442'                         ..NN
012556  014462'                         ..OO
012560  014512'                         ..PP
012562  014534'                         ..QQ
012564  014550'                         ..RR
012566  014564'                         ..SS
012570  014622'                         ..TT
012572  014640'                         ..UU
012574  014662'                         ..VV
012576  014702'                         ..WW
012600  014722'                         ..XX
012602  014746'                         ..YY
012604  014772'                         ..ZZ
012606  015014'                         ..LBRK          ;LFT BRACKET
012610  015032'                         ..BSLH          ;BACK SLASH
012612  015050'                         ..RBRK          ;RT BRACKET
012614  015066'                         ..CRT           ;CARROT
012616  015102'                         ..USC           ;UNDERSCORE
012620  012720'                         ..NULL          ;140-177 ARE NULL (NLISTED).
```

```
                                ; THESE ARE THE CHAR SUBPIX ROUTINES.
                                ; EACH IS BUILT ON A 5 X 7 DOT MATRIX (ALA VT05).
                                ; <CR> WILL INVOKE A "STOP".  CPU MUST CALCULATE THE START
                                ; CO-ORDS FOR NEXT LINE, STUFF IN "..CRX AND ..CRX+2" AND
                                ; RESUME, TO XCT PSUEDO-CRLF.
                                ;
                                ; THE FOLLOWING MACRO (SXY4) APPLIES A 4X SCALE FACTOR AT
                                ; ASSEMBLY TIME TO YIELD A CHAR SIZE OF 30 X 40 (OCTAL)
                                ; WHICH PRODUCES A MEDIUM SIZE CHAR FONT.
                                ;
                                        .MACRO  SXY4 A,B,C
                                        SXY     A,B*4,C*4
                                        .ENDM   SXY4

1303 012720  165000             ..NULL: DPOP                    ;ALL UNDEFINED CHARS EXIT HERE.

                                ;****** CHAR SET IS .NLISTED ***********
```



```
                                .SBTTL  *** 256 LEVEL LUMINANCE (GREY-SCALE) TABLE
                                ;
                                ; THIS IA A TABLE OF 256 (8 BITS) LUMINANCE VALUES
                                ; WHICH SHOULD PRODUCE A LINEAR 12 BIT GREY-SCALE.
                                ;
                                ; ON THE GREY-SCALE, 36 = BLUE, 120 = RED, AND 230 = GREEN.
                                ;
015112                          LUMTBL:
015112 000000 000400 001000     0,      400,    1000,   1
015120 000001
015122 001400 000401 002000     1400,   401,    2000,   1001
015130 001001
015132 002400 000002 003000     2400,   2,      3000,   402
015140 000402
015142 003400 002401 004000     3400,   2401,   4000,   3
015150 000003
015152 003001 002020 005000     3001,   2020,   5000,   2420
015160 002420
015162 000004 005400 003002     4,      5400,   3002,   6000
015170 006000
015172 001040 006400 002403     1040,   6400,   2403,   5
015200 000005
015202 003003 002040 007400     3003,   2040,   7400,   5020    ; LEVEL 37
015210 005020
015212 002404 000060 000006     2404,   60,     6,      1441
015220 001441
015222 003004 002023 001042     3004,   2023,   1042,   2405
015230 002405
015232 000007 004440 002024     7,      4440,   2024,   1061
015240 001061
015242 000100 000010 000462     100,    10,     462,    2025
015250 002025
015252 001044 002407 000011     1044,   2407,   11,     501
015260 000501
015262 007440 001063 000120     7440,   1063,   120,    12
015270 000012
015272 000502 003500 001064     502,    3500,   1064,   2411
015300 002411
015302 005407 000503 002030     5407,   503,    2030,   1065    ; LEVEL 77
015310 001065
015312 000140 000014 000504     140,    14,     504,    2031
015320 002031
015322 001066 002413 000015     1066,   2413,   15,     505
015330 000505
015332 007444 002540 000160     7444,   2540,   160,    16
015340 000016
015342 003014 003504 002541     3014,   3504,   2541,   125
015350 000125
015352 000017 002160 002034     17,     2160,   2034,   200
015360 000200
015362 000126 003142 002161     126,    3142,   2161,   3506
015370 003506
015372 001072 001600 007066     1072,   1600,   7066,   2162
015400 002162
015402 002036 000220 001601     2036,   220,    1601,   3126    ; LEVEL 137
015410 003126
015412 002163 005161 002563     2163,   5161,   2563,   131
```

```
        015420  000131
   1912 015422  004600  000513  005162  4600,   513,    5162,   240
        015430  000240
   1913 015432  000132  003146  002165  132,    3146,   2165,   1222
        015440  001222
   1914 015442  004220  004074  004602  4220,   4074,   4602,   2166
        015450  002166
   1915 015452  003513  000260  004075  3513,   260,    4075,   3132
        015460  003132
   1916 015462  003640  001152  002477  3640,   1152,   2477,   3223
        015470  003223
   1917 015472  000607  003567  000300  607,    3567,   300,    210
        015500  000210
   1918 015502  005476  002243  005241  5476,   2243,   5241,   1154    ; LEVEL 177
        015510  001154
   1919 015512  004152  004660  003153  4152,   4660,   3153,   2172
        015520  002172
   1920 015522  003517  000320  004117  3517,   320,    4117,   646
        015530  000646
   1921 015532  002173  001230  002555  2173,   1230,   2555,   157
        015540  000157
   1922 015542  000647  007624  000340  647,    7624,   340,    2574
        015550  002574
   1923 015552  005554  002321  002213  5554,   2321,   2213,   2703
        015560  002703
   1924 015562  000267  003265  004612  267,    3265,   4612,   3647
        015570  003647
   1925 015572  000360  001741  003266  360,    1741,   3266,   4631
        015600  004631
   1926 015602  002177  005213  005703  2177,   5213,   5703,   1634    ; LEVEL 237
        015610  001634
   1927 015612  000671  003651  005176  671,    3651,   5176,   272
        015620  000272
   1928 015622  001617  006266  001362  1617,   6266,   1362,   4360
        015630  004360
   1929 015632  000273  007232  000655  273,    7232,   655,    3653
        015640  003653
   1930 015642  001237  004235  004743  1237,   4235,   4743,   1256
        015650  001256
   1931 015652  007615  001312  005237  7615,   1312,   5237,   3345
        015660  003345
   1932 015662  000731  007616  002712  731,    7616,   2712,   1747
        015670  001747
   1933 015672  006235  002331  003656  6235,   2331,   3656,   4364
        015700  004364
   1934 015702  000277  003275  002332  277,    3275,   2332,   5330    ; LEVEL 277
        015710  005330
   1935 015712  002714  005712  003276  2714,   5712,   3276,   7745
        015720  007745
   1936 015722  001370  004366  007364  1370,   4366,   7364,   3277
        015730  003277
   1937 015732  002334  002371  004367  2334,   2371,   4367,   5714
        015740  005714
   1938 015742  004751  006276  001372  4751,   6276,   1372,   2717
        015750  002717
   1939 015752  005715  004752  006277  5715,   4752,   6277,   1373
        015760  001373
```



```
   1940 015762  004371  007367  004753  4371,   7367,   4753,   2337
        015770  002337
   1941 015772  001374  004372  005717  1374,   4372,   5717,   4754
        016000  004754
   1942 016002  007752  001375  004373  7752,   1375,   4373,   1757    ; LEVEL 337
        016010  001757
   1943 016012  004755  007753  005337  4755,   7753,   5337,   4374
        016020  004374
   1944 016022  007372  003357  006355  7372,   3357,   6355,   7754
        016030  007754
   1945 016032  005374  002776  005774  5374,   2776,   5774,   7337
        016040  007337
   1946 016042  006356  005357  004376  6356,   5357,   4376,   7374
        016050  007374
   1947 016052  004776  007756  006757  4776,   7756,   6757,   4377
        016060  004377
   1948 016062  007375  006376  007757  7375,   6376,   7757,   2777
        016070  002777
   1949 016072  006776  005777  006376  6776,   5777,   6376,   6377
        016100  006377
   1950 016102  007776  006777  007377  7776,   6777,   7377,   7777    ; LEVEL 377
        016110  007777
```

```
1952                                .SBTTL  *** PATCH AREA AND END
1953                                ;
1954                                ; PATCHES GO HERE.
1955                                ;
1956 016112                        PATCH:  .BLKW   40
1957
1958            016212'             LASTAD= .               ; END OF MODULE.
1959
1960                                .SBTTL  ****** TH - TH - TH - TH - THAT'S ALL FOLKS ! ******
1961            000001                      .END
```



```
ACAT    012320R  CLRMEM= 170140  GHY    = 124000  MOVE    003546R  RES2    000060R
ACSR    000102R  CNTNU   003102R  GHASCL  000626R  MSGN$ = 104403  RPNT  = 130000
ADDBAS= 000010   CONFIG  000056R  GRN    = 003140  MSGS$ = 104402  RSTRT   000112R
ADDR    000006R  CSRA    000100R  GRX     004526R  MSGS  = 104401  R256  = 000100
ADDRX   000300R  CUIM  = 146016   GRY     004216R  MSX   = 020000  R6    =%000006
ADDR22= 001000   CUIOFF= 146012   GWBUFS= 104414  MSY   = 000100  R7    =%000007
ALL   = 003774   CUIS  = 146013   GXI    = 174100  MXY   = 020000  SAVDPU  000452R
APNT  = 114000   CUOFF = 146040   GYI    = 174100  M128  = 000002  SBADR   000102R
ASB     000106R  CUON  = 146060   HAFX   = 000776  M256  = 000003  SDPC    000252R
ASTAT   000104R  CURD  = 146100   HAFY   = 000776  M32   = 000000  SDSR    000254R
AWAS    000110R  DATCK$= 104411   HAFY50= 000776  M64   = 000001  SDXR    000256R
BADPAD  000336R  DATER$= 104404   HAFY60= 000736  NEXT    001774R  SDYR    000260R
BARS    005116R  DBEGIN  003726R  HILIM   003360R  NONE  = 002000  SETCB = 152000
BEGIN   000000R  DBUFR   005132R  HRDCNT  000044R  NULL  = 000000  SETHB = 150000
BELL    003724R  DEVON   000304R  HRDER$= 104405  OCTGNS  004056R  SETMEM= 170100
BIT0  = 000001   DING    003704R  HRDPAS  000050R  OGB     012164R  SETRR = 000014
BIT1  = 000002   DIVCHK  000306R  HS1    = 040000  ONE     001042R  SETUP   001364R
BIT10 = 002000   DJMP  = 160000   HSX     004734R  OPEN  = 000000  SHADES  000674R
BIT11 = 004000   DLOOP   000330R  HSY     004424R  ORB     012152R  SIOFF = 171000
BIT12 = 010000   DNOP  = 164000   HUES    002354R  OTOA$ = 104420  SION  = 171400
BIT13 = 020000   DONE1   002454R  I      = 040000  PADDR   000322R  SKP1  = 000401
BIT14 = 040000   DONE2   002464R  ICONT   000036R  PART1   001632R  SKP2  = 000402
BIT15 = 100000   DPC     000222R  ICOUNT  000040R  PART2   002546R  SKP3  = 000403
BIT2  = 000004   DPOP  = 165000   IDNUM   000122R  PASCNT  000034R  SLDR    000266R
BIT3  = 000010   DRR     000224R  INIT    000030R  PASS    002536R  SLMR    000270R
BIT4  = 000020   DSR     000224R  INTR    000120R  PATCH   016112R  SLSR    000264R
BIT5  = 000040   DSTMT   000326R  JSMV    000242R  PIRQ$ = 000004  SOFCNT  000042R
BIT6  = 000100   DSUB  = 100000   JSSV    000246R  POPSP = 005726  SOFER$= 104406
BIT7  = 000200   DVID1   000014R  JS0    = 000000  POPSP2= 022626  SOFPAS  000046R
BIT8  = 000400   DX     = 000002  JS1    = 000400  PROTEC= 176000  SPUINT  000032R
BIT9  = 001000   DXR     000226R  JS2    = 001000  PRTY  = 000000  SPSIZ = 000040
BLU   = 002170   DX1     003536R  JS3    = 001400  PRTY0 = 000000  SR1     000016R
BLXIT   000756R  DX2     003542R  LASTAD= 016212R  PRTY1 = 000040  SR2     000020R
BM04  = 134000   DYI   = 000002   LDI    = 002000  PRTY2 = 000100  SR3     000022R
BM08  = 135000   DYNI  = 000004   LDR     000234R  PRTY3 = 000140  SR4     000024R
BM14  = 136000   DYR     000230R  LDUN   = 100000  PRTY4 = 000200  START   001000R
BM18  = 137000   EABITS  000324R  LMR     000236R  PRTY5 = 000240  STAT    000026R
BREAK$= 104407   ENDITS= 104413   LOGO    003746R  PRTY6 = 000300  STOP  = 172000
BR1     000012R  END$  = 104410   LOLIM   003356R  PRTY7 = 000340  STOPI = 173400
BR2     000013R  ERRTYP  000106R  LREAD = 010000  PS    = 177776  STOPN = 173000
BTOD$ = 104421   EXITS = 104400   LSET  = 010000  PSW   = 177776  STOP1   002022R
B04     005066R  EXSM2 = 000050   LSR     000232R  PUSH  = 005746  STOP2   003114R
B08     005102R  EXSM4 = 000060   LUMTBL  015112R  PUSH2 = 024646  STPV    000240R
B1      012136R  EX2   = 000010   LUIFLG  000262R  RAND    003310R  SUB.1   004154R
B1A     012150R  EX4   = 000020   LUTIM   001256R  RAND$ = 104417  SUB.2   005132R
B14     005036R  FCD     000274R  LUTV    000250R  RANNUM  000054R  SUB.3   012270R
B18     005052R  FCD1    000276R  LVEC  = 110000  RB      012176R  SVEC  = 104000
B2      012174R  FIDX    000272R  L0    = 002000  RDWRT = 176074  SVR0    000062R
B3      012222R  FR2     002076R  MAP22$= 104416  RDWRT1= 176076  SVR1    000064R
B4      012260R  FR3     002106R  MATCH   000344R  READ  = 176050  SVR2    000066R
CANT    001564R  FR4     002116R  MAXX  = 001776  RED   = 002500  SVR3    000070R
CDATA$= 104412   FR5     002130R  MAXY  = 001776  RELOC   000332R  SVR4    000072R
CHAR  = 100000   FR6     002224R  MAXY50= 001776  RELOK   000334R  SVR5    000074R
CH0   = 000000   GB      012210R  MAXY60= 001676  RESM    002012R  SVR6    000076R
CH1   = 000400   GCOFF = 000100   MEMERR= 000004  RESTRA  001064R  SWE   = 000010
CH2   = 001000   GETPA$= 104415   MODNAM  000000R  RESTRT  001026R  SWITCH  000344R
CH3   = 001400   GHX   = 120000   MODSP   000222R  RES1    000056R  SWTCH = 170200
```

SEQ 0043

```
SYNC   = 164001          WDTO    000114R         ..CR    012732R         ..MM    014424R         ..SS    014564R
SYSCNT  000052R          WHITE = 003774          ..CRT   015066R         ..MNS   013310R         ..TT    014622R
TEMPC   000342R          WR1   = 176034          ..CRX   012740R         ..NN    014442R         ..USC   015102R
TEMPS   000340R          WR11  = 176036          ..DD    014224R         ..NULL  012720R         ..UU    014640R
TIMOUT  000366R          XFLAG   000005R         ..DUL   013036R         ..NUM   013010R         ..VV    014662R
TOTV    000244R          XVAL  = 001650          ..DOT   013322R         ..OO    014462R         ..WW    014702R
TRPOFF= 000022           YVAL  = 001550          ..EE    014246R         ..PCT   013070R         ..XX    014722R
TRP1    003362R          $1    = 000000          ..EQ    014004R         ..PLS   013260R         ..YY    014746R
TRP2    003404R          $SVPC = 012420R         ..EXC   012754R         ..PP    014512R         ..ZZ    014772R
T0      000310R          $X    = 002000          ..FF    014254R         ..QQ    014534R         ..00    013346R
T1      000312R          $Y    = 000000          ..GG    014272R         ..QUES  014034R         ..11    013400R
T2      000314R          ..AA    014116R         ..HH    014310R         ..QUO   012772R         ..22    013420R
T3      000316R          ..AND   013130R         ..II    014330R         ..QUO1  013170R         ..33    013452R
UNXLI   000512R          ..AST   013236R         ..JJ    014352R         ..RANG  014022R         ..44    013512R
VADDR   000320R          ..ATS   014062R         ..KK    014372R         ..RBRK  015050R         ..55    013532R
VECBAS= 000020           ..BB    014142R         ..LANG  013770R         ..RPAR  013220R         ..66    013562R
VECTOR  000010R          ..BSLH  015032R         ..LBRK  015014R         ..RR    014550R         ..77    013614R
VECTXX  000302R          ..CC    014176R         ..LF    012722R         ..SEMI  013752R         ..88    013634R
WASADR  000104R          ..CMA   013276R         ..LL    014410R         ..SLSH  013332R         ..99    013704R
WDFR    000116R          ..COLN  013734R         ..LPAR  013202R         ..SPC   012746R

. ABS.  000000      000
        016212      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  8480 WORDS  ( 34 PAGES)
DYNAMIC MEMORY:  9786 WORDS  ( 37 PAGES)
ELAPSED TIME:  00:02:47
```

```
                        .REM    -
```

IDENTIFICATION

PRODUCT CODE: AC-S210A-MC

PRODUCT NAME: CXTMDA0 DEC/X11 TM78 MAGTAPE EXERCISER MODULE

DATE: JUNE 1,1980

MAINTAINER: DIAGNOSTIC ENGINEERING

```
                        .REM _
```

TABLE OF CONTENTS

1.0 ABSTRACT

THE TMD IS AN IOMODX MODULE THAT CAN EXERCISE UP TO 8 TM78 CONTROLLERS (DRIVES) WITH UP TO 4 TU78 TAPE TRANSPORTS (SLAVES) ON A SINGLE TM78. EACH TU78 WILL BE EXERCISED BY DOING A WRITE, READ REVERSE, READ FORWARD AND AN IN-CORE COMPARE. THIS SEQUENCE OF FUNCTIONS WILL BE DEFINED AS A CYCLE. AN "END OF PASS" WILL BE REACHED AFTER 500 CYCLES, OR WHEN END OF TAPE (EOT) IS DETECTED. WHEN "END OF PASS" IS REACHED, A "TAPE MARK" FUNCTION IS EXECUTED BEFORE CONTINUING.

WHEN EOT IS DETECTED AND AFTER THE TAPEMARK IS WRITTEN, THE ERROR SUMMARY IS PRINTED, ALL TU78(S) ARE REWOUND, AND THE DENSITY CHANGED.
THIS MODULE UTILIZES THE AUTO REPOSITIONING FEATURES OF THE TM78 TO ATTEMPT RECOVERY FROM WRITE AND READ STATUS ERRORS. THEREFORE, RECOVERY SEQUENCES ARE AS DEFINED FOR THE TM78 OPERATIONAL MICRO CODE, AND RECOVERABLE/NON-RECOVERABLE ERROR REPORTS AND STATISTIC UPDATE ARE BASED SOLEY ON THE OPERATIONAL MICROCODE RETRY ALGORITHM RESULTS.

STATISTICS MAINTAINED FOR EACH TU78 ARE:

```
                        RECOVERABLE READ ERRORS
                        UNRECOVERABLE READ ERRORS
                        RECOVERABLE WRITE ERRORS

                    READ STATUS ERRORS ARE NOT PRINTED ON A TRY BY TRY BASIS BUT
                    RATHER THE FINAL OUTCOME OF EACH RETRY (RECOVERABLE OR NON-
                    RECOVERABLE) IS PRINTED ON A RECORD BY RECORD BASIS.

                    WRITE STATUS ERRORS ARE NOT PRINTED ON A TRY BY TRY BASIS BUT
                    ARE ALSO CLASSIFIED AS RECOVERABLE OR NON-RECOVERABLE. HOWEVER,
                    A UNRECOVERABLE WRITE ERROR WILL CAUSE THE UNIT TO BE DROPPED
                    AS THE OPERATIONAL MICRO CODE CANNOT CONTINUE, AND
                    MAINTAIN A READABLE (ANSI STANDARD) TAPE.


2.0     REQUIREMENTS
        ------------

        HARDWARE:   AT LEAST 1 TU78 SLAVE ON A TM78 CONTROLLER AND AN
                    RH11/RH70.

        SOFTWARE:   TMB REQUIRES 6K BYTES OF STORAGE - (3K WORDS)


3.0     PASS DEFINITION
        ---------------

        ONE PASS CONSISTS OF 500 CYCLES OF A WRITE, READ REVERSE,
        READ FORWARD, AND DATA CHECK OPERATIONS.


4.0     EXECUTION TIME
        --------------

        ONE PASS OF TMD TAKES APPROXIMATELY MINUTE.


5.0     CONFIGURATION REQUIREMENTS
        --------------------------

        DEFAULT PARAMETERS:     DEVADR: 172400    VECTOR: 224    BR1: 5

        REQUIRED PARAMETERS:    NONE


6.0     DEVICE/OPTION SETUP
        -------------------

        POWER UP ALL TU78'S AND MAKE READY AT LOAD POINT AND WRITE ENABLED.


7.0     MODULE OPERATION TEST SEQUENCE
        ------------------------------

        REFER TO SECTION 13.0.
```

```
524
525                          8.0     DRIVER OPERATION OPTIONS
526                                  ------------------------
527
528                                  SR1 - THE DEFAULT FOR ALL OPTIONS IS ZERO (0).
529                                  ---
530
531                                  BIT0 = 0:  ALLOW DATA COMPARE
532
533                                  BIT0 = 1:  INHIBIT DATA COMPARE
534
535                                  BIT1 = 0:  MODULE WILL EXERCISE ALL TU78'S FOUND ON THE USER
536                                             SPECIFIED MASSBUS CONTROLLER.
537
538
539                                  BIT1 = 1:  OPERATOR MUST SELECT THE TU78(S) IN ADDITION TO THE
540                                             TM78'S (DVID1) TO BE EXERCISED.  SEE SECTION 10.0 FOR
541                                             DETAILS.
542
543                                  BIT2 = 0:  REPORT RECOVERABLE/NON-RECOVERABLE ERRORS AS THEY OCCUR.
544
545                                  BIT2 = 1:  DO NOT REPORT RECOVERABLE/NON-RECOVERABLE ERRORS AS
546                                             THEY OCCUR.
547
548                                  BIT3 = 0:  PRINT ERROR SUMMARY AT EOT.
549
550                                  BIT3 = 1:  DO NOT PRINT ERROR SUMMARY AT EOT.
551
552                                  BIT4 = 0:  TEST GCR (6250 BPI)
553
554                                  BIT4 = 1:  DO NOT TEST GCR (6250 BPI)
555
556                                  BIT5 = 0:  TEST PE (1600 BPI)
557
558                                  BIT5 = 1:  DO NOT TEST PE (1600 BPI)
559
560
561                          9.0     SELECTING SLAVES
562                                  ----------------
563
564                                  BIT 1 OF SR1 = 1
565
566                                  WHEN BIT 1 IS SET IN SR1, THE OPERATOR MUST SELECT THE TU78(S)
567                                  TO BE TESTED.
568
569                                  TO SELECT THE TU78(S) FOR A PARTICULAR TM78, A "1" MUST BE SET
570                                  IN THE APPROPRIATE BIT POSITION FOR THE DESIRED TU78.  REFER TO
571                                  THE FOLLOWING TABLE.
572
573
574                                          UNIT TABLE FORMAT
575                                          ----------------
576
577                                                T U 7 8 #
578                                  <-----------------------------------
579
```

```
                        15        4   3     2     1     0
                        +-----------------------------------+
                        ! NOT USED ! TM0 ! TM0 ! TM0 ! TM0 !     !
                        !          ! TU3 ! TU2 ! TU1 ! TU0 !     !
                        !----------+-----+-----+-----+-----!     !
                        ! NOT USED ! TM1 ! TM1 ! TM1 ! TM1 !     !
                        !          ! TU3 ! TU2 ! TU1 ! TU0 !     !
                        !----------+-----+-----+-----+-----!     !
                        ! NOT USED ! TM2 ! TM2 ! TM2 ! TM2 !     !
                        !          ! TU3 ! TU2 ! TU1 ! TU0 !     !   T
                        !----------+-----+-----+-----+-----!     !
                        ! NOT USED ! TM3 ! TM3 ! TM3 ! TM3 !     !   M
                        !          ! TU3 ! TU2 ! TU1 ! TU0 !     !
                        !----------+-----+-----+-----+-----!     !   7
                        ! NOT USED ! TM4 ! TM4 ! TM4 ! TM4 !     !
                        !          ! TU3 ! TU2 ! TU1 ! TU0 !     !   8
                        !----------+-----+-----+-----+-----!     !
                        ! NOT USED ! TM5 ! TM5 ! TM5 ! TM5 !     !   #
                        !          ! TU3 ! TU2 ! TU1 ! TU0 !     !
                        !----------+-----+-----+-----+-----!     !
                        ! NOT USED ! TM6 ! TM6 ! TM6 ! TM6 !     !
                        !          ! TU3 ! TU2 ! TU1 ! TU0 !     !
                        !----------+-----+-----+-----+-----!     !
                        ! NOT USED ! TM7 ! TM7 ! TM7 ! TM7 !     !
                        !          ! TU3 ! TU2 ! TU1 ! TU0 !     !
                        !----------+-----+-----+-----+-----!     V
```

REMEMBER THAT DVID1 MUST BE SET UP FOR THE PROPER TM78'S TO BE TESTED. IF THE TM78 IN NOT SELECTED IN DVID1, THE INFORMATION FOR THAT TM78 IN THE UNIT TABLE IS IGNORED.

BIT 1 OF SR1 = 0

THE DEC/X11 DRIVER WILL SEARCH FOR AND TEST ALL TU78 SLAVES THAT RESPOND READY, ON-LINE AND WRITE ENABLED. IN THIS MODE THE CONTENTS OF DVID1 IS NOT USED.

10.0 DUAL PORT OPTION

THE TM78 PRODUCT WILL ALLOW TWO RH11/RH70 CONTROLLERS (PORTS A AND B) TO BE CONNECTED, AND EACH TU78 HAS A FRONT PANEL SWITCH THAT SPECIFIES THE PORTS THAT HAVE ACCESS TO THAT TU. THE CHOICES ARE A, B, A AND B, OR NONE.

THE A ONLY, B ONLY, OR NONE CONFIGURATIONS POSE NO PROBLEM TO THIS DRIVER, BUT THE A AND B CONFIGURATION IS A PROBLEM. THE PROBLEM EXISTS BECAUSE THE TM78 DOES NOT ARBITRATE PORT ACCESS, AND PROVIDES NO PATH TO ALLOW SOFTWARE TO DO IT.

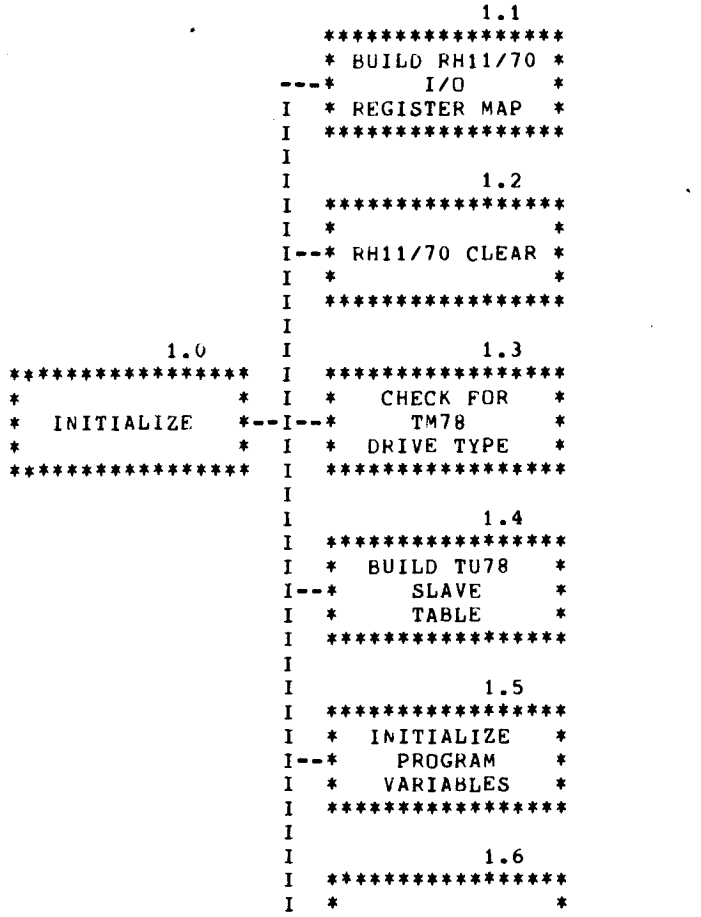
THEREFORE, THE BOTH MODE OF A TU78 IS ONLY USEABLE FOR WRITE ONLY UTILIZATION, OR FOR INTERLEAVED OPERATIONS SYNCHRONIZED THROUGH SOME EXTERNAL MECHANISM.

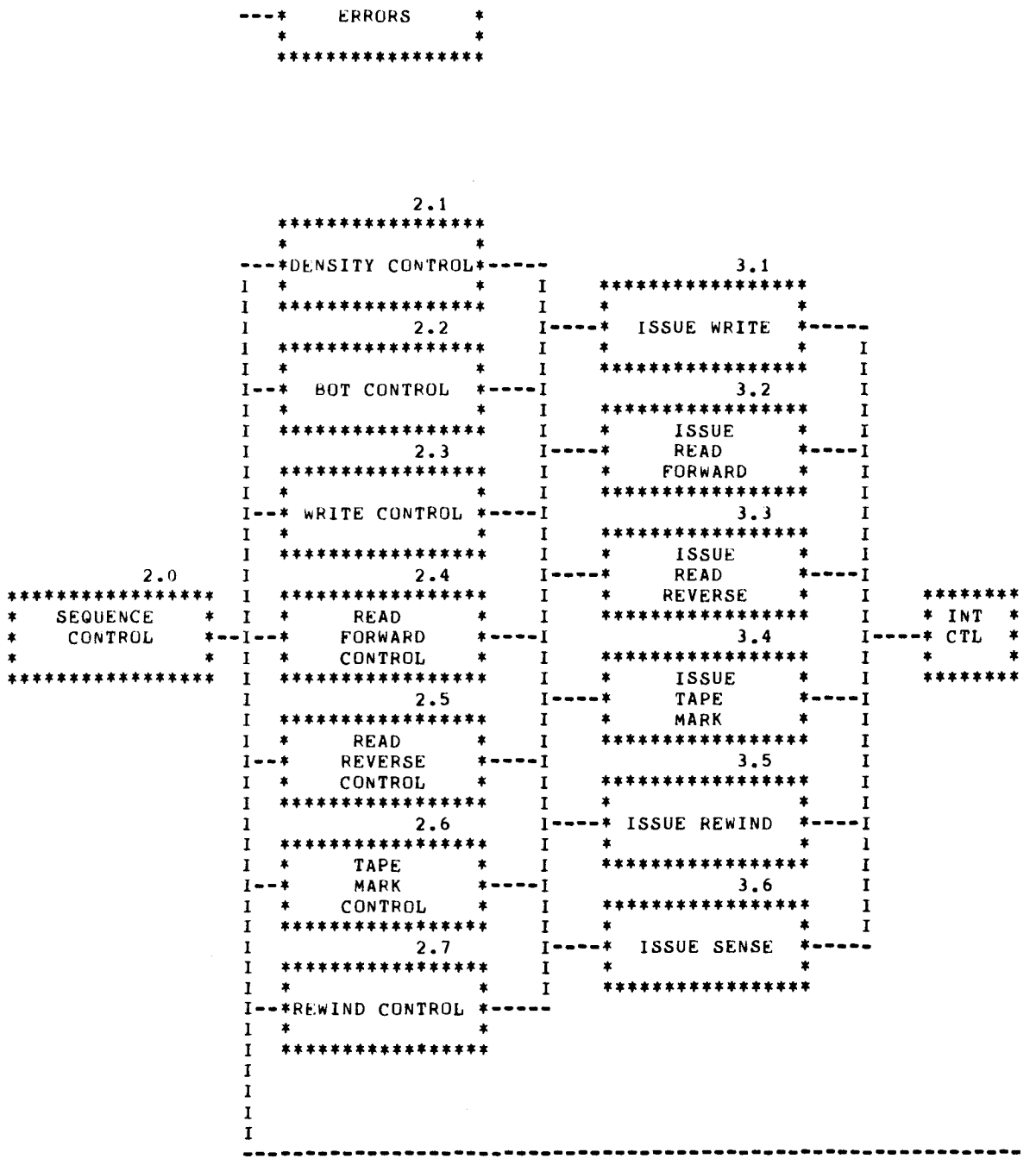
AS A RESULT DUAL MB ACCESS TO A SINGLE TU78 IS NOT SUPPORTED

BY THIS DRIVER, AND CONFIGURATION IN THAT MANNER WILL RESULT IN UNPREDICTABLE DRIVER OPERATION.

TO SUPPORT DUAL PORT ACCESS TO THE A ONLY, B ONLY TU78'S REQUIRES TWO COPIES OF THIS DRIVER TO BE ACTIVE ONE FOR EACH RH11/70 CONNECTED TO THE TM78.

11.0 HIERARCHY OF THE TM78 DEC/X11 DRIVER




ERRORS
2.1
DENSITY CONTROL
2.2
BOT CONTROL
2.3
WRITE CONTROL
2.4
READ FORWARD CONTROL
2.5
READ REVERSE CONTROL
2.6
TAPE MARK CONTROL
2.7
REWIND CONTROL
2.0
SEQUENCE CONTROL
3.1
ISSUE WRITE
3.2
ISSUE READ FORWARD
3.3
ISSUE READ REVERSE
3.4
ISSUE TAPE MARK
3.5
ISSUE REWIND
3.6
ISSUE SENSE
INT CTL

12.0 DATA STRUCTURES OF THE TM78 DEC/X11 DRIVER

	15 ... 0	
TMBIT		TM78 # CONVERTED TO A BIT POSITION
TMBIT2		TM78 # *2
TUBIT		TU78 # CONVERTED TO A BIT POSITION
TUBIT2		TU78 # *2
UCNT		UNIT COUNT
TM78		TM78 #
TU78		TU78 #
EOTCNT		EOT COUNT
TINTCD		TERMINATION INTERRUPT CODE
ITM78		INTERRUPTING TM78
ITU78		INTERRUPTING TU78
BOTCNT		# OF TU78'S AT BOT
SENSE		SENSE COMMAND INFORMATION
CNDTI		CLEAR ATTENTION INTERRUPT
ERRIDX		MODULE ERROR INDEX
A16AD		EXTENDED ADDRESS BITS 16 AND 17
PA18	*	LOWER 16 OF BUFFER ADDRESS
XMEM	*	EA BITS OF 18 BIT ADDRESS
PA22	*	LOWER 16 OF 22 BIT ADDRESS
EA22	*	EA BITS OF 22 BIT ADDRESS
FILE		FILE NUMBER
RECORD		RECORD NUMBER

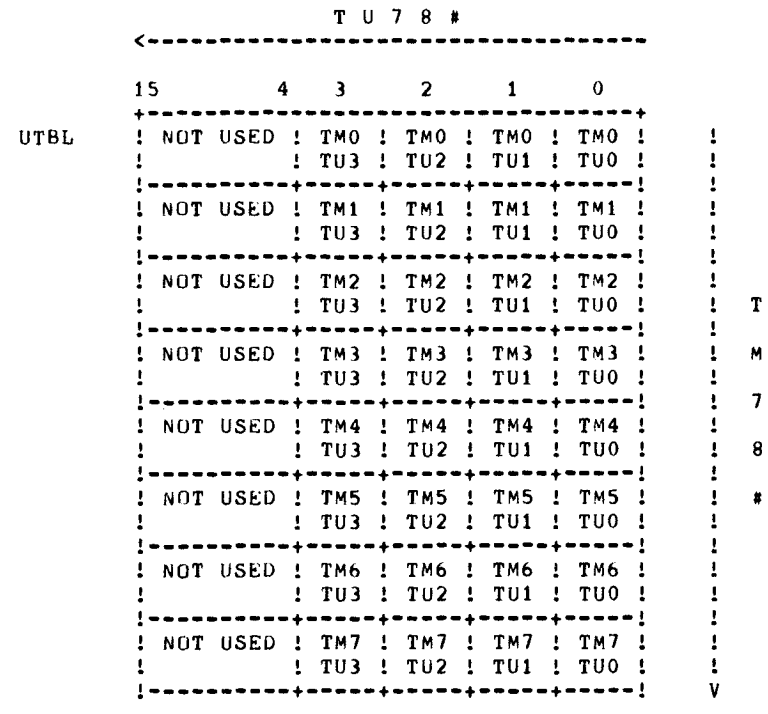
* = THESE LOCATIONS MUST BE IN THIS ORDER

"FLAGS"

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

```
804
805                     +-----------------------------------------------------------+
806                     !   !   !   !   !   !   !   !   !   !   !UNW!EXT!DAT!RTY!BOT!DEN!
807                     !   !   !   !   !   !   !   !   !   !   !INT!   !   !   !PAS!FLG!
808                     +-----------------------------------------------------------+
809                                                     ^   ^   ^   ^   ^   ^
810                                                     !   !   !   !   !   !
811                                                     !   !   !   !   !   --> DENSITY FLAG
812                                                     !   !   !   !   --> BOT PASS FLAG
813                                                     !   !   !   -->RETRY IN PROGRESS FLAG
814                                                     !   !   --> DATA TRANSFER FLAG
815                                                     !   --> EXIT FLAG
816                                                     --> INTERRUPT EXPECTED FLAG
817   "TMUNIT"
818            15  14  13  12  11  10   9   8   7   6   5   4   3   2   1   0
819           +-----------------------------------------------------------------+
820           !           NOT USED              !   !   !   !   !   !   !   !   !   !
821           !                                 !   !   !   !   !   !   !   !   !   !
822           +-----------------------------------------------------------------+
823                                             ^   ^   ^   ^   ^   ^   ^   ^
824                                             !   !   !   !   !   !   !   !
825                                             !   !   !   !   !   !   !   --> TM78 #0 SELECT
826                                             !   !   !   !   !   !   --> TM78 #1 SELECT
827                                             !   !   !   !   !   --> TM78 #2 SELECT
828                                             !   !   !   !   --> TM78 #3 SELECT
829                     0 = NOT SELECTED        !   !   !   --> TM78 #4 SELECT
830                     1 = SELECTED            !   !   --> TM78 #5 SELECT
831                                             !   --> TM78 #6 SELECT
832                                             --> TM78 #7 SELECT
833
834
835
836   "DVID1" 15  14  13  12  11  10   9   8   7   6   5   4   3   2   1   0
837           +-----------------------------------------------------------------+
838           !           NOT USED              !   !   !   !   !   !   !   !   !   !
839           !                                 !   !   !   !   !   !   !   !   !   !
840           +-----------------------------------------------------------------+
841                                             ^   ^   ^   ^   ^   ^   ^   ^
842                                             !   !   !   !   !   !   !   !
843                                             !   !   !   !   !   !   !   --> TM78 #0 SELECT
844                                             !   !   !   !   !   !   --> TM78 #1 SELECT
845                                             !   !   !   !   !   --> TM78 #2 SELECT
846                     0 = NOT SELECTED        !   !   !   !   --> TM78 #3 SELECT
847                     1 = SELECTED            !   !   !   --> TM78 #4 SELECT
848                                             !   !   --> TM78 #5 SELECT
849                                             !   --> TM78 #6 SELECT
850                                             --> TM78 #7 SELECT
851
852
853            15 ................. 7   6   5   4   3   2   1   0
854           +-----------------------------------------------------------+
855   "SR1"   !        NOT USED               !   !   !   !   !   !   !   !
856           !                               !   !   !   !   !   !   !   !
857           +-----------------------------------------------------------+
858                                           ^   ^   ^   ^   ^   ^
859                                           !   !   !   !   !   !
```

```
                                                    !   !   !   !   !   --> 0=ALLOW DATA COMPARE
                                                    !   !   !   !   !       1=INHIBIT DATA COMPARE
                                                    !   !   !   !   --> 0=TEST AVAIL. TU78'S
                                                    !   !   !   !       1=TEST USER SEL. TU78'S
                                                    !   !   !   --> 0=REPORT ERRORS
                                                    !   !   !       1=DO NOT REPORT ERRORS
                                                    !   !   --> 0=STATISTICS AT EOT
                                                    !   !       1=NO STATISTICS AT EOT
                                                    !   --> 0=TEST GCR
                                                    !       1=DO NOT TEST GCR
                                                    -->0=TEST PE
                                                       1=DO NOT TEST PE
                                        UNIT TABLE
```

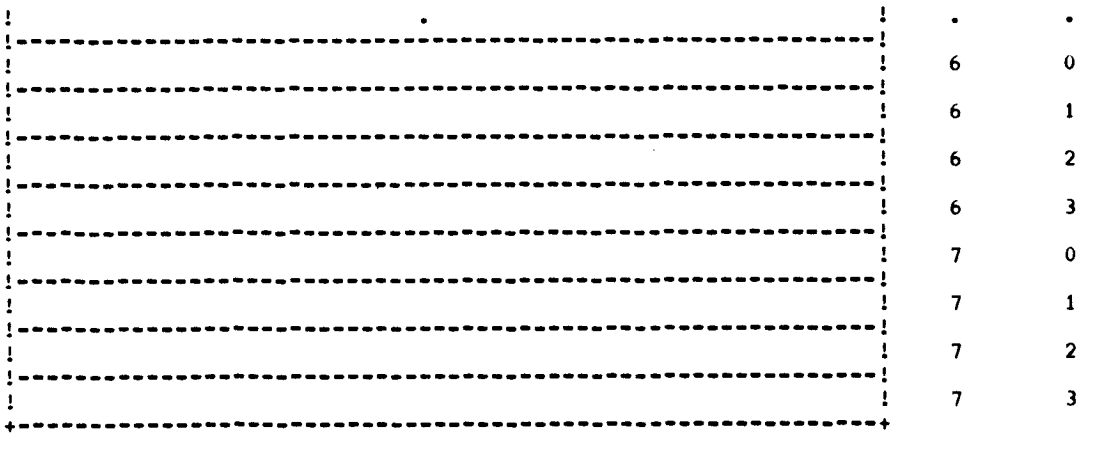



```
                                        0 = TU NOT SELECTED FOR TEST (USER SELECTED)
                                                        OR
                                            TU NOT ONLINE-WRITE ENABLE-AVAILABLE (PROGRAM SELECTED)
```

```
                        1 = TU SELECTED FOR TEST
                                    EOT TABLE




                                      T U 7 8 #
                                    <----------------------------------

                                    15        4   3     2     1     0
                                    +-----------------------------------+
                        EOTTBL      ! NOT USED ! TM0 ! TM0 ! TM0 ! TM0 !     !
                                    !          ! TU3 ! TU2 ! TU1 ! TU0 !     !
                                    !----------+-----+-----+-----+-----!     !
                                    ! NOT USED ! TM1 ! TM1 ! TM1 ! TM1 !     !
                                    !          ! TU3 ! TU2 ! TU1 ! TU0 !     !
                                    !----------+-----+-----+-----+-----!     !
                                    ! NOT USED ! TM2 ! TM2 ! TM2 ! TM2 !     !
                                    !          ! TU3 ! TU2 ! TU1 ! TU0 !     !   T
                                    !----------+-----+-----+-----+-----!     !
                                    ! NOT USED ! TM3 ! TM3 ! TM3 ! TM3 !     !   M
                                    !          ! TU3 ! TU2 ! TU1 ! TU0 !     !
                                    !----------+-----+-----+-----+-----!     !   7
                                    ! NOT USED ! TM4 ! TM4 ! TM4 ! TM4 !     !
                                    !          ! TU3 ! TU2 ! TU1 ! TU0 !     !   8
                                    !----------+-----+-----+-----+-----!     !
                                    ! NOT USED ! TM5 ! TM5 ! TM5 ! TM5 !     !   #
                                    !          ! TU3 ! TU2 ! TU1 ! TU0 !     !
                                    !----------+-----+-----+-----+-----!     !
                                    ! NOT USED ! TM6 ! TM6 ! TM6 ! TM6 !     !
                                    !          ! TU3 ! TU2 ! TU1 ! TU0 !     !
                                    !----------+-----+-----+-----+-----!     !
                                    ! NOT USED ! TM7 ! TM7 ! TM7 ! TM7 !     !
                                    !          ! TU3 ! TU2 ! TU1 ! TU0 !     !
                                    !----------+-----+-----+-----+-----!     V


                                    0 = TU NOT AT EOT

                                    1 = TU AT EOT
                                                SCRATCH UNIT TABLE
```

```
                                T U 7 8 #
                        <-----------------------------------
                        15        4   3     2     1     0
                        +-----------------------------------+
SUTBL                   ! NOT USED ! TM0 ! TM0 ! TM0 ! TM0 !   !
                        !          ! TU3 ! TU2 ! TU1 ! TU0 !   !
                        !----------+-----+-----+-----+-----!   !
                        ! NOT USED ! TM1 ! TM1 ! TM1 ! TM1 !   !
                        !          ! TU3 ! TU2 ! TU1 ! TU0 !   !
                        !----------+-----+-----+-----+-----!   !
                        ! NOT USED ! TM2 ! TM2 ! TM2 ! TM2 !   !
                        !          ! TU3 ! TU2 ! TU1 ! TU0 !   !  T
                        !----------+-----+-----+-----+-----!   !
                        ! NOT USED ! TM3 ! TM3 ! TM3 ! TM3 !   !  M
                        !          ! TU3 ! TU2 ! TU1 ! TU0 !   !
                        !----------+-----+-----+-----+-----!   !  7
                        ! NOT USED ! TM4 ! TM4 ! TM4 ! TM4 !   !
                        !          ! TU3 ! TU2 ! TU1 ! TU0 !   !  8
                        !----------+-----+-----+-----+-----!   !
                        ! NOT USED ! TM5 ! TM5 ! TM5 ! TM5 !   !  #
                        !          ! TU3 ! TU2 ! TU1 ! TU0 !   !
                        !----------+-----+-----+-----+-----!   !
                        ! NOT USED ! TM6 ! TM6 ! TM6 ! TM6 !   !
                        !          ! TU3 ! TU2 ! TU1 ! TU0 !   !
                        !----------+-----+-----+-----+-----!   !
                        ! NOT USED ! TM7 ! TM7 ! TM7 ! TM7 !   !
                        !          ! TU3 ! TU2 ! TU1 ! TU0 !   !
                        !----------+-----+-----+-----+-----!   V


                        0 = TU NOT BEING TESTED OR DROPPED

                        1 = TU BEING TESTED



                        UNRECOVERABLE READ ERROR STATISTIC TABLE
                "URREAD"

                          15                                                  00    TM78#   TU78#
                        +-------------------------------------------------------+   -----   -----
                        !                                                       !     0       0
                        !-------------------------------------------------------!
                        !                                                       !     0       1
                        !-------------------------------------------------------!
                        !                                                       !     0       2
                        !-------------------------------------------------------!
                        !                                                       !     0       3
                        !-------------------------------------------------------!
                        !                                                       !     1       0
                        !-------------------------------------------------------!
                        !                                                       !     1       1
                        !-------------------------------------------------------!
                        !                                                       !     1       2
```

```
1028                    !-----------------------------------------------------!
1029                    !                                                     !   1       3
1030                    !-----------------------------------------------------!   .       .
1031                    !                     .                               !   .       .
1032                    !                     .                               !   .       .
1033                    !                     .                               !   .       .
1034                    !                     .                               !   .       .
1035                    !-----------------------------------------------------!
1036                    !                                                     !   6       0
1037                    !-----------------------------------------------------!
1038                    !                                                     !   6       1
1039                    !-----------------------------------------------------!
1040                    !                                                     !   6       2
1041                    !-----------------------------------------------------!
1042                    !                                                     !   6       3
1043                    !-----------------------------------------------------!
1044                    !                                                     !   7       0
1045                    !-----------------------------------------------------!
1046                    !                                                     !   7       1
1047                    !-----------------------------------------------------!
1048                    !                                                     !   7       2
1049                    !-----------------------------------------------------!
1050                    !                                                     !   7       3
1051                    +-----------------------------------------------------+
1052
1053
1054                                INDEX = ((TM78#(8))+(TU78#(2)))
1055
1056
1057
1058
1059
1060                            RECOVERABLE READ ERROR STATISTIC TABLE
1061                    "RREAD"
1062
1063                      15                                            00    TM78#   TU78#
1064                    +-----------------------------------------------------+ -----   -----
1065                    !                                                     !   0       0
1066                    !-----------------------------------------------------!
1067                    !                                                     !   0       1
1068                    !-----------------------------------------------------!
1069                    !                                                     !   0       2
1070                    !-----------------------------------------------------!
1071                    !                                                     !   0       3
1072                    !-----------------------------------------------------!
1073                    !                                                     !   1       0
1074                    !-----------------------------------------------------!
1075                    !                                                     !   1       1
1076                    !-----------------------------------------------------!
1077                    !                                                     !   1       2
1078                    !-----------------------------------------------------!
1079                    !                                                     !   1       3
1080                    !-----------------------------------------------------!   .       .
1081                    !                     .                               !   .       .
1082                    !                     .                               !   .       .
1083                    !                     .                               !   .       .
```

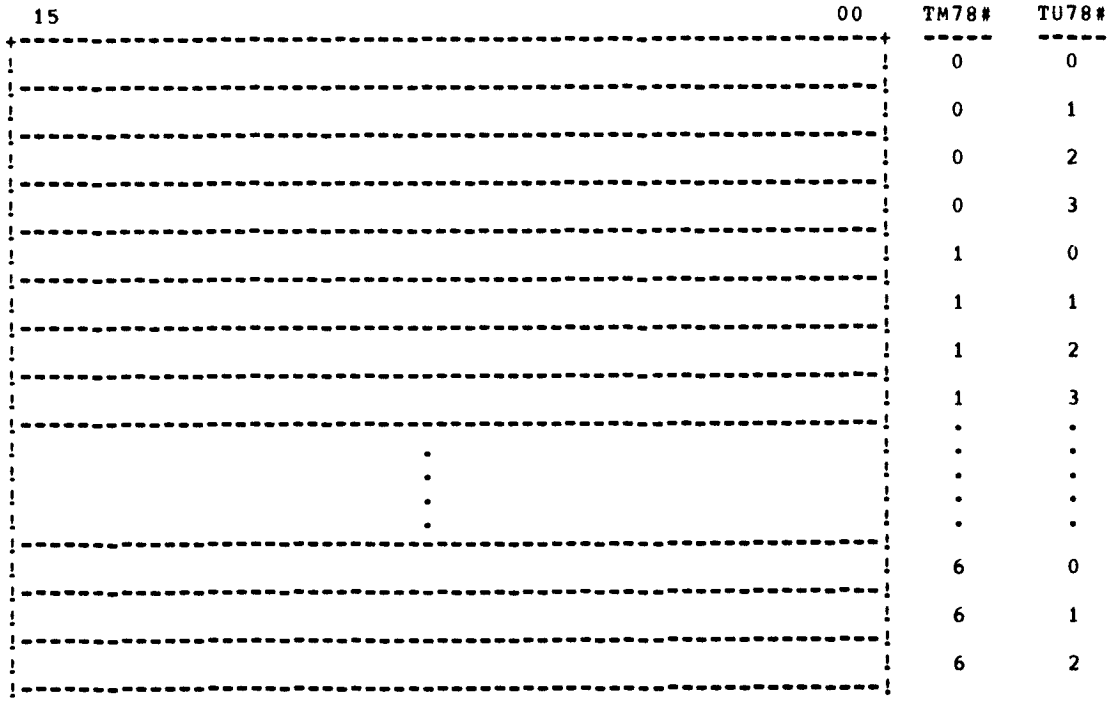


.	.
6	0
6	1
6	2
6	3
7	0
7	1
7	2
7	3

INDEX = ((TM78#(8))+(TU78#(2)))

RECOVERABLE WRITE ERROR STATISTIC TABLE

"RWRIT"



15 ... 00	TM78#	TU78#
	0	0
	0	1
	0	2
	0	3
	1	0
	1	1
	1	2
	1	3
	.	.
.	.	.
.	.	.
.	.	.
	.	.
	6	0
	6	1
	6	2

```
1140                    !                                                       !   6       3
1141                    !-------------------------------------------------------!
1142                    !                                                       !   7       0
1143                    !-------------------------------------------------------!
1144                    !                                                       !   7       1
1145                    !-------------------------------------------------------!
1146                    !                                                       !   7       2
1147                    !-------------------------------------------------------!
1148                    !                                                       !   7       3
1149                    +-------------------------------------------------------+
1150
1151
1152                                    INDEX = ((TM78#(8))+(TU78#(2)))
1153                    -
```

```
                                        IOMODX  <TMDA >,172400,224,5,0,0,1,,0,BUFIN,512.,512.
                                        MODULE  150000,TMDA ,172400,224,5,0,0,1,,0,BUFIN,512.,512.
                                        .TITLE  TMDA DEC/X11 SYSTEM EXERCISER MODULE
                                ;       DDXCOM  VERSION  6      23-MAY-78
                                        .LIST   BIN
                                ;*****************************************************************************
                                BEGIN:
000000' 046524  040504      040 MODNAM: .ASCII  /TMDA / ;MODULE NAME.
000005'    000                  XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 172400                  ADDR:   172400+0                ;1ST DEVICE ADDR.
000010' 000224                  VECTOR: 224+0                   ;1ST DEVICE VECTOR.
000012'    240                  BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013'    000                  BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014' 000001                  DVID1:  0+1                     ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN                    ;SWITCH REGISTER 4
                                ;*****************************************************************************
000026' 150000                  STAT:   150000                  ;STATUS WORD.
000030' 000252'                 INIT:   START                   ;MODULE START ADDR.
000032' 000252'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                       ;PASS COUNTER.
000036' 000001                  ICONT:  1.                      ;# OF ITERATIONS PER PASS=1.
000040' 000000                  ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042' 000000                  SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                  HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                  SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                  HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                  SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054' 000000                  RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:                         ;RESERVED FOR MONITOR USE
000056' 000000                  RES1:   0                       ;RESERVED FOR MONITOR USE
000060' 000000                  RES2:   .0                      ;RESERVED FOR MONITOR USE
000062' 000000                  SVR0:   OPEN                    ;LOC TO SAVE R0.
000064' 000000                  SVR1:   OPEN                    ;LOC TO SAVE R1.
000066' 000000                  SVR2:   OPEN                    ;LOC TO SAVE R2.
000070' 000000                  SVR3:   OPEN                    ;LOC TO SAVE R3.
000072' 000000                  SVR4:   OPEN                    ;LOC TO SAVE R4.
000074' 000000                  SVR5:   OPEN                    ;LOC TO SAVE R5.
000076' 000000                  SVR6:   OPEN                    ;LOC TO SAVE R6.
000100' 000000                  CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102'                         SBADR:                          ;ADDR OF GOOD DATA, OR
000102' 000000                  ACSR:   OPEN                    ;CONTENTS OF CSR.
000104'                         WASADR:                         ;ADDR OF BAD DATA, OR
000104' 000000                  ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106'                         ERRTYP:                         ;TYPE OF ERROR
000106' 000000                  ASB:    OPEN                    ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN                    ;ACTUAL DATA.
000112' 001566'                 RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122' 000000                  IDNUM:  0                       ;MODULE IDENTIFICATION NUMBER=0
000124' 011174'                 RBUFVA: BUFIN                   ;READ BUFFER VIRTUAL ADDRESS
000126' 000000                  RBUFPA: OPEN                    ;READ BUFFER PHYSICAL ADDRESS
```



```
                                RBUFEA: OPEN                    ;READ BUFFER EA BITS
                                RBUFSZ: 512.                    ;SIZE OF THE READ BUFFER
                                WBUFPA: OPEN                    ;WRITE BUFFER PHYSICAL ADDRESS
                                WBUFEA: OPEN                    ;WRITE BUFFER EA BITS
                                WBUFRQ: 512.                    ;WRITE BUFFER SIZE REQUESTED
                                WBUFSZ: OPEN                    ;WRITE BUFFER SIZE AVAILABLE
                                CDERCT: OPEN                    ;CDATA/DATCK ERROR COUNT
                                CDWDCT: OPEN                    ;CDATA/DATCK WORD COUNT
                                FREE:   OPEN                    ;RESERVED FOR FUTURE USE
                                        .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                        .NLIST
                                        .WORD   0
                                        .LIST
                                        .ENDR
000252'                         MODSP:
                                ;*****************************************************************************
                                .REM    -
                                13.0    PDL FOR THE TM78 DEC/X11 DRIVER
                                        ------------------------------


                                BEGINROUTINE    MODULE 1.0 (DRIVER INITIALIZATION)
                                :  CLEAR "FLAGS"
                                :  CALL MODULE 1.1 (BUILD I/O REGISTER MAP)
                                :  CALL MODULE 1.2 (RH11/70 INITIALIZE)
                                :  CALL MODULE 1.3 (CHECK TM78 DRIVE TYPE)
                                :  IF TMUNIT = 0
                                :  :  THEN-CLEAR RH11/70 REGISTER CS1
                                :  :       PRINT ERROR 1
                                :  :       DROP THIS DEC/X11 DRIVER (DEC/X11 ENDMOD)
                                :  :  ELSE-CONTINUE
                                :  ENDIF
                                :  CALL MODULE 1.4 (BUILD TU78 UNIT TABLE)
                                :  IF UNIT COUNT = 0
                                :  :  THEN-CLEAR RH11/70 REGISTER CS1
                                :  :       PRINT ERROR 11
                                :  :       DROP THIS DEC/X11 DRIVER (DEC/X11 ENDMOD)
                                :  :  ELSE-CONTINUE
                                :  ENDIF
                                :  CALL MODULE 1.5 (INITIALIZE PROGRAM DATA STRUCTURES)
                                :  IF "SR1" BIT4 AND BIT5=1 (NO DENSITY SELECTED)
                                :  :  THEN-CLEAR RH11/70 REGISTER CS1
                                :  :       PRINT ERROR 12
                                :  :       DROP THIS DEC/X11 DRIVER (DEC/X11 ENDMOD)
                                :  :  ELSE-CONTINUE
                                :  ENDIF
                                ENDROUTINE

                                -

000252' 005067  010216          START:  CLR     FLAGS           ;CLEAR THE FLAGS
000256' 004767  000144                  CALL    M1.1            ;CALL MODULE 1.1 (BUILD I/O REGISTER MAP)
000262' 004767  000174                  CALL    M1.2            ;CALL MODULE 1.2 (RH11/70 INITIALIZE)
000266' 004767  000200                  CALL    M1.3            ;CALL MODULE 1.3 (CHECK TM78 DRIVE TYPE)
000272' 005767  010200                  TST     TMUNIT          ;IF TMUNIT=0
000276' 001011                          BNE     1$              ;ELSE-CONTINUE
```

```
1196  000300' 005077  010570                  CLR     @CS1            ;THEN-CLEAR TH/70 REGISTER CS1
1197  000304' 012767  000001  010146          MOV     #1,ERRINX       ;PRINT ERROR 1
1198  000312' 004767  006316                  CALL    M4.1
1199  000316' 104410  000000'                 ENDS,BEGIN              ;
1200  000322' 004767  000366          1$:     CALL    M1.4            ;CALL MODULE 1.4 (BUILD TU78 UNIT TABLE)
1201  000326' 005767  010106                  TST     UCNT            ;IF UNIT COUNT=0
1202  000332' 001011                          BNE     2$              ;ELSE-CONTINUE
1203  000334' 005077  010534                  CLR     @CS1            ;THEN-CLEAR RH11/70 REGISTER CS1
1204  000340' 012767  000011  010112          MOV     #11,ERRINX      ;PRINT ERROR 11
1205  000346' 004767  006262                  CALL    M4.1
1206  000352' 104410  000000'                 ENDS,BEGIN              ;
1207  000356' 004767  000762          2$:     CALL    M1.5            ;CALL MODULE 1.5 (INITIALIZE DATA STRUCTURES)
1208  000362' 016701  177430                  MOV     SR1,R1          ;
1209  000366' 042701  177717                  BIC     #-1-BIT4-BIT5,R1  ;
1210  000372' 022701  000060                  CMP     #BIT4+BIT5,R1   ;IF NO DENSITY SELECTED
1211  000376' 001011                          BNE     3$              ;ELSE-CONTINUE
1212  000400' 005077  010470                  CLR     @CS1            ;THEN-CLEAR RH11/70 REGISTER CS1
1213  000404' 012767  000012  010046          MOV     #12,ERRINX      ;PRINT ERROR 12
1214  000412' 004767  006216                  CALL    M4.1
1215  000416' 104410  000000'                 ENDS,BEGIN              ;
1216  000422' 000167  001142          3$:     JMP     M2.0            ;DEC/X11 RESTART CODE
```



```
1218                                    .REM    -
1219                                    BEGINROUTINE    MODULE 1.1 (BUILD I/O REGISTER MAP)
1220                                    :  GET DEC/X11 HEADER WORD "ADDR"
1221                                    :  CLEAR LOOP COUNT
1222                                    :  BGNDO
1223                                    :  :  STORE ("ADDR"+LOOP COUNT) AT (RHTBL+LOOP COUNT)
1224                                    :  :  ACCESS THE UNIBUS ADDRESS ("ADDR"+LOOP COUNT)
1225                                    :  :  ADD 2 TO LOOP COUNT
1226                                    :  :  DO UNTIL LOOP COUNT = 100(8)
1227                                    :  ENDDO
1228                                    ENDROUTINE
1229
1230                                    -
1231
1232  000426' 016701  177354          M1.1:   MOV     ADDR,R1         ;GET DEC/X11 HEADER WORD "ADDR"
1233  000432' 005711                          TST     (R1)            ;CATCH WRONG DEVICE ADDRESS EARLY
1234  000434' 005002                          CLR     R2              ;CLEAR THE LOOP COUNT
1235  000436' 010162  011074'          1$:     MOV     R1,RHTBL(R2)    ;STORE ("ADDR"+LOOP COUNT) AT (RHTBL+LOOP COUNT)
1236  000442' 062701  000002                  ADD     #2,R1           ;NEXT REGISTER
1237  000446' 062702  000002                  ADD     #2,R2           ;ADD 2 TO THE LOOP COUNT
1238  000452' 022702  000100                  CMP     #100,R2         ;DO UNTIL THE LOOP COUNT=100
1239  000456' 001367                          BNE     1$              ;
1240  000460' 000207                          RTS     PC              ;ENDROUTINE
```

```
                                        .REM    -
                                        BEGINROUTINE    MODULE 1.2 (RH11/70 INITIALIZE)
                                        :  SET THE "CLR" BIT IN RH11/70 CS2 (000040)
                                        ENDROUTINE

                                        -

000462' 052777  000040  010414  M1.2:   BIS     #BIT5,@CS2      ;SET THE CLEAR BIT IN RH11/70 CS2
000470' 000207                          RTS     PC              ;ENDROUTINE
                                        .REM    -
                                        BEGINROUTINE    MODULE 1.3 (CHECK TM78 DRIVE TYPE)
                                        :  CLEAR THE TM78#
                                        :  IF BIT1 OF SR1 = 1 (USER SELECTED TU78'S)
                                        :  :  THEN - COPY "DVID1" TO "TMUNIT"
                                        :  :  ELSE - SET "TMUNIT" = 377(8)
                                        :  ENDIF
                                        :  BGNDO
                                        :  :  IF BIT#(TM78#) OF "TMUNIT" = 0
                                        :  :  :  THEN-CONTINUE
                                        :  :  :  ELSE-LOAD RH11/70 CS2 REG WITH TM78#
                                        :  :  :        SET TRE IN RH11/70 IN REG CS1
                                        :  :  :        CLEAR REGISTER CS1
                                        :  :  :        IF NED BIT IN RH11/70 REGISTER CS2 = 0
                                        :  :  :        :  THEN - IF BITS 8:0 OF RH11/70 REGISTER 26 = 101(8)
                                        :  :  :        :         :  THEN-CONTINUE
                                        :  :  :        :         :  ELSE-CLEAR BIT#(TM78#) OF "TMUNIT"
                                        :  :  :        :         ENDIF
                                        :  :  :        :  ELSE - CLEAR BIT#(TM78#) OF "TMUNIT"
                                        :  :  :        ENDIF
                                        :  :  ENDIF
                                        :  :  INCREMENT THE TM78#
                                        :  :  DO UNTIL THE TM78# = 10(8)
                                        :  ENDDO
                                        ENDROUTINE

                                        -

000472' 005067  007744          M1.3:   CLR     TM78            ;CLEAR THE TM78#
000476' 012767  000377  007772          MOV     #377,TMUNIT     ;SELECT ALL TM78'S
000504' 032767  000002  177304          BIT     #BIT1,SR1       ;IF SR1 BIT1 = 1 (USER SELECTED TU78'S)
000512' 001403                          BEQ     1$              ;ELSE - CONTINUE
000514' 016767  177274  007754          MOV     DVID1,TMUNIT    ;THEN - COPY DEC/X11 VARIABLE "DVID1" TO "TMUNIT"
000522' 004767  000734          1$:     CALL    TMCONV
000526' 036767  007676  007742          BIT     TMBIT,TMUNIT    ;IF BIT#(TM78#) OF "TMUNIT"=0
000534' 001433                          BEQ     3$              ;THEN-CONTINUE
000536' 016777  007700  010340          MOV     TM78,@CS2       ;SELECT UNIT #
000544' 000240                          NOP                     ;WAIT A BIT
000546' 012777  040000  010320          MOV     #BIT14,@CS1     ;SET BIT TRE IN RH11/70 REGISTER CS1
000554' 005077  010314                  CLR     @CS1            ;CLEAR BIT TRE IN RH11/70 REGISTER CS1
000560' 032777  010000  010316          BIT     #BIT12,@CS2     ;IF BIT NED IN RH11/70 REGISTER CS2 = 0
000566' 001404                          BEQ     2$              ;THEN - GO TO THEN
000570' 046767  007634  007700          BIC     TMBIT,TMUNIT    ;ELSE - CLEAR BIT#(TM78#) OF "TMUNIT"
000576' 000412                          BR      3$              ;CONTINUE
000600' 017701  010316          2$:     MOV     @DT,R1          ;IF BITS 8:0 OF RH11/70 REGISTER26=101
000604' 042701  177000                  BIC     #177000,R1
000610' 022701  000101                  CMP     #101,R1
```



```
000614' 001412                          BEQ     4$              ;THEN-CONTINUE
000616' 046767  007606  007652          BIC     TMBIT,TMUNIT    ;ELSE-CLEAR BIT#(TM78#) OF "TMUNIT"
000624' 005267  007612          3$:     INC     TM78            ;INCREMENT THE TM78#
000630' 022767  000010  007604          CMP     #10,TM78        ;DO UNTIL THE TM78#=10
000636' 001331                          BNE     1$
000640' 000424                          BR      6$              ;EXIT
000642' 012704  000500          4$:     MOV     #500,R4         ;SET LOOP COUNTER
000646' 005777  010274          5$:     TST     @ST78           ;TM READY ?
000652' 100764                          BMI     3$              ;YES , CHECK NEXT TM78
000654' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
 (1) 000660' 104407  000000'            BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
000664' 005304                          DEC     R4              ;COUNT DOWN
000666' 001367                          BNE     5$              ;BR IF NOT TIMED OUT
000670' 012767  000013  007562          MOV     #13,ERRINX      ;SET ERROR #13
000676' 004767  005732                  CALL    M4.1            ;GO TO ERROR TYPE ROUTINE
000702' 046767  007522  007566          BIC     TMBIT,TMUNIT    ;DROP THIS TM78
000710' 000745                          BR      3$              ;CHECK NEXT TM78
000712' 000207          6$:     RTS     PC              ;ENDROUTINE
```

```
                                .REM    -
                                BEGINROUTINE    MODULE 1.4 (BUILD TU78 UNIT TABLE)
                                :  SET UP INTERRUPT VECTOR (MODULE 4.0) AND BR LEVEL
                                :  IF SR1 BIT1=1 (USER SELECTED TU78'S)
                                :  :  THEN-COPY USER TABLE "UTBL" TO SCRATCH UNIT TABLE "SUTBL"
                                :  :       CLEAR TM78# "TM78"
                                :  :       CLEAR UNIT COUNT "UCNT"
                                :  :       BGNDO
                                :  :       :  IF BIT#(TM78#) OF "TMUNIT" = 0
                                :  :       :  :  THEN-CONTINUE
                                :  :       :  :  ELSE-LOAD RH11/70 CS2 REGISTER WITH TM78#
                                :  :       :  :       CLEAR THE TU78#
                                :  :       :  :       BGNDO
                                :  :       :  :       :  IF SCRATCH UNIT TABLE (TM78#,TU78#) = 1
                                :  :       :  :       :  :  THEN-CALL MODULE 3.6 (SENSE)
                                :  :       :  :       :  :       IF TERM. INT. CODE=DONE
                                :  :       :  :       :  :       :  THEN-CONTINUE
                                :  :       :  :       :  :       :  ELSE-CLEAR THE TU78 STATUS
                                :  :       :  :       :  :       ENDIF
                                :  :       :  :       :  :       IF TU78 IS RDY.PRES.ONL.AVAIL AND NOT FPT
                                :  :       :  :       :  :       :  THEN-INCREMENT UNIT COUNT
                                :  :       :  :       :  :       :  ELSE-DROP TU78 - CLEAR BIT (SCRATCH UNIT TABLE(TM78#
                                :  :       :  :       :  :       ENDIF
                                :  :       :  :       :  :  ELSE-CONTINUE
                                :  :       :  :       :  ENDIF
                                :  :       :  :       :  INCREMENT THE TU78#
                                :  :       :  :       :  DO UNTIL THE TU78# = 4
                                :  :       :  :       ENDDO
                                :  :       :  ENDIF
                                :  :       :  INCREMENT THE TM78#
                                :  :       :  DO UNTIL THE TM78# = 10(8)
                                :  :       ENDDO
                                :  :  ELSE-CLEAR THE TM78#
                                :  :       CLEAR THE UNIT COUNT
                                :  :       BGNDO
                                :  :       :  IF BIT#(TM78#) OF TMUNIT = 0
                                :  :       :  :  THEN - CLEAR (SCRATCH UNIT TABLE(TM78#(2)))
                                :  :       :  :         CLEAR (UNIT TABLE(TM78#(2))
                                :  :       :  :  ELSE - LOAD THE RH11/70 CS2 REGISTER WITH TM78#
                                :  :       :  :         CLEAR THE TU78#
                                :  :       :  :         BGNDO
                                :  :       :  :         :  CALL MODULE 3.6 (SENSE)
                                :  :       :  :         :  IF TERM. INT. CODE=DONE
                                :  :       :  :         :  :  THEN-CONTINUE
                                :  :       :  :         :  :  ELSE-CLEAR THE TU78 STATUS
                                :  :       :  :         :  ENDIF
                                :  :       :  :         :  IF TU78 IS RDY.PRES.ONL.AVAIL AND NOT FPT
                                :  :       :  :         :  :  THEN-SET BIT (SCRATCH UNIT TABLE(TM78#,TU78#))
                                :  :       :  :         :  :       SET BIT (UNIT TABLE(TM78#, TU78#))
                                :  :       :  :         :  :       INCREMENT UNIT COUNT
                                :  :       :  :         :  :  ELSE-CONTINUE
                                :  :       :  :         :  ENDIF
                                :  :       :  :         :  INCREMENT THE TU78#
                                :  :       :  :         :  DO UNTIL THE TU78# = 4
                                :  :       :  :         ENDDO
                                :  :       :  ENDIF
```



```
                                :  :       :  INCREMENT THE TM78#
                                :  :       :  DO UNTIL THE TM78# = 10(8)
                                :  :       ENDDO
                                :  ENDIF
                                ENDROUTINE
                                -
000714' 016700  177070          M1.4:   MOV     VECTOR,R0       ;SET UP INTERRUPT VECTOR
000720' 012720  006314'                 MOV     #M4.0,(R0)+
000724' 116720  177062                  MOVB    BR1,(R0)+       ;SETUP BR LEVEL
000730' 105010                          CLRB    (R0)
000732' 032767  000002  177056          BIT     #BIT1,SR1       ;IF SR1 BIT1=1
000740' 001504                          BEQ     M1.4X           ;ELSE-GO TO ELSE CODE
000742' 005001                          CLR     R1              ;SET LOOP COUNT
000744' 016161  010514' 010554' 1$:     MOV     UTBL(R1),SUTBL(R1)  ;COPY UNIT TABLE TO SCRATCH UNIT TABLE
000752' 062701  000002                  ADD     #2,R1
000756' 022701  000020                  CMP     #20,R1
000762' 001370                          BNE     1$
000764' 005067  007452                  CLR     TM78            ;CLEAR THE TM78#
000770' 005067  007444                  CLR     UCNT            ;CLEAR THE UNIT COUNT
000774' 004767  000462          2$:     CALL    TMCONV
001000' 036767  007424  007470          BIT     TMBIT,TMUNIT    ;IF BIT#(TM78#) OF "TMUNIT"=0
001006' 001452                          BEQ     7$              ;THEN-CONTINUE
001010' 016777  007426  010066          MOV     TM78,@CS2       ;ELSE-LOAD RH11/70 CS2 REGISTER WITH TM78#
001016' 005067  007422                  CLR     TU78            ;CLEAR THE TU78#
001022' 004767  000476          3$:     CALL    TUCONV
001026' 016701  007400                  MOV     TMBIT2,R1
001032' 036761  007376  010554'         BIT     TUBIT,SUTBL(R1) ;IF SCRATCH UNIT TABLE (TM78#,TU78#)=1
001040' 001427                          BEQ     6$              ;ELSE-CONTINUE
001042' 004767  005164                  CALL    M3.6            ;CALL MODULE 3.6 (SENSE)
001046' 022767  000001  007374          CMP     #DONE,TINTCD    ;IF SENSE TERMINATION INTERRUPT CODE=DONE
001054' 001402                          BEQ     4$              ;THEN-CONTINUE
001056' 005067  007374                  CLR     SENSE           ;ELSE-CLEAR SENSE
001062' 016701  007370          4$:     MOV     SENSE,R1
001066' 042701  017177                  BIC     #017177,R1
001072' 022701  160200                  CMP     #160200,R1      ;IF TU78 IS RDY.PRES.ONL.AVAIL AND NOT FPT
001076' 001406                          BEQ     5$              ;THEN-GO DO THEN
001100' 016701  007326                  MOV     TMBIT2,R1       ;ELSE-DROP TU78
001104' 046761  007324  010554'         BIC     TUBIT,SUTBL(R1) ;CLEAR BIT (SCRATCH UNIT TABLE(TM78#,TU78#))
001112' 000402                          BR      6$
001114' 005267  007320          5$:     INC     UCNT            ;INCREMENT THE UNIT COUNT
001120' 005267  007320          6$:     INC     TU78            ;INCREMENT THE TU78#
001124' 022767  000004  007312          CMP     #4,TU78         ;DO UNTIL THE TU78#=4
001132' 001333                          BNE     3$
001134' 005267  007302          7$:     INC     TM78            ;INCREMENT THE TM78#
001140' 022767  000010  007274          CMP     #10,TM78        ;DO UNTIL THE TM78#=10
001146' 001312                          BNE     2$
001150' 000474                          BR      M1.4OT
001152' 005067  007264          M1.4X:  CLR     TM78            ;CLEAR THE TM78#
001156' 005067  007256                  CLR     UCNT            ;CLEAR THE UNIT COUNT
001162' 004767  000274          1$:     CALL    TMCONV
001166' 016701  007240                  MOV     TMBIT2,R1
001172' 005061  010554'                 CLR     SUTBL(R1)       ;CLEAR THE SCRATCH UNIT TABLE FOR THIS TM78
001176' 005061  010514'                 CLR     UTBL(R1)        ;CLEAR THE UNIT TABLE FOR THIS TM78
001202' 036767  007222  007266          BIT     TMBIT,TMUNIT    ;IF BIT#(TM78#) OF TMUNIT = 0
001210' 001446                          BEQ     4$              ;THEN - CONTINUE
```

```
1428  001212' 016777  007224  007664          MOV     TM78,@CS2       ;ELSE - LOAD THE RH11/70 CS2 REGISTER WITH TM78#
1429  001220' 005067  007220                  CLR     TU78            ;CLEAR THE TU78#
1430  001224' 004767  000274          2$:     CALL    TUCONV          ;
1431  001230' 004767  004776                  CALL    M3.6            ;CALL MODULE 3.6 (SENSE)
1432  001234' 022767  000001  007206          CMP     #DONE.TINTCD    ;IF SENSE TERMINATION INTERRUPT CODE=DONE
1433  001242' 001402                          BEQ     5$              ;THEN-CONTINUE
1434  001244' 005067  007206                  CLR     SENSE           ;ELSE-CLEAR SENSE
1435  001250' 016701  007202          5$:     MOV     SENSE,R1
1436  001254' 042701  017177                  BIC     #017177,R1
1437  001260' 022701  160200                  CMP     #160200,R1      ;IF TU78 IS RDY.PRES.ONL.AVAIL AND NOT FPT
1438  001264' 001012                          BNE     3$              ;ELSE-CONTINUE
1439  001266' 016701  007140                  MOV     TMBIT2,R1
1440  001272' 056761  007136  010554'         BIS     TUBIT,SUTBL(R1) ;SET BIT SCRATCH UNIT TABLE(TM78#,TU78#)
1441  001300' 056761  007130  010514'         BIS     TUBIT,UTBL(R1)  ;SET BIT UNIT TABLE(TM78#,TU78#)
1442  001306' 005267  007126                  INC     UCNT            ;INCREMENT THE UNIT COUNT
1443  001312' 005267  007126          3$:     INC     TU78            ;INCREMENT THE TU78#
1444  001316' 022767  000004  007120          CMP     #4,TU78         ;DO UNTIL THE TU78#=4
1445  001324' 001337                          BNE     2$              ;
1446  001326' 005267  007110          4$:     INC     TM78            ;INCREMENT THE TM78#
1447  001332' 022767  000010  007102          CMP     #10,TM78        ;DO UNTIL THE TM78#=10
1448  001340' 001310                          BNE     1$
1449  001342' 000207                  M1.40T: RTS     PC              ;RETURN TO USER.
```



```
1451                                    .REM    -
1452                                    BEGINROUTINE    MODULE 1.5 (INITIALIZE PROGRAM DATA STRUCTURES)
1453                                    :  CLEAR THE RECOVERABLE WRITE STATISTIC TABLE
1454                                    :  CLEAR THE RECOVERABLE READ STATISTIC TABLE
1455                                    :  CLEAR THE UNRECOVERABLE READ STATISTIC TABLE
1456                                    :  CLEAR THE EOT TABLE
1457                                    :  IF SR1 BIT4 = 0 (TEST GCR)
1458                                    :  :  THEN - SET BIT0 OF FLAGS (GCR)
1459                                    :  :  ELSE - CLEAR BIT0 OF FLAGS (PE)
1460                                    :  ENDIF
1461                                    :  SET THE BOT PASS FLAG
1462                                    :  CLEAR RECORD NUMBER
1463                                    :  SET FILE NUMBER=1
1464                                    ENDROUTINE
1465
1466                                    -
1467
1468  001344' 005001                  M1.5:   CLR     R1              ;CLEAR THE LOOP COUNT
1469  001346' 005061  010774'          1$:    CLR     RWRIT(R1)       ;CLEAR RECOVERABLE WRITE STATISTIC TABLE
1470  001352' 005061  010674'                 CLR     RREAD(R1)       ;CLEAR RECOVERABLE READ STATISTIC TABLE
1471  001356' 005061  010574'                 CLR     URREAD(R1)      ;CLEAR UNRECOVERABLE READ STATISTIC TABLE
1472  001362' 062701  000002                  ADD     #2,R1           ;INCREMENT THE LOOP COUNT
1473  001366' 022701  000100                  CMP     #64.,R1         ;DO UNTIL THE LOOP COUNT=64.
1474  001372' 001365                          BNE     1$
1475  001374' 005001                          CLR     R1              ;CLEAR THE LOOP COUNT
1476  001376' 005061  010534'          2$:    CLR     EOTTBL(R1)      ;CLEAR THE EOT TABLE
1477  001402' 062701  000002                  ADD     #2,R1           ;INCREMENT THE LOOP COUNT
1478  001406' 022701  000020                  CMP     #16.,R1         ;DO UNTIL THE LOOP COUNT=16
1479  001412' 001371                          BNE     2$
1480  001414' 052767  000001  007052          BIS     #BIT0,FLAGS     ;SET THE DENSITY TO GCR
1481  001422' 032767  000020  176366          BIT     #BIT4,SR1       ;IF SR1 BIT4 = 0 (TEST GCR)
1482  001430' 001410                          BEQ     3$              ;THEN - SET THE DENSITY TO GCR
1483  001432' 042767  000001  007034          BIC     #BIT0,FLAGS     ;ELSE - SET THE DENSITY TO PE
1484  001440' 005067  007040                  CLR     RECORD          ;CLEAR RECORD NUMBER
1485  001444' 012767  000001  007030          MOV     #1,FILE         ;SET FILE NUMBER=1
1486  001452' 052767  000002  007014  3$:     BIS     #BIT1,FLAGS     ;SET THE BOT PASS FLAG
1487  001460' 000207                          RTS     PC              ;ENDROUTINE
1488
1489  001462' 016701  006754          TMCONV: MOV     TM78,R1
1490  001466' 012702  000001                  MOV     #1,R2
1491  001472' 005701                  1$:     TST     R1
1492  001474' 001403                          BEQ     2$
1493  001476' 005301                          DEC     R1
1494  001500' 006302                          ASL     R2
1495  001502' 000773                          BR      1$
1496  001504' 010267  006720          2$:     MOV     R2,TMBIT
1497  001510' 016701  006726                  MOV     TM78,R1
1498  001514' 006301                          ASL     R1
1499  001516' 010167  006710                  MOV     R1,TMBIT2
1500  001522' 000207                          RTS     PC
```

```
1502  001524' 016701  006714          TUCONV: MOV     TU78,R1
1503  001530' 012702  000001                  MOV     #1,R2
1504  001534' 005701                  1$:     TST     R1
1505  001536' 001403                          BEQ     2$
1506  001540' 005301                          DEC     R1
1507  001542' 006302                          ASL     R2
1508  001544' 000773                          BR      1$
1509  001546' 010267  006662          2$:     MOV     R2,TUBIT
1510  001552' 016701  006666                  MOV     TU78,R1
1511  001556' 006301                          ASL     R1
1512  001560' 010167  006652                  MOV     R1,TUBIT2
1513  001564' 000207                          RTS     PC
1514                                  .REM    -
1515                                  BEGINROUTINE    MODULE 2.0 (DRIVER SEQUENCE CONTROL)
1516                                  :  BGNDO
1517                                  :  :  ENABLE RH11/70 INTERRUPTS
1518                                  :  :  CLEAR INTERRUPT EXPECTED FLAG
1519                                  :  :  CLEAR THE TM78#
1520                                  :  :  INCREMENT RECORD NUMBER
1521                                  :  :  BGNDO
1522                                  :  :  :  GET WRITE BUFFER (DEC/X11 GWBUFF)
1523                                  :  :  :  CLEAR THE TU78#
1524                                  :  :  :  BGNDO
1525                                  :  :  :  :  IF THE TU78 SELECTED BIT (SCRATCH UNIT TABLE(TM78#,TU78#)) = 0
1526                                  :  :  :  :  :  THEN-CONTINUE
1527                                  :  :  :  :  :  ELSE-IF BOT PASS FLAG NOT=0
1528                                  :  :  :  :  :        :  THEN-CALL MODULE 2.7 (REWIND CONTROL)
1529                                  :  :  :  :  :        :  ELSE-IF EOT FLAG=1(EOTTBL(TM78#,TU78#))
1530                                  :  :  :  :  :        :        :  THEN-CONTINUE
1531                                  :  :  :  :  :        :        :  ELSE-CALL MODULE 2.3 (WRITE CONTROL)
1532                                  :  :  :  :  :        :        :        IF TU78 SELECTED BIT (SCRATCH UNIT TABLE(TM78#,TU78#
1533                                  :  :  :  :  :        :        :        :  THEN-CALL MODULE 2.5 (READ REVERSE CONTROL)
1534                                  :  :  :  :  :        :        :        :  ELSE-CONTINUE
1535                                  :  :  :  :  :        :        :        ENDIF
1536                                  :  :  :  :  :        :        :        IF TU78 SELECTED BIT (SCRATCH UNIT TABLE(TM78#,TU78#
1537                                  :  :  :  :  :        :        :        :  THEN-CALL MODULE 2.8 (COMPARE DATA)
1538                                  :  :  :  :  :        :        :        :        CALL MODULE 2.4 (READ FORWARD CONTROL)
1539                                  :  :  :  :  :        :        :        :  ELSE-CONTINUE
1540                                  :  :  :  :  :        :        :        ENDIF
1541                                  :  :  :  :  :        :        :        IF TU78 SELECTED BIT (SCRATCH UNIT TABLE(TM78#,TU78#
1542                                  :  :  :  :  :        :        :        :  THEN-CALL MODULE 2.8 (COMPARE DATA)
1543                                  :  :  :  :  :        :        :        :  ELSE-CONTINUE
1544                                  :  :  :  :  :        :        :        ENDIF
1545                                  :  :  :  :  :        :        ENDIF
1546                                  :  :  :  :  :        ENDIF
1547                                  :  :  :  :  ENDIF
1548                                  :  :  :  :  INCREMENT THE TU78#
1549                                  :  :  :  :  DO UNTIL THE TU78# = 4
1550                                  :  :  :  ENDDO
1551                                  :  :  :  INCREMENT THE TM78# BY 1
1552                                  :  :  :  DO UNTIL THE TM78# = 10(8)
1553                                  :  :  ENDDO
1554                                  :  :  CALL MODULE 2.10 (UNIT CONTROL)
1555                                  :  :  IF UNIT COUNT=0
1556                                  :  :  :  THEN-CLEAR RH11/70 REGISTER CS1
1557                                  :  :  :        PRINT ERROR 11
```



```
1558                                  :  :  :        DROP THIS DEC/X11 DRIVER (ENDMOD)
1559                                  :  :  :  ELSE-CONTINUE
1560                                  :  :  ENDIF
1561                                  :  :  IF BOT PASS FLAG=1
1562                                  :  :  :  THEN-BGNDO
1563                                  :  :  :        :  CALL MODULE 2.10 (UNIT CONTROL)
1564                                  :  :  :        :  CALL MODULE 2.2 (BOT CONTROL)
1565                                  :  :  :        :  DO UNTIL UNIT COUNT=0 OR UNIT COUNT=BOT COUNT
1566                                  :  :  :        ENDDO
1567                                  :  :  :        CLEAR THE EOT COUNT
1568                                  :  :  :        CLEAR THE EOT TABLE
1569                                  :  :  :        SET FILE NUMBER=1
1570                                  :  :  :        CLEAR RECORD NUMBER
1571                                  :  :  :  ELSE-CONTINUE
1572                                  :  :  ENDIF
1573                                  :  :  CLEAR THE BOT PASS FLAG
1574                                  :  :  IF UNIT COUNT=EOT COUNT
1575                                  :  :  :  THEN-SET BOT PASS FLAG
1576                                  :  :  :        CALL MODULE 2.9 (STATISTICS)
1577                                  :  :  :        CALL MODULE 2.1 (DENSITY CONTROL)
1578                                  :  :  :  ELSE-CONTINUE
1579                                  :  :  ENDIF
1580                                  :  :  IF RECORD NUMBER=500. OR BOT PASS FLAG=1
1581                                  :  :  :  THEN-CALL MODULE 2.6 (TAPE MARK CONTROL)
1582                                  :  :  :        INCREMENT FILE NUMBER
1583                                  :  :  :  ELSE-CONTINUE
1584                                  :  :  ENDIF
1585                                  :  :  DO UNTIL RECORD NUMBER=500. OR BOT PASS FLAG=1
1586                                  :  ENDDO
1587                                  :  CLEAR THE RECORD NUMBER
1588                                  :  END THE PASS (DEC/X11 ENDIT)
1589                                  ENDROUTINE
1590
1591                                  -
1592  001566' 000240                  RESTRT: NOP                     ;DEC/X11 RESTART TAG
1593  001570' 042767  000040  006676  M2.0:   BIC     #BIT5,FLAGS     ;CLEAR INTERRUPT EXPECTED FLAG
1594  001576' 012777  000100  007270          MOV     #BIT6,@CS1      ;ENABLE RH11/70 INTERRUPTS
1595  001604' 005067  006632                  CLR     TM78            ;CLEAR THE TM78#
1596  001610' 104414  000000'                GWBUF$, BEGIN           ;GET WRITE BUFFER INFORMATION
1597  001614' 005267  006664                  INC     RECORD          ;INCREMENT THE RECORD NUMBER
1598  001620' 005067  006620          1$:     CLR     TU78            ;CLEAR TU78#
1599  001624' 004767  177632          2$:     CALL    TMCONV
1600  001630' 004767  177670                  CALL    TUCONV
1601  001634' 016701  006572                  MOV     TMBIT2,R1
1602  001640' 036761  006570  010554'        BIT     TUBIT,SUTBL(R1) ;IF TU78 SELECTED BIT=0
1603  001646' 001451                          BEQ     6$              ;THEN-CONTINUE
1604  001650' 032767  000002  006616          BIT     #BIT1,FLAGS     ;ELSE-IF BOT PASS FLAG NOT=0
1605  001656' 001403                          BEQ     3$
1606  001660' 004767  002450                  CALL    M2.7            ;THEN-CALL MODULE 2.7 (REWIND CONTROL)
1607  001664' 000442                          BR      6$              ;
1608  001666' 016701  006540          3$:     MOV     TMBIT2,R1       ;ELSE-IF EOT FLAG BIT=1
1609  001672' 036761  006536  010534'        BIT     TUBIT,EOTTBL(R1)
1610  001700' 001034                          BNE     6$              ;THEN-CONTINUE
1611  001702' 004767  000666                  CALL    M2.3            ;ELSE-CALL MODULE 2.3 (WRITE CONTROL)
1612  001706' 016701  006520                  MOV     TMBIT2,R1
1613  001712' 036761  006516  010554'        BIT     TUBIT,SUTBL(R1) ;IF TU SELECTED BIT=1
```

```
1614  001720' 001402                          BEQ     4$              ;ELSE=CONTINUE
1615  001722' 004767  001620                  CALL    M2.5            ;THEN=CALL MODULE 2.5 (READ REVERSE CONTROL)
1616  001726' 016701  006500          4$:     MOV     TMBIT2,R1
1617  001732' 036761  006476  010554'        BIT     TUBIT,SUTBL(R1) ;IF TU SELECTED BIT=1
1618  001740' 001404                          BEQ     5$              ;ELSE=CONTINUE
1619  001742' 004767  002520                  CALL    M2.8            ;THEN=CALL MODULE 2.8 (COMPARE DATA)
1620  001746' 004767  001224                  CALL    M2.4            ;CALL MODULE 2.4 (READ FORWARD CONTROL)
1621  001752' 016701  006454          5$:     MOV     TMBIT2,R1
1622  001756' 036761  006452  010554'        BIT     TUBIT,SUTBL(R1) ;IF TU SELECTED BIT=1
1623  001764' 001402                          BEQ     6$              ;ELSE=CONTINUE
1624  001766' 004767  002474                  CALL    M2.8            ;THEN=CALL MODULE 2.8 (COMPARE DATA)
1625  001772' 005267  006446          6$:     INC     TU78            ;INCREMENT THE TU78#
1626  001776' 022767  000004  006440          CMP     #4,TU78         ;DO UNTIL THE TU78#=4
1627  002004' 001307                          BNE     2$
1628
1629  002006' 005267  006430                  INC     TM78            ;INCREMENT THE TM78#
1630  002012' 022767  000010  006422          CMP     #10,TM78        ;DO UNTIL THE TM78#=10
1631  002020' 001277                          BNE     1$
1632
1633  002022' 004767  003114                  CALL    M2.10           ;CALL MODULE 2.10 (UNIT CONTROL)
1634  002026' 005767  006406                  TST     UCNT            ;IF UNIT COUNT=0
1635  002032' 001011                          BNE     7$              ;ELSE=CONTINUE
1636  002034' 012767  000011  006416          MOV     #11,ERRINX      ;PRINT ERROR 11
1637  002042' 004767  004566                  CALL    M4.1
1638  002046' 005077  007022                  CLR     @CS1            ;THEN=CLEAR RH11/70 REGISTER CS1
1639  002052' 104410  000000'                 ENDS,BEGIN              ;
1640
1641  002056' 032767  000002  006410  7$:     BIT     #BIT1,FLAGS     ;IF BOT PASS FLAG=1
1642  002064' 001432                          BEQ     11$             ;ELSE=CONTINUE
1643  002066' 004767  003050          8$:     CALL    M2.10           ;CALL MODULE 2.10 (UNIT CONTROL)
1644  002072' 004767  000244                  CALL    M2.2            ;CALL MODULE 2.2 (BOT CONTROL)
1645  002076' 005767  006336                  TST     UCNT            ;DO UNTIL THE UNIT
1646  002102' 001404                          BEQ     9$              ;COUNT=0 OR THE UNIT
1647  002104' 026767  006344  006326          CMP     BOTCNT,UCNT     ;COUNT=THE BOT COUNT
1648  002112' 001365                          BNE     8$
1649  002114' 005001                  9$:     CLR     R1
1650  002116' 005067  006324                  CLR     EOTCNT          ;CLEAR THE EOT COUNT
1651  002122' 005061  010534'          10$:    CLR     EOTTBL(R1)      ;CLEAR THE EOT TABLE
1652  002126' 062701  000002                  ADD     #2,R1
1653  002132' 022701  000020                  CMP     #20,R1
1654  002136' 001371                          BNE     10$
1655  002140' 012767  000001  006334          MOV     #1,FILE         ;SET FILE NUMBER=1
1656  002146' 005067  006332                  CLR     RECORD          ;CLEAR RECORD NUMBER
1657
1658  002152' 042767  000002  006314  11$:    BIC     #BIT1,FLAGS     ;CLEAR THE BOT PASS FLAG
1659  002160' 026767  006254  006260          CMP     UCNT,EOTCNT     ;IF UNIT COUNT=EOT COUNT
1660  002166' 001007                          BNE     12$             ;ELSE=CONTINUE
1661  002170' 052767  000002  006276          BIS     #BIT1,FLAGS     ;SET THE BOT PASS FLAG
1662  002176' 004767  002316                  CALL    M2.9            ;THEN=CALL MODULE 2.9 (STATISTICS)
1663  002202' 004767  000064                  CALL    M2.1            ;CALL MODULE 2.1 (DENSITY CONTROL)
1664
1665  002206' 022767  000764  006270  12$:    CMP     #500.,RECORD    ;IF RECORD NUMBER=500.
1666  002214' 001404                          BEQ     13$             ;
1667  002216' 032767  000002  006250          BIT     #BIT1,FLAGS     ;OR BOT PASS FLAG=1
1668  002224' 001404                          BEQ     14$             ;ELSE=CONTINUE
1669  002226' 004767  001660          13$:    CALL    M2.6            ;THEN=CALL MODULE 2.6 (TAPE MARK CONTROL)
```



```
1670  002232' 005267  006244                  INC     FILE            ;INCREMENT THE FILE NUMBER
1671  002236' 022767  000764  006240  14$:    CMP     #500.,RECORD    ;DO UNTIL RECORD NUMBER=500.
1672  002244' 001406                          BEQ     15$
1673  002246' 032767  000002  006220          BIT     #BIT1,FLAGS     ;OR BOT PASS FLAG=1
1674  002254' 001002                          BNE     15$
1675  002256' 000167  177306                  JMP     M2.0
1676  002262' 005067  006216          15$:    CLR     RECORD
1677  002266' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
 (1)                                                                  ;MONITOR SHALL TEST END OF PASS
1678                                  .REM    -
1679                                  BEGINROUTINE    MODULE 2.1 (DENSITY CONTROL)
1680                                  :  IF DENSITY FLAG=0
1681                                  :  :  THEN-IF BIT4 OF SR1 (TEST GCR SWITCH)=0
1682                                  :  :       :  THEN-SET THE DENSITY FLAG (GCR)
1683                                  :  :       :  ELSE-CONTINUE
1684                                  :  :       ENDIF
1685                                  :  :  ELSE-IF BIT5 OF SR1 (TEST PE SWITCH)=0
1686                                  :  :       :  THEN-CLEAR THE DENSITY FLAG (PE)
1687                                  :  :       :  ELSE-CONTINUE
1688                                  :  :       ENDIF
1689                                  :  ENDIF
1690                                  ENDROUTINE
1691
1692                                  -
1693
1694  002272' 032767  000001  006174  M2.1:   BIT     #BIT0,FLAGS     ;IF DENSITY FLAG=0
1695  002300' 001010                          BNE     1$              ;ELSE=GO TO ELSE
1696  002302' 032767  000020  175506          BIT     #BIT4,SR1       ;THEN=IF BIT4 OF SR1=0
1697  002310' 001013                          BNE     2$              ;ELSE=CONTINUE
1698  002312' 052767  000001  006154          BIS     #BIT0,FLAGS     ;THEN=SET THE DENSITY FLAG
1699  002320' 000407                          BR      2$              ;
1700  002322' 032767  000040  175466  1$:     BIT     #BIT5,SR1       ;IF BIT5 OF SR1=0
1701  002330' 001003                          BNE     2$              ;ELSE=CONTINUE
1702  002332' 042767  000001  006134          BIC     #BIT0,FLAGS     ;THEN=CLEAR THE DENSITY FLAG
1703  002340' 000207                  2$:     RTS     PC              ;RETURN
```

```
                                .REM    -
                                BEGINROUTINE    MODULE 2.2 (BOT CONTROL)
                                :  CLEAR THE BOT COUNT
                                :  CLEAR THE TM78#
                                :  BGNDO
                                :  :  CLEAR THE TU78#
                                :  :  BGNDO
                                :  :  :  IF BIT SET (SCRATCH UNIT TABLE(TM78#, TU78#))
                                :  :  :  :  THEN-CALL MODULE 3.6 (SENSE)
                                :  :  :  :        SELECT TERMINATION INTERRUPT CODE
                                :  :  :  :        :  DONE-IF BIT10 OF SENSE=1 (TU AT BOT)
                                :  :  :  :        :        :  THEN-INCREMENT BOT COUNT
                                :  :  :  :        :        :  ELSE-CONTINUE
                                :  :  :  :        :        ENDIF
                                :  :  :  :        :        IF "TRE" BIT IN RH11/70 CS1 REGISTER=1
                                :  :  :  :        :        :  THEN-PRINT ERROR 10
                                :  :  :  :        :        :  ELSE-CONTINUE
                                :  :  :  :        :        ENDIF
                                :  :  :  :        :  TM FAULT B-PRINT ERROR 7
                                :  :  :  :        :              DROP TM78-CLEAR ALL BITS (SCRATCH UNIT TABLE (TM78#))
                                :  :  :  :        :  ALL OTHERS-PRINT ERROR 2
                                :  :  :  :        :              DROP TU78-CLEAR BIT (SCRATCH UNIT TABLE(TM78#, TU78#))
                                :  :  :  :        ENDSELECT
                                :  :  :  :  ELSE-CONTINUE
                                :  :  :  ENDIF
                                :  :  :  INCREMENT THE TU78#
                                :  :  :  DO UNTIL THE TU78#=4
                                :  :  ENDDO
                                :  :  INCREMENT THE TM78#
                                :  :  DO UNTIL THE TM78#=10(8)
                                :  ENDDO
                                ENDROUTINE

                                -

002342' 005067  006106          M2.2:   CLR     BOTCNT          ;CLEAR BOT COUNT
002346' 005067  006070                  CLR     TM78            ;CLEAR THE TM78#
002352' 005067  006066          1$:     CLR     TU78            ;CLEAR THE TU78#
002356' 004767  177100          2$:     CALL    TMCONV          ;
002362' 004767  177136                  CALL    TUCONV
002366' 016701  006040                  MOV     TMBIT2,R1
002372' 036761  006036  010554'         BIT     TUBIT,SUTBL(R1) ;IF BIT SET (SCRATCH UNIT TABLE(TM78#,TU78#))
002400' 001460                          BEQ     6$              ;ELSE=CONTINUE
002402' 004767  003624                  CALL    M3.6            ;THEN=CALL MODULE 3.6 (ISSUE SENSE)
002406' 016701  006020                  MOV     TMBIT2,R1
002412' 022767  000001  006030          CMP     #DONE,TINTCD    ;DONE?
002420' 001020                          BNE     4$              ;NO-CONTINUE
002422' 032767  002000  006026          BIT     #BIT10,SENSE    ;YES-IF TU AT BOT?
002430' 001402                          BEQ     3$              ;ELSE=CONTINUE
002432' 005267  006016                  INC     BOTCNT          ;THEN-INCREMENT BOT COUNT
002436' 032777  040000  006430  3$:     BIT     #BIT14,@CS1     ;IF 'TRE' BIT IN RH11/70 CS1=1
002444' 001436                          BEQ     6$              ;ELSE-CONTINUE
                                                                ;THEN-
002446' 012767  000010  006004          MOV     #10,ERRINX
002454' 004767  004154                  CALL    M4.1
002460' 000430                          BR      6$
```



```
002462' 022767  000032  005760  4$:     CMP     #TMFB,TINTCD    ;TM FAULT B?
002470' 001012                          BNE     5$              ;NO-CONTINUE
002472' 012767  000007  005760          MOV     #7,ERRINX
002500' 004767  004130                  CALL    M4.1
002504' 016701  005722                  MOV     TMBIT2,R1       ;DROP TM78
002510' 005061  010554'                 CLR     SUTBL(R1)
002514' 000412                          BR      6$
002516' 012767  000002  005734  5$:     MOV     #2,ERRINX
002524' 004767  004104                  CALL    M4.1
002530' 016701  005676                  MOV     TMBIT2,R1       ;DROP TU78
002534' 046761  005674  010554'         BIC     TUBIT,SUTBL(R1)
002542' 005267  005676          6$:     INC     TU78            ;INCREMENT THE TU78#
002546' 022767  000004  005670          CMP     #4,TU78         ;DO UNTIL THE TU78#=4
002554' 001300                          BNE     2$

002556' 005267  005660                  INC     TM78            ;INCREMENT THE TM78#
002562' 022767  000010  005652          CMP     #10,TM78        ;DO UNTIL THE TM78#=10
002570' 001270                          BNE     1$
002572' 000207                          RTS     PC              ;RETURN
```

```
                                        .REM    -
                                        BEGINROUTINE    MODULE 2.3 (WRITE CONTROL)
                                        :  CLEAR RETRY IN PROGRESS FLAG
                                        :  CLEAR THE EXIT FLAG
                                        :  BGNDO
                                        :  :  CALL MODULE 3.1 (WRITE)
                                        :  :  SELECT TERMINATION INTERRUPT CODE
                                        :  :  :  DONE-SET THE EXIT FLAG
                                        :  :  :        IF RETRY IN PROGRESS FLAG=1
                                        :  :  :        :  THEN-UPDATE RECOVERABLE WRITE ERROR STATISTICS
                                        :  :  :        :       PRINT ERROR 3
                                        :  :  :        :  ELSE-CONTINUE
                                        :  :  :        ENDIF
                                        :  :  :        IF "TRE" BIT IN RH11/70 CS1 REGISTER=1
                                        :  :  :        :  THEN-PRINT ERROR 10
                                        :  :  :        :  ELSE-CONTINUE
                                        :  :  :        ENDIF
                                        :  :  :  EOT -SET THE EOT FLAG (EOT TABLE (TM78#,TU78#))
                                        :  :  :        SET THE EXIT FLAG
                                        :  :  :        IF RETRY IN PROGRESS FLAG NOT=0
                                        :  :  :        :  THEN-UPDATE RECOVERABLE WRITE ERROR STATISTICS
                                        :  :  :        :       PRINT ERROR 3
                                        :  :  :        :  ELSE-CONTINUE
                                        :  :  :        ENDIF
                                        :  :  :  BAD TAPE-SET THE EXIT FLAG
                                        :  :  :           PRINT ERROR 4
                                        :  :  :           DROP THE TU78-CLEAR BIT (SCRATCH UNIT TABLE(TM78#, TU78#))
                                        :  :  :  RETRY-SET RETRY IN PROGRESS FLAG
                                        :  :  :  TM FAULT B-PRINT ERROR 7
                                        :  :  :            DROP TM78-CLEAR ALL BITS (SCRATCH UNIT TABLE (TM78#))
                                        :  :  :            SET THE EXIT FLAG
                                        :  :  :  ALL OTHER-PRINT ERROR 2
                                        :  :  :            DROP TU78-CLEAR BIT (SCRATCH UNIT TABLE(TM78#, TU78#))
                                        :  :  :            SET THE EXIT FLAG
                                        :  :  ENDSELECT
                                        :  :  DO UNTIL THE EXIT FLAG NOT=0
                                        :  ENDDO
                                        ENDROUTINE
                                        -
002574' 042767  000004  005672  M2.3:   BIC     #BIT2,FLAGS     ;CLEAR RETRY IN PROGRESS FLAG
002602' 042767  000020  005664          BIC     #BIT4,FLAGS     ;CLEAR EXIT FLAG
002610' 004767  002444          1$:     CALL    M3.1            ;CALL MODULE 3.1 (WRITE)
002614' 016701  005612                  MOV     TMBIT2,R1
002620' 022767  000001  005622          CMP     #DONE,TINTCD    ;DONE?
002626' 001034                          BNE     4$              ;NO-CONTINUE
002630' 052767  000020  005636  2$:     BIS     #BIT4,FLAGS     ;YES-SET THE EXIT FLAG
002636' 032767  000004  005630          BIT     #BIT2,FLAGS     ;IF RETRY IN PROGRESS FLAG=1
002644' 001413                          BEQ     3$              ;ELSE-CONTINUE
002646' 016701  005560                  MOV     TMBIT2,R1
002652' 066701  005560                  ADD     TUBIT2,R1       ;ADD IN THE TU DISPLACEMENT
002656' 005261  010774'                 INC     RWRIT(R1)       ;THEN-UPDATE RECOVERABLE WRITE ERROR STATS.
002662' 012767  000003  005570          MOV     #3,ERRINX
002670' 004767  003740                  CALL    M4.1
```



```
002674' 032777  040000  006172  3$:     BIT     #BIT14,@CS1     ;IF 'TRE' BIT IN RH11/70 CS1=1
002702' 001530                          BEQ     9$              ;ELSE-CONTINUE
002704' 012767  000010  005546          MOV     #10,ERRINX
002712' 004767  003716                  CALL    M4.1
002716' 000522                          BR      9$

002720' 022767  000004  005522  4$:     CMP     #EOT,TINTCD     ;EOT?
002726' 001026                          BNE     5$              ;NO-CONTINUE
002730' 016701  005476                  MOV     TMBIT2,R1       ;SET THE EOT FLAG
002734' 056761  005474  010534'         BIS     TUBIT,EOTTBL(R1)
002742' 052767  000020  005524          BIS     #BIT4,FLAGS     ;YES-SET THE EXIT FLAG
002750' 032767  000004  005516          BIT     #BIT2,FLAGS     ;IF RETRY IN PROGRESS FLAG=1
002756' 001502                          BEQ     9$              ;ELSE-CONTINUE
002760' 066701  005452                  ADD     TUBIT2,R1       ;ADD IN THE TU DISPLACEMENT
002764' 005261  010774'                 INC     RWRIT(R1)       ;THEN-UPDATE RECOVERABLE WRITE ERROR STATS.
002770' 012767  000003  005462          MOV     #3,ERRINX
002776' 004767  003632                  CALL    M4.1
003002' 000470                          BR      9$

003004' 022767  000027  005436  5$:     CMP     #BTAPE,TINTCD   ;BAD TAPE?
003012' 001016                          BNE     6$              ;NO-CONTINUE
003014' 052767  000020  005452          BIS     #BIT4,FLAGS     ;YES-SET THE EXIT FLAG
003022' 012767  000004  005430          MOV     #4,ERRINX
003030' 004767  003600                  CALL    M4.1
003034' 016701  005372                  MOV     TMBIT2,R1       ;DROP TU78
003040' 046761  005370  010554'         BIC     TUBIT,SUTBL(R1)
003046' 000446                          BR      9$

003050' 022767  000022  005372  6$:     CMP     #RTRY,TINTCD    ;RETRY?
003056' 001004                          BNE     7$              ;NO-CONTINUE
003060' 052767  000004  005406          BIS     #BIT2,FLAGS     ;YES-SET RETRY IN PROGRESS FLAG
003066' 000436                          BR      9$
003070' 022767  000032  005352  7$:     CMP     #TMFB,TINTCD    ;TM FAULT B?
003076' 001015                          BNE     8$              ;NO-CONTINUE
003100' 012767  000007  005352          MOV     #7,ERRINX
003106' 004767  003522                  CALL    M4.1
003112' 016701  005314                  MOV     TMBIT2,R1       ;DROP TM78
003116' 005061  010554'                 CLR     SUTBL(R1)
003122' 052767  000020  005344          BIS     #BIT4,FLAGS     ;SET THE EXIT FLAG
003130' 000415                          BR      9$

003132' 052767  000020  005334  8$:     BIS     #BIT4,FLAGS     ;SET THE EXIT FLAG
003140' 012767  000002  005312          MOV     #2,ERRINX
003146' 004767  003462                  CALL    M4.1
003152' 016701  005254                  MOV     TMBIT2,R1       ;DROP TU78
003156' 046761  005252  010554'         BIC     TUBIT,SUTBL(R1)

003164' 032767  000020  005302  9$:     BIT     #BIT4,FLAGS     ;DO UNTIL EXIT FLAG=1
003172' 001606                          BEQ     1$
003174' 000207                          RTS     PC              ;RETURN
```

```
                                        .REM    -
                                        BEGINROUTINE    MODULE 2.4 (READ FORWARD CONTROL)
                                        :  CLEAR THE RETRY IN PROGRESS FLAG
                                        :  CLEAR THE EXIT FLAG
                                        :  CALL MODULE 3.2 (READ FORWARD)
                                        :  BGNDO
                                        :  :  SELECT TERMINATION INTERRUPT CODE
                                        :  :  :  DONE-SET THE EXIT FLAG
                                        :  :  :        IF RETRY IN PROGRESS FLAG=1
                                        :  :  :        :  THEN-UPDATE RECOVERABLE READ ERROR STATISTICS
                                        :  :  :        :       PRINT ERROR 5
                                        :  :  :        :  ELSE-CONTINUE
                                        :  :  :        ENDIF
                                        :  :  :        IF "TRE" BIT IN RH11/70 CS1 REGISTER=1
                                        :  :  :        :  THEN-PRINT ERROR 10
                                        :  :  :        :  ELSE-CONTINUE
                                        :  :  :        ENDIF
                                        :  :  :  RETRY-SET THE RETRY IN PROGRESS FLAG
                                        :  :  :         CALL MODULE 3.2 (READ FORWARD)
                                        :  :  :  READ OPP-SET THE RETRY IN PROGRESS FLAG
                                        :  :  :           CALL MODULE 3.3 (READ REVERSE)
                                        :  :  :  UNREADABLE-SET THE EXIT FLAG
                                        :  :  :             UPDATE UNRECOVERABLE READ ERROR STATISTICS
                                        :  :  :             PRINT ERROR 6
                                        :  :  :  TM FAULT B-PRINT ERROR 7
                                        :  :  :             DROP TM78-CLEAR ALL BITS (SCRATCH UNIT TABLE (TM78#))
                                        :  :  :             SET THE EXIT FLAG
                                        :  :  :  ALL OTHER-PRINT ERROR 2
                                        :  :  :             DROP THE TU78-CLEAR BIT (SCRATCH UNIT TABLE(TM78#, TU78#)
                                        :  :  :             SET THE EXIT FLAG
                                        :  :  ENDSELECT
                                        :  :  DO UNTIL THE EXIT FLAG NOT=0
                                        :  ENDDO
                                        ENDROUTINE

                                        -

003176' 042767  000004  005270  M2.4:   BIC     #BIT2,FLAGS     ;CLEAR RETRY IN PROGRESS FLAG
003204' 042767  000020  005262          BIC     #BIT4,FLAGS     ;CLEAR EXIT FLAG
003212' 004767  002310                  CALL    M3.2            ;CALL MODULE 3.2 (READ FORWARD)
003216' 016701  005210                  MOV     TMBIT2,R1

003222' 022767  000001  005220  1$:     CMP     #DONE,TINTCD    ;DONE?
003230' 001034                          BNE     3$              ;NO-CONTINUE
003232' 052767  000020  005234          BIS     #BIT4,FLAGS     ;YES-SET EXIT FLAG
003240' 032767  000004  005226          BIT     #BIT2,FLAGS     ;IF RETRY IN PROGRESS=1
003246' 001413                          BEQ     2$
003250' 016701  005156                  MOV     TMBIT2,R1       ;THEN-UPDATE RECOVERABLE READ ERRORS
003254' 066701  005156                  ADD     TUBIT2,R1       ;ADD IN THE TU DISPLACEMENT
003260' 005261  010674'                 INC     RREAD(R1)
003264' 012767  000005  005166          MOV     #5,ERRINX
003272' 004767  003336                  CALL    M4.1

003276' 032777  040000  005570  2$:     BIT     #BIT14,@CS1     ;IF TRE BIT IN RH11/70 CS1=1
003304' 001513                          BEQ     8$              ;ELSE-CONTINUE
003306' 012767  000010  005144          MOV     #10,ERRINX
```



```
003314' 004767  003314                  CALL    M4.1
003320' 000505                          BR      8$
003322' 022767  000022  005120  3$:     CMP     #RTRY,TINTCD    ;RETRY
003330' 001006                          BNE     4$              ;NO-CONTINUE
003332' 052767  000004  005134          BIS     #BIT2,FLAGS     ;YES-SET RETRY IN PROGRESS FLAG
003340' 004767  002162                  CALL    M3.2            ;CALL MODULE 3.2 (READ FORWARD)
003344' 000473                          BR      8$

003346' 022767  000023  005074  4$:     CMP     #RDOPP,TINTCD   ;READ OPPOSITE
003354' 001006                          BNE     5$              ;NO-CONTINUE
003356' 052767  000004  005110          BIS     #BIT2,FLAGS     ;YES-SET RETRY IN PROGRESS FLAG
003364' 004767  002414                  CALL    M3.3            ;CALL MODULE 3.3 (READ REVERSE)
003370' 000461                          BR      8$

003372' 022767  000024  005050  5$:     CMP     #UNREAD,TINTCD  ;UNREADABLE?
003400' 001017                          BNE     6$              ;NO-CONTINUE
003402' 052767  000020  005064          BIS     #BIT4,FLAGS     ;YES-SET EXIT FLAG
003410' 016701  005016                  MOV     TMBIT2,R1       ;UPDATE UNRECOVERABLE READ ERRORS
003414' 066701  005016                  ADD     TUBIT2,R1       ;ADD IN THE TU DISPLACEMENT
003420' 005261  010574'                 INC     URREAD(R1)
003424' 012767  000006  005026          MOV     #6,ERRINX       ;PRINT ERROR 6
003432' 004767  003176                  CALL    M4.1
003436' 000436                          BR      8$

003440' 022767  000032  005002  6$:     CMP     #TMFB,TINTCD    ;TM FAULT B?
003446' 001015                          BNE     7$              ;NO-CONTINUE
003450' 052767  000020  005016          BIS     #BIT4,FLAGS     ;YES-SET THE EXIT FLAG
003456' 012767  000007  004774          MOV     #7,ERRINX
003464' 004767  003144                  CALL    M4.1
003470' 016701  004736                  MOV     TMBIT2,R1       ;DROP TM78
003474' 005061  010554'                 CLR     SUTBL(R1)
003500' 000415                          BR      8$

003502' 052767  000020  004764  7$:     BIS     #BIT4,FLAGS     ;SET THE EXIT FLAG
003510' 012767  000002  004742          MOV     #2,ERRINX
003516' 004767  003112                  CALL    M4.1
003522' 016701  004704                  MOV     TMBIT2,R1       ;DROP TU78
003526' 046761  004702  010554'         BIC     TUBIT,SUTBL(R1)

003534' 032767  000020  004732  8$:     BIT     #BIT4,FLAGS     ;DO UNTIL EXIT FLAG=1
003542' 001627                          BEQ     1$
003544' 000207                          RTS     PC              ;RETURN
```

```
                                .REM    -
                                BEGINROUTINE    MODULE 2.5 (READ REVERSE CONTROL)
                                :  CLEAR THE RETRY IN PROGRESS FLAG
                                :  CLEAR THE EXIT FLAG
                                :  CALL MODULE 3.3 (READ REVERSE)
                                :  BGNDO
                                :  :  SELECT TERMINATION INTERRUPT CODE
                                :  :  :  DONE-SET THE EXIT FLAG
                                :  :  :        IF RETRY IN PROGRESS FLAG NOT=0
                                :  :  :        :  THEN-UPDATE RECOVERABLE READ ERROR STATISTICS
                                :  :  :        :       PRINT ERROR 5
                                :  :  :        :  ELSE-CONTINUE
                                :  :  :        ENDIF
                                :  :  :        IF "TRE" BIT IN RH11/70 CS1 REGISTER=1
                                :  :  :        :  THEN-PRINT ERROR 10
                                :  :  :        :  ELSE-CONTINUE
                                :  :  :        ENDIF
                                :  :  :  RETRY-SET THE RETRY IN PROGRESS FLAG
                                :  :  :         CALL MODULE 3.3 (READ REVERSE)
                                :  :  :  READ OPP-SET THE RETRY IN PROGRESS FLAG
                                :  :  :           CALL MODULE 3.2 (READ FORWARD)
                                :  :  :  UNREADABLE-SET THE EXIT FLAG
                                :  :  :             UPDATE UNRECOVERABLE READ ERROR STATISTICS
                                :  :  :             PRINT ERROR 6
                                :  :  :  TM FAULT B-PRINT ERROR 7
                                :  :  :             DROP TM78-CLEAR ALL BITS (SCRATCH UNIT TABLE (TM78#))
                                :  :  :             SET THE EXIT FLAG
                                :  :  :  ALL OTHER-PRINT ERROR 2
                                :  :  :           DROP THE TU78-CLEAR BIT (SCRATCH UNIT TABLE(TM78#, TU78#)
                                :  :  :           SET THE EXIT FLAG
                                :  :  ENDSELECT
                                :  :  DO UNTIL THE EXIT FLAG NOT=0
                                :  ENDDO
                                ENDROUTINE

                                -
003546' 042767  000004  004720  M2.5:   BIC     #BIT2,FLAGS     ;CLEAR RETRY IN PROGRESS FLAG
003554' 042767  000020  004712          BIC     #BIT4,FLAGS     ;CLEAR EXIT FLAG
003562' 004767  002216                  CALL    M3.3            ;CALL MODULE 3.3 (READ REVERSE)

003566' 022767  000001  004654  1$:     CMP     #DONE,TINTCD    ;DONE?
003574' 001034                          BNE     3$              ;NO-CONTINUE
003576' 052767  000020  004670          BIS     #BIT4,FLAGS     ;YES-SET EXIT FLAG
003604' 032767  000004  004662          BIT     #BIT2,FLAGS     ;IF RETRY IN PROGRESS=1
003612' 001413                          BEQ     2$
003614' 016701  004612                  MOV     TMBIT2,R1       ;THEN-UPDATE RECOVERABLE READ ERRORS
003620' 066701  004612                  ADD     TUBIT2,R1       ;ADD IN THE TU DISPLACEMENT
003624' 005261  010674'                 INC     RREAD(R1)
003630' 012767  000005  004622          MOV     #5,ERRINX
003636' 004767  002772                  CALL    M4.1

003642' 032777  040000  005224  2$:     BIT     #BIT14,@CS1     ;IF TRE BIT IN RH11/70 CS1=1
003650' 001513                          BEQ     8$              ;ELSE-CONTINUE
003652' 012767  000010  004600          MOV     #10,ERRINX
003660' 004767  002750                  CALL    M4.1
```



```
003664' 000505                          BR      8$
003666' 022767  000022  004554  3$:     CMP     #RTRY,TINTCD    ;RETRY
003674' 001006                          BNE     4$              ;NO-CONTINUE
003676' 052767  000004  004570          BIS     #BIT2,FLAGS     ;YES-SET RETRY IN PROGRESS FLAG
003704' 004767  002074                  CALL    M3.3            ;CALL MODULE 3.3 (READ REVERSE)
003710' 000473                          BR      8$

003712' 022767  000023  004530  4$:     CMP     #RDOPP,TINTCD   ;READ OPPOSITE
003720' 001006                          BNE     5$              ;NO-CONTINUE
003722' 052767  000004  004544          BIS     #BIT2,FLAGS     ;YES-SET RETRY IN PROGRESS FLAG
003730' 004767  001572                  CALL    M3.2            ;CALL MODULE 3.2 (READ FORWARD)
003734' 000461                          BR      8$

003736' 022767  000024  004504  5$:     CMP     #UNREAD,TINTCD  ;UNREADABLE?
003744' 001017                          BNE     6$              ;NO-CONTINUE
003746' 052767  000020  004520          BIS     #BIT4,FLAGS     ;YES-SET EXIT FLAG
003754' 016701  004452                  MOV     TMBIT2,R1       ;UPDATE UNRECOVERABLE READ ERRORS
003760' 066701  004452                  ADD     TUBIT2,R1       ;ADD IN THE TU DISPLACEMENT
003764' 005261  010574'                 INC     URREAD(R1)
003770' 012767  000006  004462          MOV     #6,ERRINX       ;PRINT ERROR 6
003776' 004767  002632                  CALL    M4.1
004002' 000436                          BR      8$

004004' 022767  000032  004436  6$:     CMP     #TMFB,TINTCD    ;TM FAULT B?
004012' 001015                          BNE     7$              ;NO-CONTINUE
004014' 052767  000020  004452          BIS     #BIT4,FLAGS     ;YES-SET THE EXIT FLAG
004022' 012767  000007  004430          MOV     #7,ERRINX
004030' 004767  002600                  CALL    M4.1
004034' 016701  004372                  MOV     TMBIT2,R1       ;DROP TM78
004040' 005061  010554'                 CLR     SUTBL(R1)
004044' 000415                          BR      8$

004046' 052767  000020  004420  7$:     BIS     #BIT4,FLAGS     ;SET THE EXIT FLAG
004054' 012767  000002  004376          MOV     #2,ERRINX
004062' 004767  002546                  CALL    M4.1
004066' 016701  004340                  MOV     TMBIT2,R1       ;DROP TU78
004072' 046761  004336  010554'         BIC     TUBIT,SUTBL(R1)

004100' 032767  000020  004366  8$:     BIT     #BIT4,FLAGS     ;DO UNTIL EXIT FLAG=1
004106' 001627                          BEQ     1$
004110' 000207                          RTS     PC              ;RETURN
```

```
                                .REM    -
                                BEGINROUTINE    MODULE 2.6 (TAPE MARK CONTROL)
                                :  CLEAR THE TM78#
                                :  BGNDO
                                :  :  CLEAR THE TU78#
                                :  :  BGNDO
                                :  :  :  IF BIT (SCRATCH UNIT TABLE(TM78#, TU78#)) = 1, AND
                                :  :  :     BIT (EOT TABLE(TM78#, TU78#)) = 0
                                :  :  :  :  THEN-CALL MODULE 3.4 (TAPE MARK)
                                :  :  :  :       SELECT TERMINATION INTERRUPT CODE
                                :  :  :  :       :  DONE-IF "TRE" BIT IN RH11/70 CS1 REGISTER=1
                                :  :  :  :       :       :  THEN-PRINT ERROR 10
                                :  :  :  :       :       :  ELSE-CONTINUE
                                :  :  :  :       :      ENDIF
                                :  :  :  :       :  TM FAULT B-PRINT ERROR 7
                                :  :  :  :       :             DROP TM78-CLEAR ALL BITS (SCRATCH UNIT TABLE (TM78#))
                                :  :  :  :       :  ALL OTHERS-PRINT ERROR 2
                                :  :  :  :       :            DROP TU78-CLEAR BIT (SCRATCH UNIT TABLE(TM78#, TU78#))
                                :  :  :  :       ENDSELECT
                                :  :  :  :  ELSE-CONTINUE
                                :  :  :  ENDIF
                                :  :  :  INCREMENT THE TU78#
                                :  :  :  DO UNTIL THE TU78#=4
                                :  :  ENDDO
                                :  :  INCREMENT THE TM78#
                                :  :  DO UNTIL THE TM78#=10(8)
                                :  ENDDO
                                ENDROUTINE

                                -
004112' 005067  004324          M2.6:   CLR     TM78            ;CLEAR THE TM78#
004116' 005067  004322          1$:     CLR     TU78            ;CLEAR THE TU78#
004122' 004767  175334          2$:     CALL    TMCONV          ;
004126' 004767  175372                  CALL    TUCONV
004132' 016701  004274                  MOV     TMBIT2,R1
004136' 036761  004272  010554'          BIT     TUBIT,SUTBL(R1) ;IF BIT SET (SCRATCH UNIT TABLE(TM78#,TU78#))
004144' 001456                          BEQ     5$              ;ELSE-CONTINUE
004146' 036761  004262  010534'          BIT     TUBIT,EOTTBL(R1)        ;IF BIT (EOT TABLE(TM78#, TU78#)) = 0
004154' 001052                          BNE     5$              ;ELSE - CONTINUE
004156' 004767  001664                  CALL    M3.4            ;CALL MODULE 3.4 (TAPE MARK CONTROL)
004162' 016701  004244                  MOV     TMBIT2,R1
004166' 022767  000001  004254          CMP     #DONE,TINTCD    ;DONE?
004174' 001012                          BNE     3$              ;NO-CONTINUE
004176' 032777  040000  004670          BIT     #BIT14,@CS1     ;YES-IF TRE BIT IN RH11/70 CS1=1
004204' 001436                          BEQ     5$              ;ELSE-CONTINUE
004206' 012767  000010  004244          MOV     #10,ERRINX
004214' 004767  002414                  CALL    M4.1
004220' 000430                          BR      5$

004222' 022767  000032  004220  3$:     CMP     #TMFB,TINTCD    ;TM FAULT B?
004230' 001012                          BNE     4$              ;NO-CONTINUE
004232' 012767  000007  004220          MOV     #7,ERRINX       ;YES - PRINT ERROR 7
004240' 004767  002370                  CALL    M4.1
004244' 016701  004162                  MOV     TMBIT2,R1       ;DROP TM78
004250' 005061  010554'                 CLR     SUTBL(R1)
```



```
004254' 000412                          BR      5$

004256' 012767  000002  004174  4$:     MOV     #2,ERRINX       ;PRINT ERROR 2
004264' 004767  002344                  CALL    M4.1
004270' 016701  004136                  MOV     TMBIT2,R1       ;DROP TU78
004274' 046761  004134  010554'          BIC     TUBIT,SUTBL(R1)

004302' 005267  004136          5$:     INC     TU78            ;INCREMENT THE TU78#
004306' 022767  000004  004130          CMP     #4,TU78         ;DO UNTIL THE TU78#=4
004314' 001302                          BNE     2$

004316' 005267  004120                  INC     TM78            ;INCREMENT THE TM78#
004322' 022767  000010  004112          CMP     #10,TM78        ;DO UNTIL THE TM78#=10
004330' 001272                          BNE     1$
004332' 000207                          RTS     PC              ;RETURN
```

```
                                        .REM    -
                                        BEGINROUTINE    MODULE 2.7 (REWIND CONTROL)
                                        :  CALL MODULE 3.5 (REWIND)
                                        :  SELECT TERMINATION INTERRUPT
                                        :  :  DONE-IF "TRE" BIT IN RH11/70 CS1 REGISTER=1
                                        :  :       :  THEN-PRINT ERROR 10
                                        :  :       :  ELSE-CONTINUE
                                        :  :      ENDIF
                                        :  :  REWINDING-CONTINUE
                                        :  :            IF "TRE" BIT IN RH11/70 CS1 REGISTER=1
                                        :  :            :  THEN-PRINT ERROR 8
                                        :  :            :  ELSE-CONTINUE
                                        :  :           ENDIF
                                        :  :  TM FAULT B-PRINT ERROR 7
                                        :  :             DROP TM78-CLEAR ALL BITS (SCRATCH UNIT TABLE (TM78#))
                                        :  :  ALL OTHERS-PRINT ERROR 2
                                        :  :             DROP TU78-CLEAR BIT (SCRATCH UNIT TABLE(TM78#, TU78#))
                                        :  ENDSELECT
                                        ENDROUTINE

                                        -

004334' 004767  001610          M2.7:   CALL    M3.5            ;CALL MODULE 3.5 (REWIND CONTROL)
004340' 022767  000001  004102          CMP     #DONE,TINTCD    ;DONE?
004346' 001012                          BNE     2$              ;NO-CONTINUE
004350' 032777  040000  004516  1$:     BIT     #BIT14,@CS1     ;YES-IF IN RH11/70 CS1=1
004356' 001442                          BEQ     4$              ;ELSE-CONTINUE
                                                                ;THEN-PRINT ERROR 8
004360' 012767  000010  004072          MOV     #10,ERRINX
004366' 004767  002242                  CALL    M4.1
004372' 000434                          BR      4$

004374' 022767  000007  004046  2$:     CMP     #REW,TINTCD     ;REWINDING
004402' 001762                          BEQ     1$              ;YES-CHEAT-USE DONE CODE
004404' 022767  000032  004036          CMP     #TMFB,TINTCD    ;NO-FAULT B?
004412' 001012                          BNE     3$              ;NO-CONTINUE
004414' 012767  000007  004036          MOV     #7,ERRINX       ;YES - PRINT ERROR 7
004422' 004767  002206                  CALL    M4.1
004426' 016701  004000                  MOV     TMBIT2,R1       ;DROP TM78
004432' 005061  010554'                 CLR     SUTBL(R1)
004436' 000412                          BR      4$
004440' 012767  000002  004012  3$:     MOV     #2,ERRINX       ;PRINT ERROR 2
004446' 004767  002162                  CALL    M4.1
004452' 016701  003754                  MOV     TMBIT2,R1       ;DROP TU78
004456' 046761  003752  010554'         BIC     TUBIT,SUTBL(R1)
004464' 000207                  4$:     RTS     PC              ;RETURN
```



```
                                        .REM    -
                                        BEGINROUTINE    MODULE 2.8 (DATA COMPARE)
                                        :  IF INHIBIT DATA COMPARE FLAG (BIT0 OF SR1)=1
                                        :  :  THEN-CONTINUE
                                        :  :  ELSE-IF READ WAS A READ REVERSE
                                        :  :       :  THEN-COMPARE WRITE BUFFER WITH READ BUFFER (DEC/X11 CKDATA)
                                        :  :       :  ELSE-CONTINUE
                                        :  :      ENDIF
                                        :  ENDIF
                                        ENDROUTINE

                                        -

004466' 032767  000001  173322  M2.8:   BIT     #BIT0,SR1       ;IF INHIBIT DATA COMPARE FLAG=1
004474' 001010                          BNE     1$              ;THEN-CONTINUE
004476' 022767  000077  004002          CMP     #RREV,RDFCN     ;ELSE-WAS IT A READ REVERSE?
004504' 001404                          BEQ     1$              ;YES - THEN NO DATA COMPARE
004506' 104412  000000' 000126'         CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
 (1)  004514' 004516'                   .+2                     ; IF ERROR, CONTINUE
004516' 000207                  1$:     RTS     PC              ;RETURN

                                        ;NOTE:  IF ANYONE TRIES TO ADD A DATA COMPARE ON A READ REVERSE
                                        ;       OPERATION THERE WILL HAVE TO BE SOME WORK DONE WITH THE GETPA
                                        ;       AND MAP22 MACROS TO MAKE THE ADDRESSES APPEAR CORRECT FOR THE
                                        ;       READ REVERSE DATA COMPARE.  THE REASON BEING THAT THE DATA
                                        ;       COMPARE MACRO UTILIZES THE "RBUFPA" AND "RBUFEA" WORDS IN THE
                                        ;       HEADER AS A POINTER TO THE READ BUFFER.  HOWEVER, ON A READ
                                        ;       REVERSE OPERATION THESE WORDS POINT TO THE END OF THE BUFFER,
                                        ;       AND THE DATA COMPARE MACRO ASSUMES THAT "RBUFPA" AND "RBUFEA"
                                        ;       A STARTING BUFFER ADDRESSES.
```

```
                                .REM    -
                                BEGINROUTINE    MODULE 2.9 (STATISTIC PRINTOUT)
                                :  IF BIT3 OF SR1=0
                                :  :  THEN-PRINT STATISTIC HEADER LINE
                                :  :       CLEAR TM78#
                                :  :       BGNDO
                                :  :       :  CLEAR THE TU78#
                                :  :       :  BGNDO
                                :  :       :  :  IF BIT SET (UNIT TABLE(TM78#, TU78#))
                                :  :       :  :  :  THEN-LET INDEX=((TM78#(8))+(TU78#(2)))
                                :  :       :  :  :        GET (RECOVERABLE READ ERROR TABLE (INDEX))
                                :  :       :  :  :        GET (RECOVERABLE WRITE ERROR TABLE (INDEX))
                                :  :       :  :  :        GET UNRECOVERABLE READ ERROR TABLE (INDEX))
                                :  :       :  :  :        PRINT TM78#, TU78#, REC. READ, UN-REC. READ, REC. WRITE
                                :  :       :  :  :  ELSE-CONTINUE
                                :  :       :  :  ENDIF
                                :  :       :  :  INCREMENT THE TU78#
                                :  :       :  :  DO UNTIL THE TU78#=4
                                :  :       :  ENDDO
                                :  :       :  INCREMENT THE TM78#
                                :  :       :  DO UNTIL THE TM78#=10(8)
                                :  :       ENDDO
                                :  :  ELSE-CONTINUE
                                :  ENDIF
                                ENDROUTINE
                                -
004520' 032767  000010  173270  M2.9:   BIT     #BIT3,SR1       ;IF BIT3 OF SR1=0
004526' 001103                          BNE     4$              ;ELSE-CONTINUE
004530' 104401  000000' 004740'         MSG$,BEGIN,STATHD       ;ASCII MESSAGE CALL
004536' 005067  003700                  CLR     TM78            ;CLEAR THE TM78#
004542' 005067  003676          1$:     CLR     TU78            ;CLEAR THE TU78#
004546' 004767  174710                  CALL    TMCONV
004552' 004767  174746          2$:     CALL    TUCONV
004556' 016701  003650                  MOV     TMBIT2,R1
004562' 036761  003646  010514'         BIT     TUBIT,UTBL(R1)  ;IF BIT SET (UNIT TABLE(TM78#,TU78#))
004570' 001446                          BEQ     3$              ;ELSE-CONTINUE
004572' 016702  003634                  MOV     TMBIT2,R2       ;THEN-LET INDEX=((TM78#(8))+(TU78#(2)))
004576' 006302                          ASL     R2              ;TIMES 4
004600' 006302                          ASL     R2              ;TIMES 8
004602' 066702  003630                  ADD     TUBIT2,R2       ;ADD TU78# X 2
004606' 016267  010674' 003664          MOV     RREAD(R2),TEMP  ;GET THE RECOVERABLE READS
                                        ;*****************************************************************
 (1)                                                            ;CONVERT TEMP TO ASCII AND
 (1)                                                            ;STORE AT STRR
 (1)  004614' 104421  000000' 010500'   BTOD$,BEGIN,TEMP,STRR
 (1)  004622' 005072'
 (1)                                    ;*************************************************************
004624' 016267  010774' 003646          MOV     RWRIT(R2),TEMP  ;GET THE RECOVERABLE WRITES
                                        ;*****************************************************************
 (1)                                                            ;CONVERT TEMP TO ASCII AND
 (1)                                                            ;STORE AT STRW
 (1)  004632' 104421  000000' 010500'   BTOD$,BEGIN,TEMP,STRW
 (1)  004640' 005131'
 (1)                                    ;*************************************************************
004642' 016267  010574' 003630          MOV     URREAD(R2),TEMP ;GET THE UNRECOVERABLE READS
```



```
                                        ;*************************************************************
 (1)                                                            ;CONVERT TEMP TO ASCII AND
 (1)                                                            ;STORE AT STURR
 (1)  004650' 104421  000000' 010500'   BTOD$,BEGIN,TEMP,STURR
 (1)  004656' 005111'
 (1)                                    ;*************************************************************
                                        ;*****************************************************************
 (1)                                                            ;CONVERT TM78 TO ASCII AND
 (1)                                                            ;STORE AT STTM
 (1)  004660' 104420  000000' 010442'   OTOA$,BEGIN,TM78,STTM
 (1)  004666' 005046'
 (1)                                    ;*************************************************************
                                        ;*****************************************************************
 (1)                                                            ;CONVERT TU78 TO ASCII AND
 (1)                                                            ;STORE AT STTU
 (1)  004670' 104420  000000' 010444'   OTOA$,BEGIN,TU78,STTU
 (1)  004676' 005057'
 (1)                                    ;*************************************************************
004700' 104401  000000' 005044'         MSG$,BEGIN,STATLN       ;ASCII MESSAGE CALL
004706' 005267  003532          3$:     INC     TU78            ;INCREMENT THE TU78#
004712' 022767  000004  003524          CMP     #4,TU78         ;DO UNTIL THE TU78#=4
004720' 001314                          BNE     2$
004722' 005267  003514                  INC     TM78            ;INCREMENT THE TM78#
004726' 022767  000010  003506          CMP     #10,TM78        ;DO UNTIL THE TM78#=10
004734' 001302                          BNE     1$
004736' 000207                  4$:     RTS     PC              ;RETURN
004740' 020045  020040  046524  STATHD: .ASCII  '%   TM78#    TU78#    REC. READS    UN-REC. READS    '
004746' 034067  020043  020040
004754' 052040  033525  021470
004762' 020040  020040  042522
004770' 027103  051040  040505
004776' 051504  020040  020040
005004' 047125  051055  041505
005012' 020056  042522  042101
005020' 020123  020040     040
005025'     122 041505  020056          .ASCIZ  'REC. WRITES %'
005032' 051127  052111  051505
005040' 022440     000
                005044'              .EVEN
005044' 020040                  STATLN: .ASCII  '  '
005046' 030060  030060  030060  STTM:   .ASCII  '000000'
005054' 020040     040                  .ASCII  '  '
005057'     060 030060  030060  STTU:   .ASCII  '000000'
005064'     060
005065'     040 020040  020040          .ASCII  '     '
005072' 030060  030060  027060  STRR:   .ASCII  '00000.'
005100' 020040  020040  020040          .ASCII  '          '
005106' 020040     040
005111'     060 030060  030060  STURR:  .ASCII  '00000.'
005116'     056
005117'     040 020040  020040          .ASCII  '          '
005124' 020040  020040     040
005131'     060 030060  030060  STRW:   .ASCII  '00000.'
005136'     056
005137'     045     000                 .ASCIZ  '%'
                005142'              .EVEN
```

```
2307                                    .REM    -
2308                                    BEGINROUTINE    MODULE 2.10 (UNIT CONTROL)
2309                                    :  CLEAR THE UNIT COUNT
2310                                    :  CLEAR THE EOT COUNT
2311                                    :  CLEAR THE TM78#
2312                                    :  BGNDO
2313                                    :  :  CLEAR THE TU78#
2314                                    :  :  BGNDO
2315                                    :  :  :  IF BIT SET (SCRATCH UNIT TABLE(TM78#, TU78#))
2316                                    :  :  :  :  THEN-INCREMENT THE UNIT COUNT
2317                                    :  :  :  :  ELSE-CONTINUE
2318                                    :  :  :  ENDIF
2319                                    :  :  :  IF BIT SET (EOT TABLE(TM78#, TU78#))
2320                                    :  :  :  :  THEN-INCREMENT THE EOT COUNT
2321                                    :  :  :  :  ELSE-CONTINUE
2322                                    :  :  :  ENDIF
2323                                    :  :  :  INCREMENT THE TU78#
2324                                    :  :  :  DO UNTIL THE TU78#=4
2325                                    :  :  ENDDO
2326                                    :  :  INCREMENT THE TM78#
2327                                    :  :  DO UNTIL THE TM78#=10(8)
2328                                    :  ENDDO
2329                                    ENDROUTINE
2330                                    -
2331
2332  005142' 005067  003272            M2.10:  CLR     UCNT            ;CLEAR THE UNIT COUNT
2333  005146' 005067  003274                    CLR     EOTCNT          ;CLEAR THE EOT COUNT
2334  005152' 005067  003264                    CLR     TM78            ;CLEAR THE TM78 #
2335  005156' 005067  003262            1$:     CLR     TU78            ;CLEAR THE TU78#
2336  005162' 004767  174274                    CALL    TMCONV
2337  005166' 004767  174332            2$:     CALL    TUCONV
2338  005172' 016701  003234                    MOV     TMBIT2,R1       ;
2339  005176' 036761  003232  010554'           BIT     TUBIT,SUTBL(R1) ;IF SCRATCH UNIT TABLE (TM78#,TU78#)=1
2340  005204' 001402                            BEQ     3$              ;ELSE-CONTINUE
2341  005206' 005267  003226                    INC     UCNT            ;THEN-INCREMENT THE UNIT COUNT
2342  005212' 036761  003216  010534' 3$:      BIT     TUBIT,EOTTBL(R1)  ;IF EOT TABLE (TM78#,TU78#)=1
2343  005220' 001402                            BEQ     4$              ;ELSE-CONTINUE
2344  005222' 005267  003220                    INC     EOTCNT          ;THEN-INCREMENT THE EOT COUNT
2345  005226' 005267  003212            4$:     INC     TU78            ;INCREMENT THE TU78#
2346  005232' 022767  000004  003204            CMP     #4,TU78         ;DO UNTIL THE TU78#=4
2347  005240' 001352                            BNE     2$
2348  005242' 005267  003174                    INC     TM78            ;INCREMENT THE TM78#
2349  005246' 022767  000010  003166            CMP     #10,TM78        ;DO UNTIL THE TM78#=10
2350  005254' 001340                            BNE     1$
2351  005256' 000207                            RTS     PC              ;RETURN
```



```
2353                                    .REM    -
2354                                    BEGINROUTINE    MODULE 3.1 (WRITE)
2355                                    :  LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
2356                                    :  LOAD THE RH11/70 CS1 REGISTER WITH THE "TRE" BIT (040000)
2357                                    :  CLEAR THE RH11/70 CS1 REGISTER
2358                                    :  LOAD THE RH11/70 WC REGISTER WITH 2'S COMPLEMENT OF BUFFER SIZE (DEC/X11 WBUFSZ)
2359                                    :  LOAD THE RH11/70 BC REGISTER WITH (WBUFSZ*2)
2360                                    :  CLEAR THE TM78 DATA TRANSFER INTERRUPT CODE
2361                                    :  LOAD THE RH11/70 REGISTER 14 WITH (TU78# * FORMAT * RECORD COUNT * SKIP COUNT)
2362                                    :  IF 22 BIT ADDRESSING AVAILABLE (DEC/X11 BIT9 OF RES1)
2363                                    :  :  THEN-CONVERT THE 18 BIT WRITE BUFFER ADDRESS TO 22 BITS (DEC/X11 MAP22)
2364                                    :  :        LOAD THE RH70 BA REGISTER
2365                                    :  :        LOAD THE RH70 EXTENDED BA REGISTER
2366                                    :  :  ELSE-LOAD THE RH11 BA REGISTER
2367                                    :  ENDIF
2368                                    :  SET THE DATA TRANSFER AND INTERRUPT EXPECTED FLAGS
2369                                    :  IF DENSITY FLAG=1 (GCR)
2370                                    :  :  THEN-LOAD RH11/70 CS1 WITH (IE * WRITE GCR FUNCTION CODE * A16 * A17)
2371                                    :  :  ELSE-LOAD RH11/70 CS1 WITH (IE * WRITE  PE FUNCTION CODE * A16 * A17)
2372                                    :  ENDIF
2373                                    :  WAIT FOR AN INTERRUPT (DEC/X11 EXIT)
2374                                    ENDROUTINE
2375                                    -
2376
2377  005260' 016777  003156  003616    M3.1:   MOV     TM78,@CS2       ;LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
2378  005266' 016777  040000  003600            MOV     BIT14,@CS1      ;LOAD THE RH11/70 CS1 REGISTER WITH THE TRE BIT
2379  005274' 005077  003574                    CLR     @CS1            ;CLEAR RH11/70 CS1
2380  005300' 016777  172636  003570            MOV     WBUFSZ,@WC      ;LOAD THE RH11/70 WC REGISTER WITH
2381  005306' 005477  003564                    NEG     @WC             ;THE 2'S COMPLEMENT OF BUFFER SIZE
2382  005312' 016777  172624  003562            MOV     WBUFSZ,@BC      ;LOAD THE RH11/70 BC REGISTER WITH
2383  005320' 006377  003556                    ASL     @BC             ;THE WORD COUNT * 2
2384  005324' 005077  003556                    CLR     @XFRINT         ;CLEAR THE DATA TRANSFER INTERRUPT CODE
2385  005330' 016777  003110  003552            MOV     TU78,@TC        ;LOAD RH11/70 REGISTER 14 WITH TU78#, FORMAT REC. CNT. S
2386  005336' 032767  001000  172512            BIT     #ADDR22,RES1    ;IF 22 BIT ADDRESSING AVAILABLE
2387  005344' 001430                            BEQ     1$              ;GO DO ELSE
2388  005346' 016767  172562  003110            MOV     WBUFPA,PA18     ;THEN-CONVERT THE 18 BIT
2389  005354' 016767  172556  003104            MOV     WBUFEA,XMEM     ;PHYSICAL ADDRESS TO 22 BITS
2390  005362' 104416  000000' 010464'           MAP22$, BEGIN,PA18              ; GET 22-BIT ADDR FROM 18-BIT ADDR
2391  005370' 016777  003074  003502            MOV     PA22,@BA        ;LOAD THE RH70 BA REGISTER
2392  005376' 016777  003070  003564            MOV     EA22,@EBA       ;LOAD THE RH70 EBA REGISTER
2393  005404' 016767  003062  003050            MOV     EA22,A16A17     ;SAVE ADDRESS BITS A16 AND A17
2394  005412' 042767  177774  003042            BIC     #177774,A16A17
2395  005420' 000367  003036                    SWAB    A16A17
2396  005424' 000416                            BR      2$              ;
2397  005426' 016777  172502  003444  1$:       MOV     WBUFPA,@BA      ;ELSE-LOAD THE RH11 BA REGISTER
2398  005434' 016767  172476  003020            MOV     WBUFEA,A16A17   ;CALCULATE ADDRESS BITS A16 AND A17
2399  005442' 006367  003014                    ASL     A16A17          ;SHIFT EXTENDED ADDRESS
2400  005446' 006367  003010                    ASL     A16A17          ;BITS TO BIT POSITIONS
2401  005452' 006367  003004                    ASL     A16A17          ;8 AND 9
2402  005456' 006367  003000                    ASL     A16A17          ;
2403  005462' 052767  000050  003004  2$:       BIS     #BIT3!BIT5,FLAGS  ;SET THE DATA TRANSFER AND INTERRUPT EXPECTED FLAGS
2404  005470' 032767  000001  002776            BIT     #BIT0,FLAGS     ;IF DENSITY FLAG=1 (GCR)
2405  005476' 001403                            BEQ     3$              ;GO DO ELSE
2406  005500' 012701  000163                    MOV     #WGCR!BIT6,R1   ;THEN-LOAD WRITE GCR FUNCTION CODE AND IE BIT
2407  005504' 000402                            BR      4$              ;
2408  005506' 012701  000161            3$:     MOV     #WPE!BIT6,R1    ;ELSE-LOAD WRITE PE FUNCTION CODE AND IE BIT
```

```
2409  005512' 056701  002744          4$:     BIS     A16A17,R1       ;OR IN EXTENDED ADDRESS BITS
2410  005516' 010177  003352                  MOV     R1,@CS1         ;ISSUE THE COMMAND
2411  005522' 104400  000000'                 EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
2412
2413
2414                                  .REM    -
2415                                  BEGINROUTINE    MODULE 3.2 (READ FORWARD)
2416                                  :  LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
2417                                  :  LOAD THE RH11/70 CS1 REGISTER WITH THE "TRE" BIT (040000)
2418                                  :  CLEAR THE RH11/70 CS1 REGISTER
2419                                  :  LOAD THE RH11/70 WC REGISTER WITH 2'S COMPLEMENT OF BUFFER SIZE (DEC/X11 WBUFSZ)
2420                                  :  LOAD THE RH11/70 BC REGISTER WITH (WBUFSZ*2)
2421                                  :  CLEAR THE TM78 DATA TRANSFER INTERRUPT CODE
2422                                  :  LOAD THE RH11/70 REGISTER 14 WITH (TU78# * FORMAT * RECORD COUNT * SKIP COUNT)
2423                                  :  CONVERT THE READ BUFFER ADDRESS TO 18 BITS (DEC/X11 GETPA)
2424                                  :  IF 22 BIT ADDRESSING AVAILABLE (DEC/X11 BIT9 OF RES1)
2425                                  :  :  THEN-CONVERT 18 BIT READ BUFFER ADDRESS TO 22 BITS (DEC/X11 MAP22)
2426                                  :  :        LOAD THE RH70 BA REGISTER
2427                                  :  :        LOAD THE RH70 EXTENDED BA REGISTER
2428                                  :  :  ELSE-LOAD THE RH11 BA REGISTER
2429                                  :  ENDIF
2430                                  :  SET THE DATA TRANSFER AND INTERRUPT EXPECTED FLAGS
2431                                  :  LOAD RH11/70 CS1 WITH (IE * READ FORWARD FUNCTION CODE * A17 * A18)
2432                                  :  WAIT FOR AN INTERRUPT (DEC/X11 EXIT)
2433                                  ENDROUTINE
2434                                  -
2435
2436  005526' 012767  000071  002752  M3.2:   MOV     #RFWD,RDFCN
2437  005534' 012767  011174' 172362          MOV     #BUFIN,RBUFVA
2438
2439  005542' 016777  002674  003334  MRDCOM: MOV     TM78,@CS2       ;LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
2440  005550' 012777  040000  003316          MOV     #BIT14,@CS1     ;LOAD THE RH11/70 CS1 REGISTER WITH THE TRE BIT
2441  005556' 005077  003312                  CLR     @CS1            ;CLEAR THE RH11/70 CS1 REGISTER
2442  005562' 016777  172354  003306          MOV     WBUFSZ,@WC      ;LOAD THE RH11/70 WC REGISTER WITH THE
2443  005570' 005477  003302                  NEG     @WC             ;2'S COMPLEMENT OF BUFFER SIZE
2444  005574' 016777  172342  003300          MOV     WBUFSZ,@BC      ;LOAD THE RH11/70 BC REGISTER WITH BUFFER SIZE
2445  005602' 006377  003274                  ASL     @BC             ;WORD COUNT * 2
2446  005606' 005077  003274                  CLR     @XFRINT         ;CLEAR THE RM78 DATA TRANSFER INTERRUPT CODE
2447  005612' 016777  002626  003270          MOV     TU78,@TC        ;LOAD RH11/70 REG. 14 WITH TU78#,FMT,REC-CNT,SKIP-CNT
2448  005620' 104415  000000' 000124'        GETPAS,BEGIN, RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
2449  005626' 032767  001000  172222          BIT     #ADDR22,RES1    ;IF 22 BIT ADDRESSING AVAILABLE
2450  005634' 001430                          BEQ     1$              ;GO DO ELSE
2451  005636' 016767  172264  002620          MOV     RBUFPA,PA18     ;THEN-LOAD UP THE 18 BIT
2452  005644' 016767  172260  002614          MOV     RBUFEA,XMEM     ;ADDRESS FOR CONVERSION
2453  005652' 104416  000000' 010464'        MAP22$, BEGIN,PA18              ; GET 22-BIT ADDR FROM 18-BIT ADDR
2454  005660' 016777  002604  003212          MOV     PA22,@BA        ;LOAD THE RH70 BA REGISTER
2455  005666' 016777  002600  003274          MOV     EA22,@EBA       ;LOAD THE RH70 EBA REGISTER
2456  005674' 016767  002572  002560          MOV     EA22,A16A17     ;CALCULATE A16 + A17 FOR LATER
2457  005702' 042767  177774  002552          BIC     #177774,A16A17  ;USE IN CS1
2458  005710' 000367  002546                  SWAB    A16A17          ;
2459  005714' 000416                          BR      2$              ;
2460  005716' 016777  172204  003154  1$:     MOV     RBUFPA,@BA      ;ELSE-LOAD THE RH11 BA REGISTER
2461  005724' 016767  172200  002530          MOV     RBUFEA,A16A17   ;CALCULATE A16 + A17 FOR LATER
2462  005732' 006367  002524                  ASL     A16A17          ;USE IN CS1
2463  005736' 006367  002520                  ASL     A16A17          ;
2464  005742' 006367  002514                  ASL     A16A17          ;
```



```
2465  005746' 006367  002510                  ASL     A16A17          ;
2466  005752' 052767  000050  002514  2$:     BIS     #BIT3!BIT5,FLAGS  ;SET THE DATA TRANSFER AND INTERRUPT EXPECTED FLAGS
2467  005760' 016701  002522                  MOV     RDFCN,R1        ;GET THE READ COMMAND
2468  005764' 052701  000100                  BIS     #BIT6,R1        ;OR IN THE IE BIT
2469  005770' 056701  002466                  BIS     A16A17,R1       ;OR IN THE EXTENDED ADDRESS BITS
2470  005774' 010177  003074                  MOV     R1,@CS1         ;ISSUE THE COMMAND
2471  006000' 104400  000000'                 EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
2472                                  .REM    -
2473                                  BEGINROUTINE    MODULE 3.3 (READ REVERSE)
2474                                  :  LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
2475                                  :  LOAD THE RH11/70 CS1 REGISTER WITH THE "TRE" BIT (040000)
2476                                  :  CLEAR THE RH11/70 CS1 REGISTER
2477                                  :  LOAD THE RH11/70 WC REGISTER WITH 2'S COMPLEMENT OF BUFFER SIZE (DEC/X11 WBUFSZ)
2478                                  :  LOAD THE RH11/70 BC REGISTER WITH (WBUFSZ*2)
2479                                  :  CLEAR THE TM78 DATA TRANSFER INTERRUPT CODE
2480                                  :  LOAD THE RH11/70 REGISTER 14 WITH (TU78# * FORMAT * RECORD COUNT * SKIP COUNT)
2481                                  :  CONVERT THE READ BUFFER ADDRESS + "RBUFSZ" TO 18 BITS (DEC/X11 GETPA)
2482                                  :  IF RH70 SWITCH=0 (RH70)
2483                                  :  :  THEN-CONVERT 18 BIT READ BUFFER ADDRESS TO 22 BITS (DEC/X11 MAP22)
2484                                  :  :        LOAD THE RH70 BA REGISTER
2485                                  :  :        LOAD THE RH70 EXTENDED BA REGISTER
2486                                  :  :  ELSE-LOAD THE RH11 BA REGISTER
2487                                  :  ENDIF
2488                                  :  SET THE DATA TRANSFER AND INTERRUPT EXPECTED FLAGS
2489                                  :  LOAD RH11/70 CS1 WITH (IE * READ REVERSE FUNCTION CODE * A17 * A18)
2490                                  :  WAIT FOR AN INTERRUPT (DEC/X11 EXIT)
2491                                  ENDROUTINE
2492                                  -
2493
2494  006004' 012767  000077  002474  M3.3:   MOV     #RREV,RDFCN     ;
2495  006012' 012767  011174' 172104          MOV     #BUFIN,RBUFVA
2496  006020' 066767  172116  172076          ADD     WBUFSZ,RBUFVA
2497  006026' 066767  172110  172070          ADD     WBUFSZ,RBUFVA
2498  006034' 162767  000002  172062          SUB     #2,RBUFVA       ;START ONE WORD BEFORE END
2499  006042' 000167  177474                  JMP     MRDCOM
```

```
                                        .REM    -
                                        BEGINROUTINE    MODULE 3.4 (TAPE MARK)
                                        :  LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
                                        :  LOAD THE RH11/70 CS1 REGISTER WITH THE "TRE" BIT (040000)
                                        :  CLEAR THE RH11/70 CS1 REGISTER
                                        :  CLEAR THE DATA TRANSFER FLAG
                                        :  SET THE INTERRUPT EXPECTED FLAG
                                        :  IF DENSITY FLAG=1 (GCR)
                                        :  :  THEN-LOAD RH11/70 REGISTER (40+(TU78#*2)) WITH WTM GCR FUNCTION CODE
                                        :  :  ELSE-LOAD RH11/70 REGISTER (40+(TU78#*2)) WITH WTM PE FUNCTION CODE
                                        :  ENDIF
                                        :  LOAD RH11/70 CS1 REGISTER WITH (IE)
                                        :  WAIT FOR AN INTERRUPT (DEC/X11 EXIT)
                                        ENDROUTINE
                                        -

006046' 016777  002370  003030  M3.4:   MOV     TM78,@CS2       ;LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
006054' 012777  040000  003012          MOV     #BIT14,@CS1     ;LOAD THE RH11/70 CS1 REGISTER WITH THE TRE BIT
006062' 005077  003006                  CLR     @CS1            ;CLEAR THE RH11/70 CS1 REGISTER
006066' 042767  000010  002400          BIC     #BIT3,FLAGS     ;CLEAR THE DATA TRANSFER FLAG
006074' 052767  000040  002372          BIS     #BIT5,FLAGS     ;SET THE INTERRUPT EXPECTED FLAG
006102' 016701  002336                  MOV     TU78,R1         ;GET THE TU78#
006106' 006301                          ASL     R1              ;MULTIPLY BY 2
006110' 032767  000001  002356          BIT     #BIT0,FLAGS     ;IF DENSITY FLAG=1
006116' 001404                          BEQ     1$              ;GO DO ELSE
006120' 012771  000017  011134'         MOV     #GCRTM,@MOO(R1) ;THEN-LOAD RH11/70 REGISTER (40+(TU78#*2)) WITH WTM GCR
006126' 000403                          BR      2$
006130' 012771  000015  011134' 1$:     MOV     #PETM,@MOO(R1)  ;ELSE=LOAD RH11/70 REGISTER (R0+(TU78#*2)) WITH WTM PE F
006136' 012777  000100  002730  2$:     MOV     #BIT6,@CS1      ;LOAD RH11/70 CS1 WITH THE IE BIT
006144' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```



```
                                        .REM    -
                                        BEGINROUTINE    MODULE 3.5 (REWIND)
                                        :  LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
                                        :  LOAD THE RH11/70 CS1 REGISTER WITH THE "TRE" BIT (040000)
                                        :  CLEAR THE RH11/70 CS1 REGISTER
                                        :  CLEAR THE DATA TRANSFER FLAG
                                        :  SET THE INTERRUPT EXPECTED FLAG
                                        :  LOAD THE RH11/70 REGISTER (40+(TU78#*2)) WITH REWIND FUNCTION CODE
                                        :  LOAD THE RH11/70 CS1 REGISTER WITH (IE)
                                        :  WAIT FOR AN INTERRUPT (DEC/X11 EXIT)
                                        ENDROUTINE
                                        -

006150' 016777  002266  002726  M3.5:   MOV     TM78,@CS2       ;LOAD THE RH11/70 CS2 REGISTER WITH TM78#
006156' 012777  040000  002710          MOV     #BIT14,@CS1     ;LOAD THE RH11/70 CS1 REGISTER WITH THE TRE BIT
006164' 005077  002704                  CLR     @CS1            ;CLEAR THE RH11/70 CS1 REGISTER
006170' 042767  000010  002276          BIC     #BIT3,FLAGS     ;CLEAR THE DATA TRANSFER FLAG
006176' 052767  000040  002270          BIS     #BIT5,FLAGS     ;SET THE INTERRUPT EXPECTED FLAG
006204' 016701  002234                  MOV     TU78,R1         ;GET THE TU78#
006210' 006301                          ASL     R1              ;MULTIPLY BY 2
006212' 012771  000007  011134'         MOV     #REW,@MOO(R1)   ;LOAD RH11/70 REGISTER (40+(TU78#*2)) WITH THE REWIND FU
006220' 012777  000100  002646          MOV     #BIT6,@CS1      ;LOAD RH11/70 REGISTER CS1 WITH THE IE BIT
006226' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```

```
                                        .REM    -
                                        BEGINROUTINE    MODULE 3.6 (SENSE)
                                        :  LOAD THE RH11/70 CS2 REGISTER WITH THE TM78#
                                        :  LOAD THE RH11/70 CS1 REGISTER WITH THE "TRE" BIT (040000)
                                        :  CLEAR THE RH11/70 CS1 REGISTER
                                        :  CLEAR THE DATA TRANSFER FLAG
                                        :  SET THE INTERRUPT EXPECTED FLAG
                                        :  LOAD THE RH11/70 REGISTER (40+(TU78#*2)) WITH SENSE FUNCTION CODE
                                        :  LOAD RH11/70 CS1 REGISTER WITH (IE)
                                        :  WAIT FOR AN INTERRUPT (DEC/X11 EXIT)
                                        ENDROUTINE
                                        -
006232' 016777  002204  002644  M3.6:   MOV     TM78,@CS2       ;LOAD THE RH11/70 CS2 REGISTER WITH TM78#
006240' 012777  040000  002626          MOV     #BIT14,@CS1     ;LOAD THE RH11/70 CS1 REGISTER WITH THE TRE BIT
006246' 005077  002622                  CLR     @CS1            ;CLEAR THE RH11/70 CS1 REGISTER
006252' 042767  000010  002214          BIC     #BIT3,FLAGS     ;CLEAR THE DATA TRANSFER FLAG
006260' 052767  000040  002206          BIS     #BIT5,FLAGS     ;SET THE INTERRUPT EXPECTED FLAG
006266' 016701  002152                  MOV     TU78,R1         ;GET THE TU78#
006272' 006301                          ASL     R1              ;MULTIPLY BY 2
006274' 012771  000011  011134'         MOV     #SEN,@M00(R1)   ;LOAD THE RH11/70 REGISTER (R0+(TU78#*2)) WITH THE SENSE
006302' 012777  000100  002564          MOV     #BIT6,@CS1      ;LOAD RH11/70 REGISTER CS1 WITH THE IE BIT
006310' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```



```
                                        .REM    -
                                        BEGINROUTINE    MODULE 4.0 (INTERRUPT)
                                        -
006314' 042777  000100  002552  M4.0:   BIC     #BIT6,@CS1      ;CLEAR THE RH11/70 IE BIT
006322' 016767  002546  171550          MOV     CS1,CSRA        ;COPY CSR ADDRESS FOR ERROR PRINTOUT
006330' 017767  002540  171544          MOV     @CS1,ACSR       ;COPY CONTENTS OF CS1 TO HEADER
006336' 017767  002542  171540          MOV     @CS2,ASTAT      ;COPY CS2 CONTENTS
006344' 032767  000040  002122          BIT     #BIT5,FLAGS     ;INTERRUPT EXPECTED?
006352' 001007                          BNE     2$              ;YES-GO PROCESS
006354' 017777  002532  002530          MOV     @AS,@AS         ;NO-CLEAR PENDING INTERRUPTS
006362' 012777  000100  002504  1$:     MOV     #BIT6,@CS1      ;ENABLE-RH11/70 INTERRUPTS
006370' 000002                          RTI                     ;RETURN

006372' 032767  000010  002074  2$:     BIT     #BIT3,FLAGS     ;DATA TRANSFER INTERRUPT EXPECTED
006400' 001406                          BEQ     3$              ;NO-
006402' 042767  000040  002064          BIC     #BIT5,FLAGS     ;YES-CLEAR INTERRUPT EXPECTED FLAG
                                        ;-------------------------------------------------------------
 (1) 006410' 000004  000000' 006532'    PIRQ$,BEGIN,M4.0C               ; QUEUE UP TO CONTINUE AT M4.0C AND RTI
 (1)                                    ;-------------------------------------------------------------

006416' 017767  002470  002066  3$:     MOV     @AS,ASREG       ;COPY THE RH11/70 AS REGISTER
006424' 036767  002000  002060          BIT     TMBIT,ASREG     ;EXPECTED TM?
006432' 001004                          BNE     5$              ;YES-GO CHECK TU
006434' 016777  002052  002450  4$:     MOV     ASREG,@AS       ;NO-CLEAR THE INTERRUPT(S)
006442' 000747                          BR      1$

006444' 016777  001772  002432  5$:     MOV     TM78,@CS2       ;SELECT INTERRUPTING TM78
006452' 017767  002454  002030          MOV     @MOINT,MONINR   ;SAVE INTERRUPT CODE
006460' 017767  002430  001770          MOV     @TUSTAT,SENSE   ;SAVE SENSE INFO.
006466' 017767  002440  001756          MOV     @MOINT,ITU78    ;GET THE TU78 #
006474' 000367  001752                  SWAB    ITU78
006500' 042767  177774  001744          BIC     #177774,ITU78
006506' 026767  001732  001736          CMP     TU78,ITU78      ;EXPECTED TU?
006514' 001347                          BNE     4$              ;NO-GO CLEAR INTERRUPT
006516' 042767  000040  001750          BIC     #BIT5,FLAGS     ;YES-CLEAR INTERRUPT EXPECTED FLAG
                                        ;-------------------------------------------------------------
 (1) 006524' 000004  000000' 006532'    PIRQ$,BEGIN,M4.0C               ; QUEUE UP TO CONTINUE AT M4.0C AND RTI
 (1)                                    ;-------------------------------------------------------------

006532' 032767  000010  001734  M4.0C:  BIT     #BIT3,FLAGS     ;IF DATA TRANSFER FLAG=1
006540' 001412                          BEQ     1$              ;GO DO ELSE
006542' 016777  001674  002334          MOV     TM78,@CS2       ;THEN-LOAD TM78 # TO RH11/70
006550' 017767  002332  001672          MOV     @XFRINT,TINTCD  ;COPY DATA TRANSFER INTERRUPT
006556' 042767  177700  001664          BIC     #177700,TINTCD  ;CODE TO TERMINATION INTERRUPT CODE
006564' 000411                          BR      2$

006566' 016767  001716  001654  1$:     MOV     MONINR,TINTCD   ;GET THE NON DATA
006574' 042767  177700  001646          BIC     #177700,TINTCD  ;TRANSFER INTERRUPT CODE
006602' 016777  001622  002302          MOV     TMBIT,@AS
006610' 042767  000010  001656  2$:     BIC     #BIT3,FLAGS     ;CLEAR DATA TRANSFER FLAG
006616' 042767  000040  001650          BIC     #BIT5,FLAGS     ;CLEAR INTERRUPT EXPECTED FLAG
006624' 012777  000100  002242          MOV     #BIT6,@CS1      ;ENABLE RH11/70 INTERRUPTS
006632' 000207                          RTS     PC              ;RETURN
```

```
                                        .REM    -
                                        ;*BEGINROUTINE  MODULE 4.1 (ERRORS)
                                        -
006634' 032767  000004  171154  M4.1:   BIT     #BIT2,SR1       ;IF BIT2 OF SR1=0 (PRINT ERROR MESSAGES)
006642' 001063                          BNE     3$              ;ELSE - GET OUT
006644' 032767  000001  001622          BIT     #BIT0,FLAGS     ;IF GCR DENSITY?
006652' 001407                          BEQ     1$              ;ELSE-GO DO PE
006654' 012702  000107                  MOV     #107,R2         ;THEN-LOAD GCR CHARACTERS
006660' 012703  000103                  MOV     #103,R3
006664' 012704  000122                  MOV     #122,R4
006670' 000406                          BR      2$
006672' 012702  000120          1$:     MOV     #120,R2         ;ELSE-LOAD PE CHARACTERS
006676' 012703  000105                  MOV     #105,R3
006702' 012704  000040                  MOV     #040,R4
006706' 110267  000354          2$:     MOVB    R2,TYPE1        ;STORE THE FORMAT
006712' 110267  000442                  MOVB    R2,TYPE2        ;TYPE IN THE
006716' 110267  000521                  MOVB    R2,TYPE3        ;ERROR PRINTOUT
006722' 110267  000601                  MOVB    R2,TYPE4        ;CHARACTER STRINGS
006726' 110367  000335                  MOVB    R3,TYPE1+1
006732' 110367  000423                  MOVB    R3,TYPE2+1
006736' 110367  000502                  MOVB    R3,TYPE3+1
006742' 110367  000562                  MOVB    R3,TYPE4+1
006746' 110467  000316                  MOVB    R4,TYPE1+2
006752' 110467  000404                  MOVB    R4,TYPE2+2
006756' 110467  000463                  MOVB    R4,TYPE3+2
006762' 110467  000543                  MOVB    R4,TYPE4+2
006766' 005067  171114                  CLR     ERRTYP          ;CLEAR ERRTYPE
006772' 016702  001462                  MOV     ERRINX,R2       ;GET ERROR INDEX
006776' 006302                          ASL     R2              ;MULTIPLY BY 2
007000' 004772  007014'                 CALL    @M4.1TB(R2)     ;GO PRINT PROPER ERROR
007004' 104401  000000' 010334'         MSG$,BEGIN,CRLF ;ASCII MESSAGE CALL
007012' 000207          3$:     RTS     PC              ;RETURN

007014' 000000          M4.1TB: .WORD   0               ;UNDEFINED
007016' 007044'                 .WORD   ERR1            ;NO TM78 UNITS AVAILABLE
007020' 007114'                 .WORD   ERR2            ;UNEXPECTED INTERRUPT CODE
007022' 007216'                 .WORD   ERR3            ;RECOVERABLE WRITE ERROR
007024' 007302'                 .WORD   ERR4            ;UNRECOVERABLE WRITE ERROR
007026' 007374'                 .WORD   ERR5            ;RECOVERABLE READ ERROR
007030' 007456'                 .WORD   ERR6            ;UNRECOVERABLE READ ERROR
007032' 007542'                 .WORD   ERR7            ;TM FAULT INTERRUPT
007034' 007614'                 .WORD   ERR10           ;MASSBUS STATUS ERROR
007036' 007760'                 .WORD   ERR11           ;NO TU78 UNITS AVAILABLE
007040' 010030'                 .WORD   ERR12           ;BOTH DENSITIES DISABLED
007042' 010100'                 .WORD   ERR13           ;TM78 NOT READY

007044'                 ERR1:
 (1)                            ;*****************************************************************
 (1)  007044' 104405  000000' 000000    HRDER$,BEGIN,NULL       ;
 (1)                            ;*****************************************************************
007052' 104401  000000' 007062'         MSG$,BEGIN,ERMS1        ;ASCII MESSAGE CALL
007060' 000207                          RTS     PC              ;RETURN
007062' 047516  052040  033515  ERMS1:  .ASCIZ  'NO TM78 UNITS AVAILABLE%'
007070' 020070  047125  052111
007076' 020123  053101  044501
```



```
007104' 040514  046102  022505
007112'    000
                007114'                 .EVEN

007114'                 ERR2:
 (1)                            ;*****************************************************************
 (1)  007114' 104405  000000' 000000    HRDER$,BEGIN,NULL       ;
 (1)                            ;*****************************************************************
                                ;*****************************************************************
                                        ;CONVERT TINTCD TO ASCII AND
 (1)                                    ;STORE AT ERMS2A
 (1)  007122' 104420  000000' 010450'   OTOA$,BEGIN,TINTCD,ERMS2A
 (1)  007130' 007205'
 (1)                            ;*****************************************************************
007132' 104401  000000' 007152'         MSG$,BEGIN,ERMS2        ;ASCII MESSAGE CALL
007140' 004767  001014                  CALL    UNITS           ;GO PRINT TM78#-TU78#
007144' 004767  001110                  CALL    DROPTU          ;GO DROP UNIT MESSAGE
007150' 000207                          RTS     PC              ;RETURN
007152' 047125  054105  042520  ERMS2:  .ASCII  'UNEXPECTED INTERRUPT CODE  '
007160' 052103  042105  044440
007166' 052116  051105  052522
007174' 052120  041440  042117
007202' 020105     040
007205'    060  030060  030060  ERMS2A: .ASCIZ  '000000%'
007212' 022460     000
                007216'                 .EVEN

007216'                 ERR3:
 (1)                            ;*****************************************************************
 (1)  007216' 104406  000000' 000000    SOFER$,BEGIN,NULL       ;
 (1)                            ;*****************************************************************
007224' 104401  000000' 007244'         MSG$,BEGIN,ERMS3        ;ASCII MESSAGE CALL
007232' 004767  000722                  CALL    UNITS           ;PRINT TM78#-TU78#
007236' 004767  001074                  CALL    RECFIL          ;PRINT RECORD NUMBER FILE NUMBER
007242' 000207                          RTS     PC              ;RETURN
007244' 042522  047503  042526  ERMS3:  .ASCII  'RECOVERABLE WRITE '
007252' 040522  046102  020105
007260' 051127  052111  020105
007266' 041507     122  TYPE1:  .ASCII  'GCR'
007271'    040  051105  047522          .ASCIZ  ' ERROR%'
007276' 022522     000
                007302'                 .EVEN

007302'                 ERR4:
 (1)                            ;*****************************************************************
 (1)  007302' 104405  000000' 000000    HRDER$,BEGIN,NULL       ;
 (1)                            ;*****************************************************************
007310' 104401  000000' 007334'         MSG$,BEGIN,ERMS4        ;ASCII MESSAGE CALL
007316' 004767  000636                  CALL    UNITS           ;PRINT THE TM78#-TU78#
007322' 004767  001010                  CALL    RECFIL          ;PRINT RECORD NUMBER FILE NUMBER
007326' 004767  000726                  CALL    DROPTU          ;PRINT DROP MESSAGE
007332' 000207                          RTS     PC              ;RETURN
007334' 047125  042522  047503  ERMS4:  .ASCII  'UNRECOVERABLE WRITE '
007342' 042526  040522  046102
007350' 020105  051127  052111
007356' 020105
```

```
 2713  007360' 041507     122          TYPE2:  .ASCII  'GCR'
 2714  007363'    040  051105  047522          .ASCIZ  ' ERROR%'
       007370' 022522     000
 2715          007374'                         .EVEN
 2716
 2717  007374'                         ERR5:
  (1)                                            ;*****************************************************************
  (1)  007374' 104406  000000' 000000            SOFER$,BEGIN,NULL       ;
  (1)                                            ;*****************************************************************
 2718  007402' 104401  000000' 007422'            MSG$,BEGIN,ERMS5        ;ASCII MESSAGE CALL
 2719  007410' 004767  000544                      CALL    UNITS           ;PRINT THE TM78#-TU78#
 2720  007414' 004767  000716                      CALL    RECFIL          ;PRINT RECORD NUMBER FILE NUMBER
 2721  007420' 000207                              RTS     PC              ;RETURN
 2722  007422' 042522  047503  042526  ERMS5:  .ASCII  'RECOVERABLE READ '
       007430' 040522  046102  020105
       007436' 042522  042101     040
 2723  007443'    107  051103          TYPE3:  .ASCII  'GCR'
 2724  007446' 042440  051122  051117          .ASCIZ  ' ERROR%'
       007454' 000045
 2725                                          .EVEN
 2726
 2727  007456'                         ERR6:
  (1)                                            ;*****************************************************************
  (1)  007456' 104405  000000' 000000            HRDER$,BEGIN,NULL       ;
  (1)                                            ;*****************************************************************
 2728  007464' 104401  000000' 007504'            MSG$,BEGIN,ERMS6        ;ASCII MESSAGE CALL
 2729  007472' 004767  000462                      CALL    UNITS           ;PRINT THE TM78#-TU78#
 2730  007476' 004767  000634                      CALL    RECFIL          ;PRINT RECORD NUMBER FILE NUMBER
 2731  007502' 000207                              RTS     PC              ;RETURN
 2732  007504' 047125  042522  047503  ERMS6:  .ASCII  'UNRECOVERABLE READ '
       007512' 042526  040522  046102
       007520' 020105  042522  042101
       007526'    040
 2733  007527'    107  051103          TYPE4:  .ASCII  'GCR'
 2734  007532' 042440  051122  051117          .ASCIZ  ' ERROR%'
       007540' 000045
 2735                                          .EVEN
 2736
 2737  007542'                         ERR7:
  (1)                                            ;*****************************************************************
  (1)  007542' 104405  000000' 000000            HRDER$,BEGIN,NULL       ;
  (1)                                            ;*****************************************************************
 2738  007550' 104401  000000' 007570'            MSG$,BEGIN,ERMS7        ;ASCII MESSAGE CALL
 2739  007556' 004767  000376                      CALL    UNITS           ;PRINT TM78#-TU78#
 2740  007562' 004767  000520                      CALL    DROPTM          ;DROP THE TM78
 2741  007566' 000207                              RTS     PC              ;RETURN
 2742  007570' 046524  043040  052501  ERMS7:  .ASCIZ  'TM FAULT INTERRUPT%'
       007576' 052114  044440  052116
       007604' 051105  052522  052120
       007612' 000045
 2743                                          .EVEN
 2744
 2745  007614'                         ERR10:
  (1)                                            ;*****************************************************************
  (1)  007614' 104406  000000' 000000            SOFER$,BEGIN,NULL       ;
  (1)                                            ;*****************************************************************
```



```
 2746  007622' 017767  001246  000650          MOV     @CS1,TEMP       ;GET THE CONTENTS OF CS1
 2747                                          ;*****************************************************************
  (1)                                                                   ;CONVERT TEMP TO ASCII AND
  (1)                                                                   ;STORE AT ERM10A
  (1)  007630' 104420  000000' 010500'          OTOA$,BEGIN,TEMP,ERM10A
  (1)  007636' 007731'
  (1)                                           ;*****************************************************************
 2748  007640' 017767  001240  000632          MOV     @CS2,TEMP       ;GET THE CONTENTS OF CS2
 2749                                          ;*****************************************************************
  (1)                                                                   ;CONVERT TEMP TO ASCII AND
  (1)                                                                   ;STORE AT ERM10B
  (1)  007646' 104420  000000' 010500'          OTOA$,BEGIN,TEMP,ERM10B
  (1)  007654' 007747'
  (1)                                           ;*****************************************************************
 2750  007656' 104401  000000' 007676'          MSG$,BEGIN,ERMS10       ;ASCII MESSAGE CALL
 2751  007664' 004767  000270                    CALL    UNITS           ;PRINT TM78#-TU78#
 2752  007670' 004767  000442                    CALL    RECFIL          ;PRINT RECORD NUMBER FILE NUMBER
 2753  007674' 000207                            RTS     PC              ;RETURN
 2754  007676' 040515  051523  052502  ERMS10: .ASCII  'MASSBUS STATUS ERROR CS1 = '
       007704' 020123  052123  052101
       007712' 051525  042440  051122
       007720' 051117  041440  030523
       007726' 036440     040
 2755  007731'    060  030060  030060  ERM10A: .ASCII  '000000  CS2 = '
       007736' 020060  041440  031123
       007744' 036440     040
 2756  007747'    060  030060  030060  ERM10B: .ASCIZ  '000000%'
       007754' 022460     000
 2757          007760'                         .EVEN
 2758
 2759  007760'                         ERR11:
  (1)                                            ;*****************************************************************
  (1)  007760' 104405  000000' 000000            HRDER$,BEGIN,NULL       ;
  (1)                                            ;*****************************************************************
 2760  007766' 104401  000000' 007776'            MSG$,BEGIN,ERMS11       ;ASCII MESSAGE CALL
 2761  007774' 000207                              RTS     PC              ;RETURN
 2762  007776' 047516  052040  033525  ERMS11: .ASCIZ  'NO TU78 UNITS AVAILABLE%'
       010004' 020070  047125  052111
       010012' 020123  053101  044501
       010020' 040514  046102  022505
       010026'    000
 2763          010030'                         .EVEN
 2764
 2765  010030'                         ERR12:
  (1)                                            ;*****************************************************************
  (1)  010030' 104405  000000' 000000            HRDER$,BEGIN,NULL       ;
  (1)                                            ;*****************************************************************
 2766  010036' 104401  000000' 010046'            MSG$,BEGIN,ERMS12       ;ASCII MESSAGE CALL
 2767  010044' 000207                              RTS     PC              ;RETURN
 2768  010046' 047502  044124  042040  ERMS12: .ASCIZ  'BOTH DENSITIES DISABLED%'
       010054' 047105  044523  044524
       010062' 051505  042040  051511
       010070' 041101  042514  022504
       010076'    000
 2769          010100'                         .EVEN
 2770  010100'                         ERR13:
```

```
                                        ;*****************************************************************
 (1)                                    HRDER$,BEGIN,NULL       ;
 (1)                                    ;*****************************************************************
2771  010106' 104401  000000' 010126'  MSG$,BEGIN,ERMS13       ;ASCII MESSAGE CALL
2772  010114' 004767  000040           CALL    UNITS           ;PRINT UNITS
2773  010120' 004767  000162           CALL    DROPTM          ;PRINT DROPPING TM78
2774  010124' 000207                   RTS     PC              ;RETURN
2775  010126' 046524  034067  051040  ERMS13: .ASCIZ/TM78 READY BIT TIME OUT%/
      010134' 040505  054504  041040
      010142' 052111  052040  046511
      010150' 020105  052517  022524
      010156'    000
2776          010160'                   .EVEN
2777  010160' 016767  000256  000312  UNITS:  MOV     TM78,TEMP       ;GET THE TM78#
2778                                    ;*****************************************************************
 (1)                                                                  ;CONVERT TEMP TO ASCII AND
 (1)                                                                  ;STORE AT UNITTM
 (1)  010166' 104420  000000' 010500'       OTOA$,BEGIN,TEMP,UNITTM
 (1)  010174' 010231'
 (1)                                    ;*****************************************************************
2779  010176' 016767  000242  000274          MOV     TU78,TEMP       ;GET THE TU78#
2780                                    ;*****************************************************************
 (1)                                                                  ;CONVERT TEMP TO ASCII AND
 (1)                                                                  ;STORE AT UNITTU
 (1)  010204' 104420  000000' 010500'       OTOA$,BEGIN,TEMP,UNITTU
 (1)  010212' 010247'
 (1)                                    ;*****************************************************************
2781  010214' 104401  000000' 010224'       MSG$,BEGIN,UNITMS       ;ASCII MESSAGE CALL
2782  010222' 000207                   RTS     PC              ;RETURN
2783
2784  010224' 046524  034067     040  UNITMS: .ASCII  'TM78 '
2785  010231'    060  030060  030060  UNITTM: .ASCII  '000000   TU78 '
      010236' 020060  020040  052524
      010244' 034067     040
2786  010247'    060  030060  030060  UNITTU: .ASCIZ  '000000 %'
      010254' 020060  000045
2787                                    .EVEN
2788  010260'                          DROPTU:
 (1)  010260' 104401  000000' 010270'       MSG$,BEGIN,MSDTU        ;ASCII MESSAGE CALL
2789  010266' 000207                   RTS     PC
2790
2791  010270' 052524  034067  042040  MSDTU:  .ASCIZ  'TU78 DROPPED%'
      010276' 047522  050120  042105
      010304' 000045
2792                                    .EVEN
2793
2794  010306'                          DROPTM:
 (1)  010306' 104401  000000' 010316'       MSG$,BEGIN,MSDTM        ;ASCII MESSAGE CALL
2795  010314' 000207                   RTS     PC
2796
2797  010316' 046524  034067  042040  MSDTM:  .ASCIZ  'TM78 DROPPED%'
      010324' 047522  050120  042105
      010332' 000045
2798                                    .EVEN
2799
2800  010334' 000045                   CRLF:   .ASCIZ  '%'
```



```
2801                                    .EVEN
2802
2803  010336'                          RECFIL:
 (1)                                    ;*****************************************************************
 (1)                                                                  ;CONVERT RECORD TO ASCII AND
 (1)                                                                  ;STORE AT RECMSG
 (1)  010336' 104421  000000' 010504'       BTOD$,BEGIN,RECORD,RECMSG
 (1)  010344' 010377'
 (1)                                    ;*****************************************************************
2804                                    ;*****************************************************************
 (1)                                                                  ;CONVERT FILE TO ASCII AND
 (1)                                                                  ;STORE AT FILMSG
 (1)  010346' 104421  000000' 010502'       BTOD$,BEGIN,FILE,FILMSG
 (1)  010354' 010417'
 (1)                                    ;*****************************************************************
2805  010356' 104401  000000' 010366'       MSG$,BEGIN,RFMSG        ;ASCII MESSAGE CALL
2806  010364' 000207                   RTS     PC
2807
2808  010366' 042522  047503  042122  RFMSG:  .ASCII  'RECORD#  '
      010374' 020043     040
2809  010377'    060  030060  030060  RECMSG: .ASCII  '00000.   FILE#  '
      010404' 020056  020040  044506
      010412' 042514  020043     040
2810  010417'    060  030060  030060  FILMSG: .ASCIZ  '00000.%'
      010424' 022456     000
2811          010430'                   .EVEN
```

```
                                .SBTTL  PROGRAM VARIABLE
010430' 000000  TMBIT:  .WORD   0       ;TM78 # CONVERTED TO A BIT POSITION
010432' 000000  TMBIT2: .WORD   0       ;TM78 #2
010434' 000000  TUBIT:  .WORD   0       ;TU78 # CONVERTED TO A BIT POSITION
010436' 000000  TUBIT2: .WORD   0       ;TU78 #2
010440' 000000  UCNT:   .WORD   0       ;UNIT COUNT (NUMBER OF TU78'S ACTIVE)
010442' 000000  TM78:   .WORD   0       ;TM78 #
010444' 000000  TU78:   .WORD   0       ;TU78 #
010446' 000000  EOTCNT: .WORD   0       ;EOT COUNT (NUMBER OF TU78'S AT EOT)
010450' 000000  TINTCD: .WORD   0       ;TERMINATION INTERRUPT CODE
010452' 000000  ITU78:  .WORD   0       ;INTERRUPTING TU78 #
010454' 000000  BOTCNT: .WORD   0       ;BOT COUNT (NUMBER OF TU78'S AT BOT)
010456' 000000  SENSE:  .WORD   0       ;SENSE COMMAND INFO
010460' 000000  ERRINX: .WORD   0       ;ERROR INDEX
010462' 000000  A16A17: .WORD   0       ;EXTENDED ADDRESS BITS 16 AND 17
010464' 000000  PA18:   .WORD   0       ;LOWER 16 BITS OF 18 BIT MEMORY ADDRESS
010466' 000000  XMEM:   .WORD   0       ;EXTENDED ADDRESS BITS OF AN 18 BIT ADDRESS
010470' 000000  PA22:   .WORD   0       ;LOWER 16 BITS OF 22 BIT MEMORY ADDRESS
010472' 000000  EA22:   .WORD   0       ;EXTENDED ADDRESS BITS OF A 22 BIT ADDRESS
010474' 000000  FLAGS:  .WORD   0       ;FLAGS
010476' 000000  TMUNIT: .WORD   0       ;TM78 UNITS
010500' 000000  TEMP:   .WORD   0       ;TEMP STORAGE
010502' 000000  FILE:   .WORD   0       ;FILE NUMBER
010504' 000000  RECORD: .WORD   0       ;RECORD NUMBER
010506' 000000  RDFCN:  .WORD   0       ;READ FUNCTION CODE
010510' 000000  MONINR: .WORD   0       ;COPY OF MOTION INTERRUPT REGISTER
010512' 000000  ASREG:  .WORD   0       ;COPY OF ATTENTION SUMMARY REGISTER
```



```
                                .SBTTL  UNIT TABLE

010514' 000000  UTBL:   .WORD   0       ;TM78 = 0   TU78 0 - 3
010516' 000000          .WORD   0       ;TM78 = 1   TU78 0 - 3
010520' 000000          .WORD   0       ;TM78 = 2   TU78 0 - 3
010522' 000000          .WORD   0       ;TM78 = 3   TU78 0 - 3
010524' 000000          .WORD   0       ;TM78 = 4   TU78 0 - 3
010526' 000000          .WORD   0       ;TM78 = 5   TU78 0 - 3
010530' 000000          .WORD   0       ;TM78 = 6   TU78 0 - 3
010532' 000000          .WORD   0       ;TM78 = 7   TU78 0 - 3

                                .SBTTL  EOT TABLE

010534' 000000  EOTTBL: .WORD   0       ;TM78 = 0   TU78 0 - 3
010536' 000000          .WORD   0       ;TM78 = 1   TU78 0 - 3
010540' 000000          .WORD   0       ;TM78 = 2   TU78 0 - 3
010542' 000000          .WORD   0       ;TM78 = 3   TU78 0 - 3
010544' 000000          .WORD   0       ;TM78 = 4   TU78 0 - 3
010546' 000000          .WORD   0       ;TM78 = 5   TU78 0 - 3
010550' 000000          .WORD   0       ;TM78 = 6   TU78 0 - 3
010552' 000000          .WORD   0       ;TM78 = 7   TU78 0 - 3

                                .SBTTL  SCRATCH UNIT TABLE

010554' 000000  SUTBL:  .WORD   0       ;TM78 = 0   TU78 0 - 3
010556' 000000          .WORD   0       ;TM78 = 1   TU78 0 - 3
010560' 000000          .WORD   0       ;TM78 = 2   TU78 0 - 3
010562' 000000          .WORD   0       ;TM78 = 3   TU78 0 - 3
010564' 000000          .WORD   0       ;TM78 = 4   TU78 0 - 3
010566' 000000          .WORD   0       ;TM78 = 5   TU78 0 - 3
010570' 000000          .WORD   0       ;TM78 = 6   TU78 0 - 3
010572' 000000          .WORD   0       ;TM78 = 7   TU78 0 - 3
```

```
                                        .SBTTL  UNRECOVERABLE READ STATISTIC TABLE

010574' 000000                  URREAD: .WORD   0               ;TM78 0   TU78 0
010576' 000000                          .WORD   0               ;TM78 0   TU78 1
010600' 000000                          .WORD   0               ;TM78 0   TU78 2
010602' 000000                          .WORD   0               ;TM78 0   TU78 3
010604' 000000                          .WORD   0               ;TM78 1   TU78 0
010606' 000000                          .WORD   0               ;TM78 1   TU78 1
010610' 000000                          .WORD   0               ;TM78 1   TU78 2
010612' 000000                          .WORD   0               ;TM78 1   TU78 3
010614' 000000                          .WORD   0               ;TM78 2   TU78 0
010616' 000000                          .WORD   0               ;TM78 2   TU78 1
010620' 000000                          .WORD   0               ;TM78 2   TU78 2
010622' 000000                          .WORD   0               ;TM78 2   TU78 3
010624' 000000                          .WORD   0               ;TM78 3   TU78 0
010626' 000000                          .WORD   0               ;TM78 3   TU78 1
010630' 000000                          .WORD   0               ;TM78 3   TU78 2
010632' 000000                          .WORD   0               ;TM78 3   TU78 3
010634' 000000                          .WORD   0               ;TM78 4   TU78 0
010636' 000000                          .WORD   0               ;TM78 4   TU78 1
010640' 000000                          .WORD   0               ;TM78 4   TU78 2
010642' 000000                          .WORD   0               ;TM78 4   TU78 3
010644' 000000                          .WORD   0               ;TM78 5   TU78 0
010646' 000000                          .WORD   0               ;TM78 5   TU78 1
010650' 000000                          .WORD   0               ;TM78 5   TU78 2
010652' 000000                          .WORD   0               ;TM78 5   TU78 3
010654' 000000                          .WORD   0               ;TM78 6   TU78 0
010656' 000000                          .WORD   0               ;TM78 6   TU78 1
010660' 000000                          .WORD   0               ;TM78 6   TU78 2
010662' 000000                          .WORD   0               ;TM78 6   TU78 3
010664' 000000                          .WORD   0               ;TM78 7   TU78 0
010666' 000000                          .WORD   0               ;TM78 7   TU78 1
010670' 000000                          .WORD   0               ;TM78 7   TU78 2
010672' 000000                          .WORD   0               ;TM78 7   TU78 3

                                        .SBTTL  RECOVERABLE READ STATISTIC TABLE

010674' 000000                  RREAD:  .WORD   0               ;TM78 0   TU78 0
010676' 000000                          .WORD   0               ;TM78 0   TU78 1
010700' 000000                          .WORD   0               ;TM78 0   TU78 2
010702' 000000                          .WORD   0               ;TM78 0   TU78 3
010704' 000000                          .WORD   0               ;TM78 1   TU78 0
010706' 000000                          .WORD   0               ;TM78 1   TU78 1
010710' 000000                          .WORD   0               ;TM78 1   TU78 2
010712' 000000                          .WORD   0               ;TM78 1   TU78 3
010714' 000000                          .WORD   0               ;TM78 2   TU78 0
010716' 000000                          .WORD   0               ;TM78 2   TU78 1
010720' 000000                          .WORD   0               ;TM78 2   TU78 2
010722' 000000                          .WORD   0               ;TM78 2   TU78 3
010724' 000000                          .WORD   0               ;TM78 3   TU78 0
010726' 000000                          .WORD   0               ;TM78 3   TU78 1
010730' 000000                          .WORD   0               ;TM78 3   TU78 2
010732' 000000                          .WORD   0               ;TM78 3   TU78 3
010734' 000000                          .WORD   0               ;TM78 4   TU78 0
010736' 000000                          .WORD   0               ;TM78 4   TU78 1
010740' 000000                          .WORD   0               ;TM78 4   TU78 2
```



```
010742' 000000                          .WORD   0               ;TM78 4   TU78 3
010744' 000000                          .WORD   0               ;TM78 5   TU78 0
010746' 000000                          .WORD   0               ;TM78 5   TU78 1
010750' 000000                          .WORD   0               ;TM78 5   TU78 2
010752' 000000                          .WORD   0               ;TM78 5   TU78 3
010754' 000000                          .WORD   0               ;TM78 6   TU78 0
010756' 000000                          .WORD   0               ;TM78 6   TU78 1
010760' 000000                          .WORD   0               ;TM78 6   TU78 2
010762' 000000                          .WORD   0               ;TM78 6   TU78 3
010764' 000000                          .WORD   0               ;TM78 7   TU78 0
010766' 000000                          .WORD   0               ;TM78 7   TU78 1
010770' 000000                          .WORD   0               ;TM78 7   TU78 2
010772' 000000                          .WORD   0               ;TM78 7   TU78 3

                                        .SBTTL  RECOVERABLE WRITE STATISTIC TABLE

010774' 000000                  RWRIT:  .WORD   0               ;TM78 0   TU78 0
010776' 000000                          .WORD   0               ;TM78 0   TU78 1
011000' 000000                          .WORD   0               ;TM78 0   TU78 2
011002' 000000                          .WORD   0               ;TM78 0   TU78 3
011004' 000000                          .WORD   0               ;TM78 1   TU78 0
011006' 000000                          .WORD   0               ;TM78 1   TU78 1
011010' 000000                          .WORD   0               ;TM78 1   TU78 2
011012' 000000                          .WORD   0               ;TM78 1   TU78 3
011014' 000000                          .WORD   0               ;TM78 2   TU78 0
011016' 000000                          .WORD   0               ;TM78 2   TU78 1
011020' 000000                          .WORD   0               ;TM78 2   TU78 2
011022' 000000                          .WORD   0               ;TM78 2   TU78 3
011024' 000000                          .WORD   0               ;TM78 3   TU78 0
011026' 000000                          .WORD   0               ;TM78 3   TU78 1
011030' 000000                          .WORD   0               ;TM78 3   TU78 2
011032' 000000                          .WORD   0               ;TM78 3   TU78 3
011034' 000000                          .WORD   0               ;TM78 4   TU78 0
011036' 000000                          .WORD   0               ;TM78 4   TU78 1
011040' 000000                          .WORD   0               ;TM78 4   TU78 2
011042' 000000                          .WORD   0               ;TM78 4   TU78 3
011044' 000000                          .WORD   0               ;TM78 5   TU78 0
011046' 000000                          .WORD   0               ;TM78 5   TU78 1
011050' 000000                          .WORD   0               ;TM78 5   TU78 2
011052' 000000                          .WORD   0               ;TM78 5   TU78 3
011054' 000000                          .WORD   0               ;TM78 6   TU78 0
011056' 000000                          .WORD   0               ;TM78 6   TU78 1
011060' 000000                          .WORD   0               ;TM78 6   TU78 2
011062' 000000                          .WORD   0               ;TM78 6   TU78 3
011064' 000000                          .WORD   0               ;TM78 7   TU78 0
011066' 000000                          .WORD   0               ;TM78 7   TU78 1
011070' 000000                          .WORD   0               ;TM78 7   TU78 2
011072' 000000                          .WORD   0               ;TM78 7   TU78 3

011074'                         RHTBL:                          ;MASSBUS REGISTER TABLE
011074' 000000                  CS1:    .WORD   0               ;REGISTER +00
011076' 000000                  WC:     .WORD   0               ;REGISTER +02
011100' 000000                  BA:     .WORD   0               ;REGISTER +04
011102' 000000                  BC:     .WORD   0               ;REGISTER +06
011104' 000000                  CS2:    .WORD   0               ;REGISTER +10
011106' 000000                  XFRINT: .WORD   0               ;REGISTER +12
```

```
 2986  011110' 000000                TC:     .WORD   0               ;REGISTER +14
 2987  011112' 000000                AS:     .WORD   0               ;REGISTER +16
 2988  011114' 000000                TUSTAT: .WORD   0               ;REGISTER +20
 2989  011116' 000000                DB:     .WORD   0               ;REGISTER +22
 2990  011120' 000000                DI1:    .WORD   0               ;REGISTER +24
 2991  011122' 000000                DT:     .WORD   0               ;REGISTER +26
 2992  011124' 000000                SN:     .WORD   0               ;REGISTER +30
 2993  011126' 000000                DI2:    .WORD   0               ;REGISTER +32
 2994  011130' 000000                DI3:    .WORD   0               ;REGISTER +34
 2995  011132' 000000                MOINT:  .WORD   0               ;REGISTER +36
 2996  011134' 000000                MO0:    .WORD   0               ;REGISTER +40
 2997  011136' 000000                MO1:    .WORD   0               ;REGISTER +42
 2998  011140' 000000                MO2:    .WORD   0               ;REGISTER +44
 2999  011142' 000000                MO3:    .WORD   0               ;REGISTER +46
 3000  011144' 000000                AD78:   .WORD   0               ;REGISTER +50
 3001  011146' 000000                ST78:   .WORD   0               ;REGISTER +52
 3002  011150' 000000                        .WORD   0               ;REGISTER +54
 3003  011152' 000000                        .WORD   0               ;REGISTER +56
 3004  011154' 000000                        .WORD   0               ;REGISTER +60
 3005  011156' 000000                        .WORD   0               ;REGISTER +62
 3006  011160' 000000                        .WORD   0               ;REGISTER +64
 3007  011162' 000000                        .WORD   0               ;REGISTER +66
 3008  011164' 000000                        .WORD   0               ;REGISTER +70
 3009  011166' 000000                        .WORD   0               ;REGISTER +72
 3010  011170' 000000                EBA:    .WORD   0               ;REGISTER +74
 3011  011172' 000000                CS3:    .WORD   0               ;REGISTER +76
 3012
 3013  011174' 002000                BUFIN:  .BLKB   1024.           ;1024 BYTE READ BUFFER
```



```
 3015                                   .SBTTL  PROGRAM DEFINITIONS
 3016                                   ;
 3017                                   ;INTERRUPT CODE DEFINITIONS
 3018                                   ;
 3019
 3020           000027                  BTAPE   =       000027          ;BAD TAPE
 3021           000001                  DONE    =       000001          ;DONE
 3022           000004                  EOT     =       000004          ;END OF TAPE
 3023           000023                  RDOPP   =       000023          ;READ OPPOSITE
 3024           000022                  RTRY    =       000022          ;RETRY
 3025           000032                  TMFB    =       000032          ;TM FAULT B
 3026           000024                  UNREAD  =       000024          ;UNREADABLE
 3027
 3028                                   ;
 3029                                   ;COMMAND CODE DEFINITIONS
 3030                                   ;
 3031
 3032           000063                  WGCR    =       000063          ;WRITE GCR
 3033           000061                  WPE     =       000061          ;WRITE PE
 3034           000017                  GCRTM   =       000017          ;WRITE GCR TAPE MARK
 3035           000015                  PETM    =       000015          ;WRITE PE TAPE MARK
 3036           000007                  REW     =       000007          ;REWIND
 3037           000011                  SEN     =       000011          ;SENSE
 3038           000071                  RFWD    =       000071          ;READ FORWARD
 3039           000077                  RREV    =       000077          ;READ REVERSE
```

```
                        .REM    -
                        14.0    ERROR DEFINITIONS FOR TM78 DEC/X11 DRIVER
                                ----------------------------------------

                                ERROR 1 HRDER
                                NO TM78 UNITS AVAILABLE

                                ERROR 2 HRDER
                                UNEXPECTED INTERRUPT CODE XXXXXX
                                TM78 XXXXXX   TU78 XXXXXX
                                TU78 DROPPED

                                ERROR 3 SOFERR
                                RECOVERABLE WRITE GCR/PE ERROR
                                TM78 XXXXXX   TU78 XXXXXX
                                RECORD # XXXXX.   FILE # XXXXX.

                                ERROR 4 HRDER
                                UNRECOVERABLE WRITE GCR/PE ERROR
                                TM78 XXXXXX   TU78 XXXXXX
                                RECORD # XXXXX.   FILE # XXXXX.
                                TU78 DROPPED.

                                ERROR 5 SOFERR
                                RECOVERABLE READ/GCR PE ERROR
                                TM78 XXXXXX   TU78 XXXXXX
                                RECORD # XXXXX.   FILE # XXXXX.

                                ERROR 6 HRDER
                                UNRECOVERABLE READ GCR/PE ERROR
                                TM78 XXXXXX   TU78 XXXXXX
                                RECORD # XXXXX.   FILE # XXXXX.

                                ERROR 7 HRDER
                                TM FAULT INTERRUPT
                                TM78 XXXXXX   TU78 XXXXXX
                                TM78 DROPPED

                                ERROR 10 SOFERR
                                MASSBUS STATUS ERROR CS1=XXXXXX CS2=XXXXXX
                                TM78 XXXXXX   TU78 XXXXXX
                                RECORD # XXXXX.   FILE # XXXXX.

                                ERROR 11 HRDER
                                NO TU78 UNITS AVAILABLE

                                ERROR 12 HRDER
                                BOTH DENSITIES DISABLED

                                DATA COMPARE ERROR

                        -
          000001        .END
```



```
ACSR     000102R        1155#  2588#
ADDR     000006R        1155#  1232
ADDR22=  001000         1155#  2386   2449
AD78     011144R        3000#
AS       011112R        2592*  2601   2604*  2627*  2987#
ASB      000106R        1155#
ASREG    010512R        2601*  2602   2604   2839#
ASTAT    000104R        1155#  2589*
AWAS     000110R        1155#
A16A17   010462R        2393*  2394*  2395*  2398*  2399*  2400*  2401*  2402*  2409   2456*  2457*  2458*  2461*
                        2462*  2463*  2464*  2465*  2469   2827#
BA       011100R        2391*  2397*  2454*  2460*  2982#
BC       011102R        2382*  2383*  2444*  2445*  2983#
BEGIN    000000R        1155#  1199   1206   1215   1307   1596   1639   1677   2221   2263   2276   2278   2280
                        2281   2282   2283   2390   2411   2448   2453   2471   2530   2554   2578   2599   2616
                        2664   2680   2681   2686   2687   2688   2696   2697   2706   2707   2717   2718   2727
                        2728   2737   2738   2745   2747   2749   2750   2759   2760   2765   2766   2770   2771
                        2778   2780   2781   2788   2794   2803   2804   2805
BIT0  = 000001          1155#  1480   1483   1694   1698   1702   2217   2404   2524   2639
BIT1  = 000002          1155#  1281   1383   1486   1604   1641   1658   1661   1667   1673
BIT10 = 002000          1155#  1752
BIT11 = 004000          1155#
BIT12 = 010000          1155#  1291
BIT13 = 020000          1155#
BIT14 = 040000          1155#  1289   1755   1837   1941   2039   2129   2182   2378   2440   2518   2546   2570
BIT15 = 100000          1155#
BIT2  = 000004          1155#  1822   1829   1848   1867   1925   1933   1948   1954   2024   2031   2046   2052
                        2637
BIT3  = 000010          1155#  2261   2403   2466   2520   2548   2572   2596   2618   2628
BIT4  = 000020          1155#  1209   1210   1481   1696   1823   1828   1847   1858   1875   1878   1884   1926
                        1932   1960   1970   1977   1983   2025   2030   2058   2068   2075   2081
BIT5  = 000040          1155#  1209   1210   1249   1593   1700   2403   2466   2521   2549   2573   2590   2598
                        2615   2629
BIT6  = 000100          1155#  1594   2406   2408   2468   2529   2553   2577   2586   2593   2630
BIT7  = 000200          1155#
BIT8  = 000400          1155#
BIT9  = 001000          1155#
BOTCNT   010454R        1647   1740*  1754*  2824#
BREAKS=  104407         1155#  1307
BR1      000012R        1155#  1381
BR2      000013R        1155#
BTAPE =  000027         1856   3020#
BTODS =  104421         1155#  2276   2278   2280   2803   2804
BUFIN    011174R        1155   2437   2495   3013#
CDATAS=  104412         1155#  2221
CDERCT   000144R        1155#
CDWDCT   000146R        1155#
CONFIG   000056R        1155#
CRLF     010334R        2664   2800#
CSRA     000100R        1155#  2587#
CS1      011074R        1196*  1203*  1212*  1289*  1290*  1594*  1638*  1755   1837   1941   2039   2129   2182
                        2378*  2379*  2410*  2440*  2441*  2470*  2518*  2519*  2529*  2546*  2547*  2553*  2570*
                        2571*  2577*  2586*  2587   2588   2593*  2630*  2746   2980#
CS2      011104R        1249*  1287*  1291   1395*  1428*  2377*  2439*  2517*  2545*  2569*  2589   2607*  2620*
                        2748   2984#
CS3      011172R        3011#
```

```
DATCKS= 104411          1155#
DATERS= 104404          1155#
DB      011116R         2989#
DI1     011120R         2990#
DI2     011126R         2993#
DI3     011130R         2994#
DONE  = 000001          1402    1432    1750    1826    1930    2028    2127    2180    3021#
DROPTM  010306R         2740*   2773*   2794#
DROPTU  010260R         2690*   2710*   2788#
DT      011122R         1295    2991#
DVID1   000014R         1155#   1283
EA22    010472R         2392    2393    2455    2456    2831#
EHA     011170R         2392*   2455*   3010#
ENDITS= 104413          1155#   1677
ENDS  = 104410          1155#   1199    1206    1215    1639
EOT   = 000004          1843    3022#
EOTCNT  010446R         1650*   1659    2333*   2344*   2821#
EOTTBL  010534R         1476*   1609    1651*   1846*   2123    2342    2854#
ERMS1   007062R         2681    2683#
ERMS10  007676R         2750    2754#
ERMS11  007776R         2760    2762#
ERMS12  010046R         2766    2768#
ERMS13  010126R         2771    2775#
ERMS2   007152R         2688    2692#
ERMS2A  007205R         2687    2693#
ERMS3   007244R         2697    2701#
ERMS4   007334R         2707    2712#
ERMS5   007422R         2718    2722#
ERMS6   007504R         2728    2732#
ERMS7   007570R         2738    2742#
ERM10A  007731R         2747    2755#
ERM10B  007747R         2749    2756#
ERRINX  010460R         1197*   1204*   1213*   1310*   1636*   1758*   1763*   1768*   1834*   1839*   1852*   1859*   1871*
                        1879*   1938*   1943*   1964*   1971*   1978*   2036*   2041*   2062*   2069*   2076*   2131*   2137*
                        2143*   2185*   2193*   2198*   2661    2826#
ERRTYP  000106R         1155#   2660*
ERR1    007044R         2668    2680#
ERR10   007614R         2675    2745#
ERR11   007760R         2676    2759#
ERR12   010030R         2677    2765#
ERR13   010100R         2678    2770#
ERR2    007114R         2669    2686#
ERR3    007216R         2670    2696#
ERR4    007302R         2671    2706#
ERR5    007374R         2672    2717#
ERR6    007456R         2673    2727#
ERR7    007542R         2674    2737#
EXITS = 104400          1155#   2411    2471    2530    2554    2578
FILE    010502R         1485*   1655*   1670*   2804    2835#
FILMSG  010417R         2804    2810#
FLAGS   010474R         1190*   1480*   1483*   1486*   1593*   1604    1641    1658*   1661*   1667    1673    1694    1698*
                        1702*   1822*   1823*   1828*   1829    1847*   1848    1858*   1867*   1875*   1878*   1884    1925*
                        1926*   1932*   1933    1948*   1954*   1960*   1970*   1977*   1983    2024*   2025*   2030*   2031
                        2046*   2052*   2058*   2068*   2075*   2081    2403*   2404    2466*   2520*   2521*   2524    2548*
                        2549*   2572*   2573*   2590    2596    2598*   2615*   2618    2628*   2629*   2639    2832#
FREE    000150R         1155#
```



```
GCRTM = 000017          2526    3034#
GETPAS= 104415          1155#   2448
GWBUFS= 104414          1155#   1596
HRDCNT  000044R         1155#
HRDERS= 104405          1155#   2680    2686    2706    2727    2737    2759    2765    2770
HRDPAS  000050R         1155#
ICONT   000036R         1155#
ICOUNT  000040R         1155#
IDNUM   000122R         1155#
IMODX.= 000000          1155#   1596
INIT    000030R         1155#
INTR    000120R         1155#
ITU78   010452R         2610*   2611*   2612*   2613    2823#
MAP22$= 104416          1155#   2390    2453
MODNAM  000000R         1155#
MODSP   000252R         1155#
MOINT   011132R         2608    2610    2995#
MONINR  010510R         2608*   2625    2838#
MO0     011134R         2526*   2528*   2552*   2576*   2996#
MO1     011136R         2997#
MO2     011140R         2998#
MO3     011142R         2999#
MRDCUM  005542R         2439*   2499
MSDTM   010316R         2794    2797#
MSDTU   010270R         2788    2791#
MSGN$ = 104403          1155#
MSGS$ = 104402          1155#
MSGS  = 104401          1155#   2263    2283    2664    2681    2688    2697    2707    2718    2728    2738    2750    2760
                        2766    2771    2781    2788    2794    2805
M1.1    000426R         1191*   1232#
M1.2    000462R         1192*   1249#
M1.3    000472R         1193*   1279#
M1.4    000714R         1200*   1379#
M1.40T  001342R         1419    1449#
M1.4X   001152R         1384    1420#
M1.5    001344R         1207*   1468#
M2.0    001570R         1216    1593#   1675
M2.1    002272R         1663*   1694#
M2.10   005142R         1633*   1643*   2332#
M2.2    002342R         1644*   1740#
M2.3    002574R         1611*   1822#
M2.4    003176R         1620*   1925#
M2.5    003546R         1615*   2024#
M2.6    004112R         1669*   2116#
M2.7    004334R         1606*   2179#
M2.8    004466R         1619*   1624*   2217#
M2.9    004520R         1662*   2261#
M3.1    005260R         1824*   2377#
M3.2    005526R         1927*   1949*   2053*   2436#
M3.3    006004R         1955*   2026*   2047*   2494#
M3.4    006046R         2125*   2517#
M3.5    006150R         2179*   2545#
M3.6    006232R         1401*   1431*   1748*   2569#
M4.0    006314R         1380    2586#
M4.0C   006332R         2599    2616    2618#
M4.1    006634R         1198*   1205*   1214*   1311*   1637*   1759*   1764*   1769*   1835*   1840*   1853*   1860*   1872*
```

```
                    1880*  1939*  1944*  1965*  1972*  1979*  2037*  2042*  2063*  2070*  2077*  2132*  2138*
                    2144*  2186*  2194*  2199*  2637#
M4.1TB  007014R     2663*  2667#
NULL  = 000000      1155#  2680   2686   2696   2706   2717   2727   2737   2745   2759   2765   2770
OPEN  = 000000      1155#
OTOA$ = 104420      1155#  2281   2282   2687   2747   2749   2778   2780
PASCNT  000034R     1155#
PA18    010464R     2388*  2390   2451*  2453   2828#
PA22    010470R     2391   2454   2830#
PETM  = 000015      2528   3035#
PIRQ$ = 000004      1155#  2599   2616
POPSP = 005726      1155#
POPSP2= 022626      1155#
PRTY  = 000000      1155#
PRTY0 = 000000      1155#
PRTY1 = 000040      1155#
PRTY2 = 000100      1155#
PRTY3 = 000140      1155#
PRTY4 = 000200      1155#
PRTY5 = 000240      1155#
PRTY6 = 000300      1155#
PRTY7 = 000340      1155#
PS    = 177776      1155#
PSW   = 177776      1155#
PUSH  = 005746      1155#
PUSH2 = 024646      1155#
RAND$ = 104417      1155#
RANNUM  000054R     1155#
RBUFEA  000130R     1155#  2452   2461
RBUFPA  000126R     1155#  2221   2451   2460
RBUFSZ  000132R     1155#
RBUFVA  000124R     1155#  2437*  2448   2495*  2496*  2497*  2498*
RDFCN   010506R     2219   2436*  2467   2494*  2837#
RDOPP = 000023      1952   2050   3023#
RECFIL  010336R     2699*  2709*  2720*  2730*  2752*  2803#
RECMSG  010377R     2803   2809#
RECORD  010504R     1484*  1597*  1656*  1665   1671   1676*  2803   2836#
RESTRT  001566R     1155   1592#
RES1    000056R     1155#  2386   2449
RES2    000060R     1155#
REW   = 000007      2189   2552   3036#
REWMSG  010366R     2805   2808#
RFWD  = 000071      2436   3038#
RHTBL   011074R     1235*  2979#
RREAD   010674R     1470*  1937*  2035*  2275   2911#
RREV  = 000077      2219   2494   3039#
RSTRT   000112R     1155#
RTRY  = 000022      1865   1946   2044   3024#
RWRIT   010774R     1469*  1833*  1851*  2277   2946#
SBADR   000102R     1155#
SEN   = 000011      2576   3037#
SENSE   010456R     1404*  1405   1434*  1435   1752   2609*  2825#
SN      011124R     2992#
SOFCNT  000042R     1155#
SOFERS= 104406      1155#  2696   2717   2745
SOFPAS  000046R     1155#
```



```
SPOINT  000032R     1155#
SPSIZ = 000040         7#  1155
SR1     000016R     1155#  1208   1281   1383   1481   1696   1700   2217   2261   2637
SR2     000020R     1155#
SR3     000022R     1155#
SR4     000024R     1155#
START   000252R     1155   1190#
STAT    000026R     1155#
STATHD  004740R     2263   2291#
STATLN  005044R     2283   2294#
STHR    005072R     2276   2299#
STRW    005131R     2278   2303#
STTM    005046R     2281   2295#
STTU    005057R     2282   2297#
STURR   005111R     2280   2301#
ST78    011146R     1305   3001#
SUTBL   010554R     1386*  1399   1410*  1424*  1440*  1602   1613   1617   1622   1746   1766*  1771*  1862*
                    1874*  1882*  1974*  1981*  2072*  2079*  2121   2140*  2146*  2196*  2201*  2339   2865#
SVR0    000062R     1155#
SVR1    000064R     1155#
SVR2    000066R     1155#
SVR3    000070R     1155#
SVR4    000072R     1155#
SVR5    000074R     1155#
SVR6    000076R     1155#
SYSCNT  000052R     1155#
TC      011110R     2385*  2447*  2986#
TEMP    010500R     2275*  2276   2277*  2278   2279*  2280   2746*  2747   2748*  2749   2777*  2778   2779*
                    2780   2834#
TINTCD  010450R     1402   1432   1750   1761   1826   1843   1856   1865   1869   1930   1946   1952   1958
                    1968   2028   2044   2050   2056   2066   2127   2135   2180   2189   2191   2621*  2622*
                    2625*  2626*  2687   2822#
TMBIT   010430R     1285   1293   1299   1312   1393   1426   1496*  2602   2627   2814#
TMBIT2  010432R     1398   1409   1423   1439   1499*  1601   1608   1612   1616   1621   1745   1749   1765
                    1770   1825   1831   1845   1861   1873   1881   1928   1935   1961   1973   1980   2033
                    2059   2071   2078   2120   2126   2139   2145   2195   2200   2268   2271   2338   2815#
TMCONV  001462R     1284*  1392*  1422*  1489*  1599*  1743*  2118*  2266*  2336*
TMFB  = 000032      1761   1869   1968   2066   2135   2191   3025#
TMUNIT  010476R     1194   1280*  1283*  1285   1293*  1299*  1312*  1393   1426   2833#
TM78    010442R     1279*  1287   1300*  1301   1390*  1395   1416*  1417   1420*  1428   1446*  1447   1489
                    1497   1595*  1629*  1630   1741*  1776*  1777   2116*  2152*  2153   2264*  2281   2287*
                    2288   2334*  2348*  2349   2377   2439   2517   2545   2569   2607   2620   2777   2819#
TRPDFD= 000022      1155#
TUBIT   010434R     1399   1410   1440   1441   1509*  1602   1609   1613   1617   1622   1746   1771   1846
                    1862   1882   1981   2079   2121   2123   2146   2201   2269   2339   2342   2816#
TUBIT2  010436R     1512*  1832   1850   1936   1962   2034   2060   2274   2817#
TUCONV  001524R     1397*  1430*  1502#  1600*  1744*  2119*  2267*  2337*
TUSTAT  011114R     2609   2988#
TU78    010444R     1396*  1413*  1414   1429*  1443*  1444   1502   1510   1598*  1625*  1626   1742*  1772*
                    1773   2117*  2148*  2149   2265*  2282   2284*  2285   2335*  2345*  2346   2385   2447
                    2522   2550   2574   2613   2779   2820#
TYPE1   007266R     2648*  2652*  2656*  2702#
TYPE2   007360R     2649*  2653*  2657*  2713#
TYPE3   007443R     2650*  2654*  2658*  2723#
TYPE4   007527R     2651*  2655*  2659*  2733#
UCNT    010440R     1201   1391*  1412*  1421*  1442*  1634   1645   1647   1659   2332*  2341*  2818#
```

```
UNITMS  010224R          2781    2784#
UNITS   010160R          2689*   2698#   2708*   2719*   2729*   2739*   2751*   2772*   2777#
UNITTM  010231R          2778    2785#
UNITTU  010247R          2780    2786#
UNREAD= 000024           1958    2056    3026#
URREAD  010574R          1471*   1963#   2061*   2279    2876#
UTBL    010514R          1386    1425*   1441*   2269    2843#
VECTOR  000010R          1155#   1379
WASADR  000104R          1155#
WBUFEA  000136R          1155#   2389    2398
WBUFPA  000134R          1155#   2388    2397
WBUFRQ  000140R          1155#
WBUFSZ  000142R          1155#   2380    2382    2442    2444    2496    2497
WC      011076R          2380*   2381*   2442*   2443*   2981#
WDER    000116R          1155#
WDTO    000114R          1155#
WGCR  = 000063           2406    3032#
WPE   = 000061           2408    3033#
XFLAG   000005R          1155#
XPRINT  011106R          2384#   2446*   2621    2985#
XMEM    010466R          2389*   2452*   2829#
.     = 013174R          2221    2293#   2305#   2684#   2694#   2704#   2715#   2757#   2763#   2769#   2776#   2811#   3013#
```



```
BKMOD     95#
BREAK    185#  1307
BTOD     204#  2276    2278    2280    2803    2804
CKDATA   240#  2221
DATACK   249#
DATERR   138#
DFSEVN   272#  1155
DSEVNT   282#  1155
END      175#
ENDIT    166#  1677
ENDMOD   171#  1199    1206    1215    1639
EQUATS   288#  1155
EXIT     120#  2411    2471    2530    2554    2578
GETPA    231#  2448
GWBUFF   219#  1596
HRDER    128#  2680    2686    2706    2727    2737    2759    2765    2770
IOMOD     91#
IOMODP   115#
IOMODR   111#
IOMODX   107#  1155
MAP22    235#  2390    2453
MODULE     8#  1155
MSG      154#  2263    2283    2664    2681    2688    2697    2707    2718    2728    2738    2750    2760    2766    2771
        2781   2788    2794    2805
MSGN     158#
MSGS     162#
NBKMOD   103#
OTOA     190#  2281    2282    2687    2747    2749    2778    2780
PIRQ     179#  2599    2616
RAND     124#
SBKMOD    99#
SOFER    144#  2696    2717    2745

. ABS.  000000     000
        013174     001

ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

XTMDA0,XTMDA0/CRF=DDXCOM.P11,XTMDA0.P11
RUN-TIME: 4 8 .7 SECONDS
RUN-TIME RATIO: 47/14=3.4
CORE USED:  10K  (20 PAGES)
```

C

Compon...t Part Number : AF-S210A-M1

DIGITAL EQ... ENT CORPORATION
DIAGNOSTIC ENG...NEERING PATCH ORDER

Form : B

Component Part Description : ECO TALLY 01 | ECO Purpose : Patch | Approved : Yes

-----PRODUCT IDENTIFICATION-----

Diagnostic Package Code : CXTMD-A1 | Patch Level : 1 | Product Date : 23-Sep-80

Description : CXTMDA1 DEC/X11 TM78 MODULE | 1ST Copyright year : 1980 | Last Copyright Year : 1980

Expanded Description :

-----PRODUCT COMPONENTS AFFECTED-----

-----PRODUCTS OBSOLETED-----

-----PROBLEM/DESCRIPTION-----

UNEXPECTED INTERRUPT CODE 00000 WHEN RUNNING MONITOR
C ON AN 11/40.

-----SOLUTION/CORRECTION-----

CORRECT TYPE ERROR IN WRITE SUBROUTINE MODULE 3.1.
MOVE BIT 14, @ CS1 SHOULD BE MOV #BIT 14, @ CS1.
THIS IS A CONDITIONAL PATCH.

-----DEPO PATCH AREA-----

Loc	From	To	Mnemonic	Loc	From	To	Mnemonic
5266	16777	12777					

-----PRODUCT CHARACTERISTICS-----

Run Time (seconds) 1st Pass : 75 | 2nd Pass : 75 | ACT Sequence Number : 1351 | Multimedia Affected : *No

Processors : 05 10 20 34 35 40 44 45 50 55 60 70

Operational Environment : 02 03 04 06 50

Device Affected : TM78 TU78

Problem Reports Affected :

Kit Number : ZJ129

-----AUTHORIZATION-----

Author	: G. COOKE	Maintaining Group	: IO DIA
Submitting Engineer	: B. VANDETTE	Maintainer	: DEKNIS
Product Engineer	:	Field Service Engineer	:
Manufacturing Engineer	: R. SACINO	Release Engineer	: M. HANSEN
Charge Number	: V980-05346	Coordination Number	: MC3796

.REM _

IDENTIFICATION

PRODUCT CODE: AC-S081B-MC

PRODUCT NAME: CXRMDB0 RP04/5/6 RM02/3/5 MOD

PRODUCT DATE: JANUARY 1982

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

RMD IS AN IOMODX WHICH WILL EXERSIZE UP TO 8 RP04/5/6 OR RM02/3/5 DISK DRIVES ON AN RH11/PH70 CONTROLLER, OR DUAL-PORTED BETWEEN TWO SUCH CONTROLLERS. IN SINGLE-PORT MODE, THE ENTIRE SURFACE OF EACH DISK WILL BE TESTABLE (RANDOMLY OR SYSTEMATICALLY).

IN DUAL-PORT MODE, THE "A" CONTROLLER MODULE WILL ACCESS ONLY THE LOW HALF OF THE DISK, AND THE "B" CONTROLLER MODULE WILL ACCESS ONLY THE HIGH HALF OF THE DISK; NO COMMUNICATION WILL BE ATTEMPTED BETWEEN THE TWO PORTS. ONE COPY OF THIS MODULE MUST BE CONFIGURED FOR EACH PORT.
SINGLE/DUAL PORT MODE WILL BE AUTOMATICALLY SELECTED ON THE BASIS OF THE RPDT REGISTER OF EACH SELECTED DRIVE.
DURING TESTING, ANY BAD SECTOR FOUND WILL BE STORED AUTOMATICALLY AND WITHOUT OPERATOR INTERVENTION INTO THE MODULE'S BADSPOT TABLE. A MESSAGE CAN BE PRINTED TO NOTIFY THE OPERATOR OF THE BADSPOT ADDRESS (SEE OPERATOR OPTIONS). SECTORS LISTED IN THE TABLE WILL NOT BE TESTED ON ANY DRIVE.
THE WBUFRQ WORD IN THE HEADER MAY BE CHANGED AT WILL TO ALTER THE MODULE'S TRANSFER SIZE (AND OF COURSE A AND B-PORT SIZES NEED NOT MATCH); THIS WILL NOT ALTER THE EFFECTIVENESS OF THE BADSPOT TABLE OR CAUSE THE MODULE TO ATTEMPT ACCESS OF OUT-OF-RANGE DISK ADDRESSES. WITHIN THIS FRAMEWORK, THE MODULES WILL ACCESS AS MUCH OF THE DISK AS POSSIBLE. RANDOM SEEKS WILL BE USED UNLESS THIS OPTION IS DESELECTED IN SR1.

2. REQUIREMENTS

HARDWARE: 1 TO 8 RP04/5/6 OR RM02/3/5 DRIVES ON 1 OR 2 RH11/RH70 CONTROLLERS

STORAGE:: RMD REQUIRES:
DECIMAL WORDS: 2196
OCTAL WORDS: 4224
OCTAL BYTES: 10450

3. PASS DEFINITION

ONE PASS CONSISTS OF 300 ITERATIONS.
AN ITERATION CONSISTS OF THE FOLLOWING STEPS EXECUTED ON EACH SELECTED DRIVE:

A. WRITE DATA TO DISK. IN DUAL-PORT MODE, PORT "A" MODULE WILL TEST THE LOW HALF OF THE DISK (CYLINDERS 0 TO 407 FOR RP06, CYLINDERS 0 TO 205 FOR RP04/5), PORT "B" WILL EXERSIZE THE HIGH HALF (CYLINDERS 206 TO 410 FOR RP04/5, CYLINDERS 408 TO 814 FOR RP06).
IN SINGLE-PORT MODE, THE MODULE WILL ACCESS THE ENTIRE DISK SURFACE (CYLINDERS 0 TO 410 FOR RP04/5, CYLINDERS 0 TO 814 FOR RP06).
B. WRITE-CHECK DATA JUST WRITTEN.
C. READ 256 WORDS (1 SECTOR) INTO MODULE READ BUFFER.
D. DO IN-CORE COMPARE OF READ BUFFER WITH FIRST 256 WORDS OF WRITE BUFFER.
E. RELEASE THE DRIVE FOR OTHER PORT TO TEST.

4. EXECUTION TIME

ONE PASS OF RMD RUNNING ALONE ON A PDP-11/70 TAKES ABOUT A MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:
DVA=176700 VCT=254 SR1=5 DVC=1

REQUIRED PARAMETERS:
SR1 MUST BE SET UP TO INDICATE WHICH PORT (A/B) THIS COPY OF THE MODULE IS TO TEST, IF ANY DRIVES ARE DUAL-PORTED. ALSO, BIT 7 MUST BE SET IF THE CONTROLLER IS AN RH70, AND CLEARED IF IT IS NOT AN RH70.

6. DEVICE/OPTION SETUP

MAKE SURE ALL DRIVES ARE POWERED UP, WRITE ENABLED, AND READY.
FOR DUAL-PORTED DRIVES: MAKE SURE CONTROLLER SELECT SWITCH IS IN A/B POSITION. IF DRIVE IS NOT CYCLED UP WITH SWITCH IN CORRECT POSITION, PLACE SWITCH CORRECTLY, AND DISABLE THEN RE-ENABLE DRIVE USING DRIVE DISABLE SWITCH.

7. OPERATOR OPTIONS

SR1

BIT 2
CLEAR(0):
TYPE OUT DATA LATE ERRORS (HARD ERROR) AND COUNT THEM INTERNALLY.
SET(1):
COUNT DATA LATE ERRORS INTERNALLY, BUT DO NOT TYPE OUT.

BIT 4
CLEAR(0):
THIS COPY OF RMD TESTS A-PORT ON ANY DUAL-PORTED DRIVES.
SET(1):
THIS COPY OF RMD TESTS B-PORT ON ANY DUAL-PORTED DRIVES.

BIT 5
CLEAR(0):
TEST DISKS WITH RANDOM SEEKS.
SET(1):
DISABLE RANDOM SEEKS. INCREMENT SECTOR BY ONE EACH CYCLE.

BIT 6
CLEAR(0):
DO NOT TYPE OUT BADSPOTS.
SET(1):
ALWAYS TYPE OUT ANY BADSPOT NOT ALREADY RECORDED ON BADSPOT TABLE.

```
            BIT 7
               CLEAR(0):
                 CONTROLLER IS RH11.
               SET(1):
                 CONTROLLER IS RH70 (HAS RHBAE AND RHCS3)

      8.    NON-STANDARD PRINTOUTS
            ---------- ---------

            A.  MOST PRINTOUTS HAVE THE STANDARD FORMATS.

            B.  ERROR MESSAGE DUMP RH11 REGISTERS IN THE FOLLOWING ORDER:
                RHCS1  RHWC   RHBA   RPDA   RHCS2  RPDS   RPER1  RPAS
                RPLA   RHDB   RPMR   RPDT   RPSN   RPOF   RPDC   RPCC
                RPER2  RPER3  RPEC1  RPEC2
                   ;       REVISION DATE: 2-APR-80                          ;WMRMRP
                   ;                                                        ;WMRMRP
                   ; THIS MODULE HAS BEEN MODIFIED TO SUPPORT NOT           ;WMRMRP
                   ; JUST THE RP04/5/6 BUT ALSO THE RM02/3/5 SINGLE AND     ;WMRMRP ;RMDB5
                   ; DUAL PORT.  THE MODIFICATIONS ARE MARKED BY THE        ;WMRMRP
                   ; REFERENCE 'WMRMRP',RMDB, AND RMDB5.                    ;WMRMRP
                   ;                                                        ;WMRMRP
                   ; THE ORDER OF THE REGISTERS WILL VARY SLIGHTLY          ;WMRMRP
                   ; BETWEEN AN RM AND AN RP.  THE DIFFERENCES ARE AS       ;WMRMRP
                   ; FOLLOWS:                                              ;WMRMRP
                   ;       FOR RP          ADDRESS         FOR RM          ;WMRMRP
                   ;       ------          -------         ------          ;WMRMRP
                   ;       RPCC            776736          RMHR            ;WMRMRP
                   ;       RPER2           776740          RMMR2           ;WMRMRP
                   ;       RPER3           776742          RMER2           ;WMRMRP

            C.  THE BAD SECTOR MESSAGE (WHEN ENABLED) LOOKS LIKE:
                'DRIVE X:  BADSPOT AT (OCTAL) CYL:  XX, TRK:  XX, SEC:  XX'
                WHERE 'X' IS AN OCTAL DIGIT.  NO ERROR IS ASSOCIATED WITH THIS
                MESSAGE.

*****************************************************************************
*                                                                           *
*       EDIT:   BY:             DATE:           REASON:                     *
*                                                                           *
*                                                                           *
*  H.E.F.       -       1-JAN-82        - FIX PROBLEMS LISTED IN AIDS REPORT  *
*                                       - # CC0001217 ( 1. RM AS LOAD MEDIA;  *
*                                       - 2. DUAL PORT / WRITE CHECK RETRIES  *
*                                       - EXCEEDED; 3. SYSTEM ERRORS ON RH11  *
*                                       - CONTROLLER ON AN 11/70; 4. INCORRECT*
*                                       - HANDLING OF BAD SPOTS ON LAST SECTOR,
*                                       - 5. DUAL-PORT MXF ERRORS ).          *
*  H.E.F.       -       1-JAN-82        - ADD SUPPORT FOR RM05. THIS WAS ADDED*
*                                       - EVEN THOUGH AN RM05 MODULE (RMCA)   *
*                                       - EXISTS BECAUSE SOMETIMES WE HAVE    *
*                                       - RP04/5/6 AND RM05'S ON THE SAME     *
*                                       - CONTROLLER.                         *
*  H.E.F.       -       1-JAN-82        - AUTO-HANDLING OF BAD SECTOR ERRORS  *
```



```
*                                       - ON RM02/3/5.                        *
*****************************************************************************
-
```

```
                .SBTTL  MODULE HEADER BLOCK                                     ;DRB001
000000'                 IUMODX <RMDB  >,176700,254,5,0,0,1500,104,RBUF,256.,256.        ;WMRMRP ;RMDB5
000000'                 MODULE  150000,RMDB  ,176700,254,5,0,0,1500,104,RBUF,256.,256.
                        .TITLE  RMDB DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6.3    14-AUG-81
                                .LIST   MIN
                        ;*****************************************************************************
000000'                 BEGIN:
000000' 046522  041104      040 MODNAM: .ASCII  /RMDB / ;MODULE NAME.
000005'     000         XFLAG:  .BYTE   0PEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 176700          ADDR:   176700+0                ;1ST DEVICE ADDR.
000010' 000254          VECTOR: 254+0                   ;1ST DEVICE VECTOR.
000012'     240         BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013'     000         BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014' 000001          DVID1:  0+1                     ;DEVICE INDICATOR 1.
000016' 000000          SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000          SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000          SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000          SR4:    OPEN                    ;SWITCH REGISTER 4
                        ;*****************************************************************************
000026' 150000          STAT:   150000                  ;STATUS WORD.
000030' 002374'         INIT:   START                   ;MODULE START ADDR.
000032' 000252'         SPUINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000          PASCNT: 0                       ;PASS COUNTER.
000036' 001500          ICONT:  1500                    ;# OF ITERATIONS PER PASS=1500
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, DP
000102' 000000          ACSR:   OPEN                    ;CONTENTS OF CSR.
000104'                 WASADR:                         ;ADDR OF BAD DATA, DP
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' 002604'         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' 000104          IDNUM:  104                     ;MODULE IDENTIFICATION NUMBER=104
000124' 001242'         RBUFVA: RBUF                    ;READ BUFFER VIRTUAL ADDRESS
```



```
000126' 000000          RBUFPA: OPEN                    ;READ BUFFER PHYSICAL ADDRESS
000130' 000000          RBUFEA: OPEN                    ;READ BUFFER EA BITS
000132' 000400          RBUFSZ: 256.                    ;SIZE OF THE READ BUFFER
000134' 000000          WBUFPA: OPEN                    ;WRITE BUFFER PHYSICAL ADDRESS
000136' 000000          WBUFEA: OPEN                    ;WRITE BUFFER EA BITS
000140' 000400          WBUFRQ: 256.                    ;WRITE BUFFER SIZE REQUESTED
000142' 000000          WBUFSZ: OPEN                    ;WRITE BUFFER SIZE AVAILABLE
000144' 000000          CDERCT: OPEN                    ;CDATA/DATCK ERROR COUNT
000146' 000000          CDWDCT: OPEN                    ;CDATA/DATCK WORD COUNT
000150' 000000          FREE:   OPEN                    ;RESERVED FOR FUTURE USE
        000040                  .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                .NLIST
                                .WORD   0
                                .LIST
                                .ENDR
000252'                 MODSP:
                        ;*****************************************************************************
2                               ;                                                       ;WMRMRP
3                               ;                                                       ;WMRMRP
```

```
                                .SBTTL  MODULE PRIVATE DATA                                     ;DRB001
000252'                         BADSPT:                 ;BAD SPOT TABLE.  THERE IS ROOM FOR 96  ;DRB001
                                                        ;BAD SPOTS, IN TWO-WORD FORMAT, CYLINDER ;DRB001
                                                        ;FIRST, THEN SECTOR IN LOW BYTE OF SECOND ;DRB001
                                                        ;WORD, AND TRACK IN HIGH BYTE.          ;DRB001
000252' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000254' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000256' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000260' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000262' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000264' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000266' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000270' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000272' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000274' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000276' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000300' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000302' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000304' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000306' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000310' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000312' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000314' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000316' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000320' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000322' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000324' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000326' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000330' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000332' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000334' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000336' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000340' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000342' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000344' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000346' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000350' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000352' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000354' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000356' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000360' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000362' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000364' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000366' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000370' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000372' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000374' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000376' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000400' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000402' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000404' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000406' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000410' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000412' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000414' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000416' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
```



```
000420' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000422' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000424' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000426' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000430' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000432' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000434' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000436' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000440' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000442' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000444' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000446' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000450' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000452' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000454' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000456' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000460' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000462' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000464' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000466' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000470' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000472' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000474' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000476' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000500' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000502' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000504' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000506' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000510' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000512' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000514' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000516' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000520' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000522' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000524' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000526' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000530' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000532' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000534' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000536' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000540' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000542' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000544' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000546' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000550' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000552' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000554' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000556' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000560' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000562' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000564' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000566' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000570' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000572' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
000574' 177777                          177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
000576' 177777                          177777          ;CYLINDER ADDRESS                       ;DRB001
```

```
  416  000600' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  417  000602' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  418  000604' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  419  000606' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  420  000610' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  421  000612' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  422  000614' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  423  000616' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  424  000620' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  425  000622' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  426  000624' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  427  000626' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  428  000630' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  429  000632' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  430  000634' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  431  000636' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  432  000640' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  433  000642' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  434  000644' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  435  000646' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  436  000650' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  437  000652' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  438  000654' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  439  000656' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  440  000660' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  441  000662' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  442  000664' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  443  000666' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  444  000670' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  445  000672' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  446  000674' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  447  000676' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  448  000700' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  449  000702' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  450  000704' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  451  000706' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  452  000710' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  453  000712' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  454  000714' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  455  000716' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  456  000720' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  457  000722' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  458  000724' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  459  000726' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  460  000730' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  461  000732' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  462  000734' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  463  000736' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  464  000740' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  465  000742' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  466  000744' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  467  000746' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  468  000750' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  469  000752' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  470  000754' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  471  000756' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
```



```
  472  000760' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  473  000762' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  474  000764' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  475  000766' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  476  000770' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  477  000772' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  478  000774' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  479  000776' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  480  001000' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  481  001002' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  482  001004' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  483  001006' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  484  001010' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  485  001012' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  486  001014' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  487  001016' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  488  001020' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  489  001022' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  490  001024' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  491  001026' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  492  001030' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  493  001032' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  494  001034' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  495  001036' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  496  001040' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  497  001042' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  498  001044' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  499  001046' 177777                    177777          ;CYLINDER ADDRESS                       ;DRB001
  500  001050' 177777                    177777          ;TRACK/SECTOR ADDRESS                   ;DRB001
  501  001052'                    ENDBBK:                                                        ;DRB001
  502                                                                                           ;DRB001
  503          000004               DLATE   = BIT2                                              ;DRB001
  504          000020               BPORT   = BIT4                                              ;DRB001
  505          000040               NOPAND  = BIT5                                              ;DRB001
  506          000100               BDSPT   =BIT6                                               ;DRB001
  507          000200               RH70    = BIT7                                              ;DRB001
  508          137400               HRDCS2  =137400
  509          177777               HRDER1  =177777
  510          167777               HRDER2  =167777
  511          140153               HRDER3  =140153
  512                                                                  ;**-16                   ;DRB001
  513  001052' 000000  000000  000000  S:      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
  514  001060' 000000  000000  000000
  515  001066' 000000  000000  000000
  516  001074' 000000  000000  000000
  517  001102' 000000  000000  000000
  518  001110' 000000  000000  000000
  519  001116' 000000  000000  000000
  520  001124' 000000
  521                               ;DO NOT CHANGE THE ORDER OF THE NEXT 4 LOCATIONS
  522                               ;NEEDED FOR MAP22 ROUTINE
  523  001126' 000000               PA18:   0
  524  001130' 000000               XMEM:   0
  525  001132' 000000               PA22:   0
  526  001134' 000000               EA22:   0
  527                                                                  ;**-5                    ;DRB001
```

```
   528  001136' 000252'                NXTBBK: BADSP1          ; NEXT OPEN BADSPOT IN TABLE                    ;DRB001
   529                                                                                                         ;DRB001
   530  001140' 000000                CYLLIM: 0               ; UPPER DISK LIMIT (DEPENDS ON DISK TYPE)       ;DRB001
   531                                                         ;                                               ;WMRMRP
   532                                                         ; THE FOLLOWING ARE SOME ADDITIONAL LOCATIONS   ;WMRMRP
   533                                                         ; THAT WERE ADDED TO ALLOW THIS MODULE TO       ;WMRMRP
   534                                                         ; SUPPORT THE RP04/5/6 AND THE RM02/3           ;WMRMRP
   535                                                         ;                                               ;WMRMRP
   536  001142' 000000                TYPFLG: 0               ; DRIVE TYPE FLAG. RP04/5/6 = BIT 0=1            ;RMDB5
   537                                                         ;                   RM02/3  = BIT 1=1            ;RMDB5
   538                                                         ;                   RM05    = BIT 2=1            ;RMDB5
   539  001144' 000000                IRKLIM: 0               ; UPPER TRACK LIMIT (DEPENDS ON DISK TYPE)      ;WMRMRP
   540  001146' 000000                IRKLM1: 0               ; UPPER TRACK LIMIT +1 (DEPENDS ON DISK TYPE)   ;WMRMRP
   541  001150' 000000                SECLIM: 0               ; UPPER SECTOR LIMIT (DEPENDS ON DISK TYPE)     ;WMRMRP
   542  001152' 000000                SECLM1: 0               ; UPPER SECTOR LIMIT +1 (DEPENDS ON DISK TYPE)  ;WMRMRP
   543  001154' 000000                CYLMSK: 0               ; MASK USED TO DETERMINE DISK MAX CYL ADDRESS   ;WMRMRP
   544  001156' 000000                TRKMSK: 0               ; MASK USED TO DETERMINE DISK MAX TRK ADDRESS   ;WMRMRP
   545  001160' 000000                SECMSK: 0               ; MASK USED TO DETERMINE DISK MAX SEC ADDRESS   ;WMRMRP
   546                                                         ;                                               ;WMRMRP
   547                                                         ; END OF EXTRA LOCATIONS TO HANDLE RP'S AND RM'S ;WMRMRP
   548                                                         ;                                               ;WMRMRP
   549  001162' 000000                CYLNDR: 0               ;ABSOLUTE CYLINDER ADDRESS OF CURRENT OP.       ;DRB001
   550  001164' 000000                CYL:    0               ;RELATIVE DISK ADDRESS                          ;DRB001
   551  001166'                       DSKADR:                 ;FOR REFERENCING TRACK/SECTOR AS WORD           ;DRB001
   552  001166'     000               SECTOR: .BYTE   0       ;SECTOR ADDRESS OF CURRENT OPERATION            ;DRB001
   553  001167'     000               TRACK:  .BYTE   0       ;TRACK ADDRESS OF CURRENT OPERATION             ;DRB001
   554                                                                                                         ;DRB001
   555  001170' 000000                BADCYL: 0               ;FROM DRIVE LOCATION REGISTERS, ON FINDING BADSPOT;DRB001
   556  001172' 000000                BADSEC: 0               ;FROM RPDA ON BADSPOT OCCURANCE                 ;DRB001
   557  001174' 000000                BADTRK: 0               ;FROM RPDA+1 ON BADSPOT                 ;DRB001
   558                                                                                                         ;DRB001
   559  001176' 000000                LASCYL: 0               ;LAST CYLINDER WHICH CAN BE WRITTEN             ;DRB001
   560  001200' 000000                LASTRK: 0       ;LAST TRACK WHICH CAN BE WRITTEN                        ;DRB001
   561  001202' 000000                LASSEC: 0       ;LAST SECTOR WHICH CAN BE WRITTEN                       ;DRB001
   562                                                                                                         ;DRB001
   563  001204' 000000                SIZTRK: 0       ;NUMBER OF TRACKS IN WBUFSZ                             ;DRB001
   564  001206' 000000                SIZSEC: 0       ;REMAINDER OF WBUFSZ, IN SECTORS                        ;DRB001
   565  001210' 000000                FLAG:   0
   566  001212' 000000                DVICE:  0
   567  001214' 000000                DRIVE:  0
   568  001216' 000000                UNITNO: 0
   569                                                                                 ;**-1           ;LRB001
   570  001220' 000000                DLTCNT: 0
   571  001222' 000000                FUNC:   0
   572  001224' 000000                TIMER:  0
   573  001226' 000000                ZERO:   0
   574  001230' 000000                FERADR: 0
   575  001232' 000000                CLK:    0
   576  001234' 000000                TRY:    0
   577  001236' 000000                STORE:  0
   578  001240' 000000                HRDERR: 0
   579  001242' 000400                RBUF:   .BLKW   256.
   580
   581  002242'                       TABLE:
   582  002242' 001052'                       S
   583  002244' 001054'                       S+2
```



```
   584  002246' 001056'                       S+4
   585  002250' 001060'                       S+6
   586  002252' 001062'                       S+10
   587  002254' 001064'                       S+12
   588  002256' 001066'                       S+14
   589  002260' 001070'                       S+16
   590  002262' 001072'                       S+20
   591  002264' 001074'                       S+22
   592  002266' 001076'                       S+24
   593  002270' 001100'                       S+26
   594  002272' 001102'                       S+30
   595  002274' 001104'                       S+32
   596  002276' 001106'                       S+34
   597  002300' 001110'                       S+36
   598  002302' 001112'                       S+40
   599  002304' 001114'                       S+42
   600  002306' 001116'                       S+44
   601  002310' 001120'                       S+46
   602  002312' 001122'                       S+50
   603  002314' 001124'                       S+52
   604                                                                   ;**-1                     ;DRB001
   605  002316' 177777                        177777
   606  002320' 000000                RHCS1:  0
   607  002322' 000000                RHWC:   0
   608  002324' 000000                RHBA:   0
   609  002326' 000000                RPDA:   0
   610  002330' 000000                RHCS2:  0
   611  002332' 000000                RPDS:   0
   612  002334' 000000                RPER1:  0
   613  002336' 000000                RPAS:   0
   614  002340' 000000                RPLA:   0
   615  002342' 000000                RHDB:   0
   616  002344' 000000                RPMR:   0
   617  002346' 000000                RPDT:   0
   618  002350' 000000                RPSN:   0
   619  002352' 000000                RPOF:   0
   620  002354' 000000                RPDC:   0
   621  002356' 000000                RPCC:   0
   622  002360' 000000                RPER2:  0
   623  002362' 000000                RPER3:  0
   624  002364' 000000                RPEC1:  0
   625  002366' 000000                RPEC2:  0
   626  002370' 000000                RHBAE:  0
   627  002372' 000000                RHCS3:  0
   628                                                                   ;**-4                     ;DRB001
   629                                .SBTTL  MODULE CODE
   630  002374' 012767  000400  175512 START:  MOV     #256.,WDTO      ;256 WORDS TO MEM/ITERATION
   631  002402' 012767  000400  175506        MOV     #256.,WDFP      ;256 WORDS FROM MEM/ITERATION
   632  002410' 012767  000004  175502        MOV     #4,INTR         ;4 INTERRUPTS/ITERATION
   633  002416' 012701  000252'               MOV     #BADSPT,R1      ;BAD SPOT TABLE                 REV D
   634  002422' 022721  177777        4$:     CMP     #-1,(R1)+       ;END YET ?                      REV D
   635  002426' 001375                        BNE     4$                                      ;REV D
   636  002430' 005741                        TST     -(R1)                                   ;REV D
   637  002432' 010167  176500                MOV     R1,NXTBBK       ;RESET POINTER          ;REV D
   638  002436' 016767  175352  176546        MOV     DVID1,DVICE     ;GET DRIVES TO TEST
   639  002444' 123727  000041  000011        CMPB    @#41,#11        ;IF RP IS LOAD MEDIUM THEN
```

```
 640  002452' 001404                          BEQ     13$             ;BEGIN                                  ;RMDB
 641  002454' 123727  000041  000016  14$:    CMPB    @#41,#16        ; SEE IF RM IS LOAD MEDIUM              ;RMDB
 642  002462' 001024                          BNE     3$                                                      ;RMDB
 643  002464' 113700  000040          13$:    MOVB    @#40,R0         ;   GET LOAD-DEVICE NUMBER
 644  002470' 012701  000001                  MOV     #1,R1           ;   INITIALIZE DEVICE MASK
 645  002474' 105700                  1$:     TSTB    R0              ;   WHILE NOT POINTING AT LOAD-DEVICE DO
 646  002476' 001403                          BEQ     2$              ;   BEGIN
 647  002500' 006301                          ASL     R1              ;       SHIFT MASK TO NEXT DEVICE
 648  002502' 105300                          DECB    R0              ;       KEEP TRACK OF SHIFTING
 649  002504' 000773                          BR      1$              ;   END
 650  002506' 130167  176500          2$:     BITB    R1,DVICE        ;   IF LOAD DEVICE IS SELECTED THEN
 651  002512' 001410                          BEQ     3$              ;   BEGIN
 652  002514' 113767  000040  176474          MOVB    @#40,UNITNO     ;       MOVE LOAD-DEVICE NUMBER TO DRIVE
 653  002522' 004767  002726                  JSR     PC,DROP         ;       DROP THE LOAD-DEVICE
 654                                                                  ;   END
 655  002526' 104403  000000' 012504'        MSG$S,BEGIN,LODMED      ;ASCII MESSAGE CALL WITH COMMON HEADER
 656  002534'                         3$:                             ;END
 657  002534' 005067  176424                  CLR     CYL             ;START AT CYLINDER 0                    ;RMDB
 658  002540' 012767  000377  176420          MOV     #377,DSKADR     ;START AT TRACK 0, SECTOR 0
 659                                                                  ;(* INCREMENT BEFORE TEST *)
 660  002546' 004767  005210                  JSR     PC,SETUP
 661  002552' 004767  005112                  JSR     PC,REZET        ;CLEAR THE RH
 662  002556' 032767  000200  175232          BIT     #RH70,SR1       ;RH70 ?                                 ;OJM01
 663  002564' 001004                          BNE     5$              ; NO DONT PRINT BAE & CS3               ;OJM01;WMRMRP
 664  002566' 012767  177777  177516          MOV     #-1,TABLE+50    ; SHORTEN TEMP STORAGE                  ;OJM01;WMRMRP
 665  002574' 000403                          BR      RESTRT          ; GOTO RESTART                          ;WMRMRP
 666  002576' 012767  001122' 177506  5$:     MOV     #S+50,TABLE+50  ; SET UP TO POINT TO DATA IF REG EXISTS ;WMRMRP ;RMDB
 667  002604'                         RESTRT:
 668
 669  002604' 104415  000000' 000124'         GETPAS,BEGIN, RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
 670
 671  002612' 012767  177777  176370  LOOP1:  MOV     #-1,UNITNO      ;PRE-SET UNIT NUMBER
 672  002620' 104414  000000'                 GWBUF$, BEGIN           ;GET WRITE BUFFER INFORMATION
 673                                                                          ;**-1                           ;DRB001
 674  002624'                         LOOP2:
 675  002624' 004767  000756          3$:     JSR     PC,PICKDR       ;GO PICK A DRIVE
 676  002630' 103407                          BCS     1$              ;RETURNS HERE IF ALL DRIVES DONE        ;DRB001
 677  002632' 004767  001574                  JSR     PC,WRTLIM       ;GET DISK LIMITS FOR THIS TYPE OF DISK  ;DRB001
 678  002636' 005067  176372                  CLR     TRY             ;ELSE CLR RE TRY COUNT
 679  002642' 004767  000022                  JSR     PC,CYCLE        ;GO DO A CYCLE ON THIS DRIVE
 680  002646' 000766                          BR      LOOP2           ;DO IT TO NEXT DRIVE
 681
 682  002650' 005767  176336          1$:     TST     DVICE           ;ANYBODY LEFT TO CHECK?
 683  002654' 001002                          BNE     2$              ;BR IF YES
 684  002656' 104410  000000'                 END$,BEGIN              ;
 685  002662'                         2$:
 686  002662' 104413  000000'                 ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
 687                                                                  ;MONITOR SHALL TEST END OF PASS
 688  002666' 000751                          BR      LOOP1           ;BR BACK IF NO
 689
 690  002670'                         CYCLE:
 691                                                                          ;**-4                           ;DRB001
 692  002670' 004767  001004                  JSR     PC,PICKBR       ;SELECT A SECTOR TO TEST                ;DRB001
 693  002674' 004567  000026                  JSR     R5,WRITE        ;GO WRITE A BLOCK
 694  002700' 004567  000116                  JSR     R5,WRITCK       ;GO DO WRITE CHECK
 695  002704' 004567  000206                  JSR     R5,READ         ;GO READ A BLOCK
```



```
 696  002710' 104412  000000' 000126'         CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
 697  002716' 002720'                         .+2                     ; IF ERROR, CONTINUE
 698  002720' 004767  000266                  JSR     PC,RELEASE      ;RELEASE THE DRIVE                      ;DRB001
 699  002724' 000207                          RTS     PC              ;END CYCLE                              ;DRB001
 700
 701                                                                          ;**-6                           ;DRB001
 702                                          .MACRO  LINEUP  EABITS  ; LINE UP EA BITS FOR RHCS1
 703                                  ;       LINEUP  EABITS          ; LINE UP EA BITS FOR RHCS1
 704                                          .NLIST
 705                                          MOV     EABITS,R0       ; GET EXTENDED MEMORY BITS
 706                                          ASL     R0              ; SHIFT 4 PLACES TO THE LEFT
 707                                          ASL     R0              ; TO LINE UP WITH RHCS1
 708                                          ASL     R0              ;
 709                                          ASL     R0              ;
 710                                          MOV     R0,XMEM         ; SAVE THE SHIFTED BITS
 711                                          .LIST
 712                                          .ENDM   LINEUP
 713
 714
 715  002726' 012767  000161  176266  WRITE:  MOV     #161,FUNC       ; LOAD WRITE FUNCTION
 716  002734' 012767  002726' 176266          MOV     #WRITE,FERADR   ;SAVE WHERE WE WERE
 717  002742' 016746  175174                  MOV     WBUFSZ,-(SP)    ;GET WRITE SIZE
 718  002746' 005416                          NEG     (SP)            ;NEGATE IT
 719  002750' 012677  177346                  MOV     (SP)+,@RHWC     ; LOAD WORD COUNT
 720  002754' 016777  175154  177342          MOV     WBUFPA,@RHBA    ; LOAD BUFFER ADDRESS
 721  002762' 016777  176200  177336          MOV     DSKADR,@RPDA    ; LOAD DISK ADDRESS
 722  002770' 016777  176166  177356          MOV     CYLNDR,@RPDC    ; LOAD CYLINDER ADDRESS                 ;DRB001
 723                                  ;       LINEUP  WBUFEA          ; LINE UP EA BITS FOR RHCS1
 724  003016' 000167  000302                  JMP     GO              ; CONTINUE
 725  003022' 012767  000151  176172  WRITCK: MOV     #151,FUNC       ; LOAD WRITE-CHECK FUNCTION
 726  003030' 012767  003022' 176172          MOV     #WRITCK,FERADR  ;SAVE WHERE WE WERE
 727  003036' 016746  175100                  MOV     WBUFSZ,-(SP)    ;GET WRITE SIZE
 728  003042' 005416                          NEG     (SP)            ;NEGATE IT
 729  003044' 012677  177252                  MOV     (SP)+,@RHWC     ; LOAD WORD COUNT
 730  003050' 016777  175060  177246          MOV     WBUFPA,@RHBA    ; LOAD BUFFER ADDRESS
 731  003056' 016777  176104  177242          MOV     DSKADR,@RPDA    ; LOAD DISK ADDRESS
 732  003064' 016777  176072  177262          MOV     CYLNDR,@RPDC    ; LOAD CYLINDER ADDRESS                 ;DRB001
 733                                  ;       LINEUP  WBUFEA          ; LINE UP EA BITS FOR RHCS1
 734  003112' 000167  000206                  JMP     GO              ; CONTINUE
 735  003116' 012767  000171  176076  READ:   MOV     #171,FUNC       ; LOAD READ FUNCTION
 736  003124' 012767  003116' 176076          MOV     #READ,FERADR    ;SAVE WHERE WE WERE
 737  003132' 016746  174774                  MOV     RBUFSZ,-(SP)    ;GET READ SIZE
 738  003136' 005416                          NEG     (SP)            ;NEGATE IT
 739  003140' 012677  177156                  MOV     (SP)+,@RHWC     ; LOAD WORD COUNT
 740  003144' 016777  174756  177152          MOV     RBUFPA,@RHBA    ; LOAD BUFFER ADDRESS
 741  003152' 016777  176010  177146          MOV     DSKADR,@RPDA    ; LOAD DISK ADDRESS
 742  003160' 016777  175776  177166          MOV     CYLNDR,@RPDC    ; LOAD CYLINDER ADDRESS                 ;DRB001
 743                                  ;       LINEUP  RBUFEA          ; LINE UP EA BITS FOR RHCS1
 744  003206' 000167  000112                  JMP     GO              ; CONTINUE
 745  003212' 016777  176000  177110  RELEASE: MOV     UNITNO,@RHCS2   ;SET UP UNIT TO RELEASE                 ;DRB001
 746  003220' 012777  000013  177072          MOV     #13,@RHCS1      ;EXECUTE RELEASE COMMAND                ;DRB001
 747  003226' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
 748  003232' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
 749  003236' 000207                          RTS     PC              ;IT'S RELEASED                          ;DRB001
 750                                                                          ;**-33                          ;DRB001
 751
```

```
  752  003240' 016777  175752  177062  CLEAR:  MOV     UNITNO,@#RHCS2  ; LOAD UNIT ADDRESS
  753  003246' 012777  000011  177044          MOV     #11,@#RHCS1     ; ISSUE A DRIVE CLEAR
  754  003254' 000240                          NOP                     ;WAIT
  755  003256' 000240                          NOP                     ;FOR DRIVE CLEAR TO FINISH
  756  003260' 012777  000023  177032          MOV     #23,@#RHCS1     ;ISSUE A PACK ACK
  757  003266' 105777  177026          1$:     TSTB    @#RHCS1
  758  003272' 100401                          BMI     2$
  759  003274' 000774                          BR      1$              ; NO, WAIT TILL DONE
  760  003276' 017746  177034          2$:     MOV     @#RPAS,-(SP)    ;CLEAR AS BIT
  761  003302' 012677  177030                  MOV     (SP)+,@#RPAS
  762  003306' 012777  040000  177004          MOV     #BIT14,@#RHCS1  ; CLEAR ANY CONTROLLER ERRORS
  763  003314' 012777  010000  177030          MOV     #BIT12,@#RPOF   ; SET BIT FOR 11 FORMAT
  764  003322' 000205                          RTS     R5              ; RETURN
  765
  766  003324' 016777  175666  176776  GO:     MOV     UNITNO,@#RHCS2  ; LOAD UNIT SELECT
  767  003332' 032767  001000  174516          BIT     #ADDR22,PES1    ;22 BIT SUPPORT?                        ;DRB001
  768  003340' 001434                          BEQ     1$              ;NO                                     ;DRB001
  769  003342' 017767  176756  175556          MOV     @#RHBA,PA18     ;GET 18 BIT ADDR
  770  003350' 006267  175554                  ASR     XMEM            ;SHIFT PA BITS TO POSITION 4,5
  771  003354' 006267  175550                  ASR     XMEM
  772  003360' 006267  175544                  ASR     XMEM
  773  003364' 006267  175540                  ASR     XMEM
  774  003370' 104416  000000' 001126'         MAP22$, BEGIN,PA18              ; GET 22-BIT ADDR FROM 18-BIT ADDR
  775  003376' 016777  175530  176720          MOV     PA22,@#RHBA     ;LOAD BA REG
  776  003404' 016777  175524  176756          MOV     EA22,@#RHBAE    ;LOAD BAE REG
  777  003412' 042767  000034  175514          BIC     #34,EA22        ;CLEAR UNWANTED BITS
  778  003420' 000367  175510                  SWAB    EA22            ;LOAD INTO BITS 8,9
  779  003424' 016767  175504  175476          MOV     EA22,XMEM       ;LOAD XMEM TO SET INTO FUNCTION CODE
  780  003432' 056767  175472  175562  1$:     BIS     XMEM,FUNC       ; LOAD EXTENDED MEMORY BITS
  781  003440' 016777  175556  176652          MOV     FUNC,@#RHCS1    ; EXECUTE THE FUNCTION
  782  003446' 104400  000000'                 EXIT$S,BEGIN                    ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
  783  003452' 010046                  NTRUPT: MOV     R0,-(SP)        ;SAVE R0
  784  003454' 016700  175536                  MOV     UNITNO,R0
  785  003460' 116000  003670'                 MOVB    BITTAB(R0),R0
  786  003464' 017746  176646                  MOV     @#RPAS,-(SP)
  787  003470' 040016                          BIC     R0,(SP)
  788  003472' 012677  176640                  MOV     (R6)+,@#RPAS
  789  003476' 012600                          MOV     (SP)+,R0
  790                                          ;---------------------------------------------------------------
  791  003500' 000004  000000' 003506'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
  792                                          ;---------------------------------------------------------------
  793
  794  003506' 004767  003336          1$:     JSR     PC,ERRORS       ;GO CHECK FOR ERRORS
  795  003512' 103401                          BCS     2$              ;ERRORS DETECTED                        ;DRB001
  796  003514' 000205                          RTS     R5              ;OTHERWISE,  RETURN OK
  797  003516'                         2$:
  798
  799  003516' 105767  175516                  TSTB    HRDERR          ;DON'T RETRY ON HARD ERROR      REV D
  800  003522' 100422                          BMI     4$                                              ;DRB001
  801  003524' 005267  175504                  INC     TRY             ;COUNT AN ERROR
  802  003530' 026727  175500  000003          CMP     TRY,#3          ;TOO MANY FOR THIS CYCLE?               ;DRB001
  803  003536' 002007                          BGE     5$              ;BR IF SO
  804  003540' 004567  003750                  JSR     R5,READY        ; MAKE SURE WE HAVE THE DISK            ;RMDB
  805  003544' 103402                          BCS     6$                                              ;RMDB
  806  003546' 004767  003544                  JSR     PC,NOTRDY                                       ;RMDB
  807  003552' 000177  175452          6$:     JMP     @#ERADR         ;RE-EXECUTE THE DRIVER ROUTINE          ;DRB001 ;RMDB
```



```
  808  003556' 104403  000000' 012216'  5$:     ASGN$,BEGIN,EXCED
  809  003564' 004767  003370                  JSR     PC,BAD          ;UPDATE BAD BLOCK INFO.
  810  003570' 012605                  4$:     MOV     (SP)+,R5        ;RESTORE R5
  811                                                                  ;**-7                                   ;DRB001
  812  003572' 004767  000102          3$:     JSR     PC,PICKBR       ;TRY A DIFFERENT BLOCK
  813  003576' 005367  175414                  DEC     UNITNO          ;WANT TO RE-DO SAME DRIVE WERE ON
  814  003602' 000167  177016                  JMP     LOOP2           ;GO DO IT
  815                                                                  ;**-2                                   ;DRB001
  816
  817                                  ;       ------------------------
  818
  819                                                                  ;**-17                                  ;DRB001
  820
  821  003606'                         PICKDR:
  822  003606' 005267  175404          1$:     INC     UNITNO          ;POINT TO NEXT DRIVE
  823  003612' 026727  175400  000010          CMP     UNITNO,#8.              ;DONE LOOKING?
  824  003620' 001002                          BNE     2$              ;BR IF NO, ELSE
  825  003622' 000261                          SEC                     ;SET CARRY FOR NO-MORE-DRIVES           ;DRB001
  826  003624' 000420                          BR      5$              ;EXIT ROUTINE                           ;DRB001
  827  003626' 016700  175364          2$:     MOV     UNITNO,R0       ;USE AS AN INDEX
  828  003632' 136067  003670' 175352          BITB    BITTAB(R0),DVICE        ;TEST THIS DEVICE?
  829  003640' 001001                          BNE     3$              ;BR IF YES, ELSE
  830  003642' 000761                          BR      1$
  831  003644'                         3$:
  832                                                                  ;**-1                                   ;DRB001
  833  003644' 004567  003644                  JSR     R5,READY        ;SEE IF DRIVE IS READY
  834  003650' 103402                          BCS     4$              ;BR IF IT WAS READY                     ;DRB001
  835  003652' 004767  003440                  JSR     PC,NOTRDY       ;ELSE GO CLEAR IT AND CHECK AGAIN
  836  003656' 004767  000670          4$:     JSR     PC,GETDVT       ;IS THIS LEGAL DRIVE TYPE?              ;DRB001
  837  003662' 103751                          BCS     1$              ;ILLEGAL TYPE: DROPPED.  TRY NEXT       ;DRB001
  838  003664' 000241                          CLC                     ;RETURN, NEXT DRIVE AVAILABLE           ;DRB001
  839  003666' 000207                  5$:     RTS     PC              ;GO HOME                                ;DRB001
  840  003670' 001001                  BITTAB: 1001
  841  003672' 004004                          4004
  842  003674' 020020                          20020
  843  003676' 100100                          100100
  844
  845
  846  003700'                         PICKBR:                                 ;**-11                          ;DRB001
  847                                          ;(* FUNCTION: SELECT NEXT DISK ADDRESS TO TEST.       *)     ;DRB001
  848                                          ;(* AVOID WRITING OVER KNOWN BAD SPOTS OR DISK'S BAD  *)     ;DRB001
  849                                          ;(* SPOT FILE (IF OPERATOR TELLS US IT'S THERE).  CYL *)     ;DRB001
  850                                          ;(* IS "RELATIVE ADDRESS", MAX RANGE 0-410 (RP04/5) OR*)     ;DRB001
  851                                          ;(* 0-407 (RP06);  THIS IS MOVED TO CYLNDR AND INCRE- *)     ;DRB001
  852                                          ;(* MENTED TO SECOND HALF IF THIS COPY OF RPD IS IN   *)     ;DRB001
  853                                          ;(* PORT B MODE.                                      *)     ;DRB001
  854                                                                                                       ;DRB001
  855  003700' 032767  000040  174110          BIT     #NORAND,SR1     ;IF SEQUENTIAL SEEKS THEN               ;DRB001
  856  003706' 001403                          BEQ     1$              ;BEGIN                                  ;DRB001
  857  003710' 105267  175252                  INCB    SECTOR          ; SECTOR := SECTOR + 1                  ;DRB001
  858  003714' 000432                          BR      2$              ;END                                    ;DRB001
  859  003716'                         1$:                             ;ELSE (* RANDOM SEEKS *)                ;DRB001
  860                                                                  ;BEGIN (* FUDGE 3 UN-RELATED            ;DRB001
  861                                                                  ;  (* NUMBERS FROM 1 RAND CALL  *)      ;DRB001
  862  003716' 104417  000000'                 RAND$,BEGIN
  863  003722' 016700  174126                  MOV     RANNUM,R0       ; R0 := RANNUM                          ;DRB001
```

```
 864  003726' 046700  175222            BIC     CYLMSK,R0       ; R0 := R0 MOD 1024.                    ;DRB001;WMRMRP
 865  003732' 010067  175226            MOV     R0,CYL          ; CYL := R0                             ;DRB001
 866  003736' 016700  174112            MOV     RANNUM,R0       ; "RECHARGE" R0                         ;DRB001
 867  003742' 000300                    SWAB    R0              ; R0<0:8> := R0<8:8>                    ;DRB001
 868  003744' 046700  175206            BIC     TRKMSK,R0       ; R0 := R0 MOD 32.                      ;DRB001;WMRMRP
 869  003750' 110067  175212            MOVB    R0,SECTOR       ; SECTOR := R0                          ;DRB001
 870  003754' 016700  174074            MOV     RANNUM,R0       ; AND AGAIN                             ;DRB001
 871  003760' 000241                    CLC                     ; CARRY := 0                            ;DRB001
 872  003762' 006100                    ROL     R0              ; CARRY := R0<15:1>                     ;DRB001
 873  003764' 006100                    ROL     R0                                                      ;DRB001
 874  003766' 006100                    ROL     R0              ;                                       ;DRB001
 875  003770' 006100                    ROL     R0              ;                                       ;DRB001
 876  003772' 046700  175162            BIC     SECMSK,R0       ; R0 := (R0<0:13>*8.) + R0<13:3>        ;DRB001;WMRMRP
 877                                                            ; (* NOTE: THE FORMULAE USED ARE *)      ;DRB001
 878                                                            ; (* ABSOLUTELY IRRELEVANT       *)      ;DRB001
 879  003776' 110067  175165            MOVB    R0,TRACK        ; TRACK := R0                           ;DRB001
 880  004002'                    2$:                             ;END                                    ;DRB001
 881  004002' 126767  175160  175140    CMPB    SECTOR,SECLIM   ;IF SECTOR TOO GREAT THEN               ;DRB001;WMRMRP
 882  004010' 003412                    BLE     3$              ;BEGIN                                  ;DRB001
 883  004012' 105267  175151            INCB    TRACK           ; TRACK := TRACK + 1                    ;DRB001
 884  004016' 016700  175144            MOV     SECTOR,R0       ; R0 := SECTOR                          ;DRB001
 885  004022' 016701  175124            MOV     SECLM1,R1       ; R1 := # OF SECTORS IN TRACK           ;DRB001;WMRMRP
 886  004026' 004767  001370            JSR     PC,MODULUS      ; R0 := R0 MOD R1                       ;DRB001
 887  004032' 110067  175130            MOVB    R0,SECTOR       ; SECTOR := R0                          ;DRB001
 888  004036'                    3$:                             ;END                                    ;DRB001
 889                                    ;(* NOW, CUT VALUES DOWN TO LEGAL RANGES FOR CURRENT DRIVE TYPE *) ;DRB001
 890  004036' 126767  175125  175100    CMPB    TRACK,TRKLIM    ;IF TRACK TOO GREAT THEN                ;DRB001;WMRMRP
 891  004044' 003412                    BLE     4$              ;BEGIN                                  ;DRB001
 892  004046' 005267  175112            INC     CYL             ; CYL := CYL + 1                        ;DRB001
 893  004052' 116700  175111            MOVB    TRACK,R0        ; R0 := TRACK                           ;DRB001
 894  004056' 016701  175064            MOV     TRKLM1,R1       ; R1 := TRK LIMIT +1                    ;DRB001;WMRMRP
 895  004062' 004767  001334            JSR     PC,MODULUS      ; R0 := R0 MOD R1                       ;DRB001
 896  004066' 110067  175075            MOVB    R0,TRACK        ; TRACK := R0                           ;DRB001
 897  004072'                    4$:                             ;END                                    ;DRB001
 898  004072' 026767  175066  175040    CMP     CYL,CYLLIM      ;IF CYL GT # OF CYLINDERS IN RANGE THEN ;DRB001
 899  004100' 002420                    BLT     5$              ;BEGIN                                  ;DRB001
 900  004102' 032767  000040  173706    BIT     #NORAND,SR1     ; IF SEQUENTIAL SEEKS THEN              ;DRB001
 901  004110' 001404                    BEQ     6$              ; BEGIN                                 ;DRB001
 902  004112' 005067  175046            CLR     CYL             ;  RESET CYLINDER                       ;DRB001
 903  004116' 005067  175044            CLR     DSKADR          ;  TRACK := 0, SECTOR := 0              ;DRB001
 904  004122'                    6$:                             ; ELSE                                  ;DRB001
 905                                                            ; BEGIN                                 ;DRB001
 906  004122' 016700  175036            MOV     CYL,R0          ;  R0 := CYL                            ;DRB001
 907  004126' 016701  175006            MOV     CYLLIM,R1       ;  R1 := CYLLIM                         ;DRB001
 908  004132' 004767  001264            JSR     PC,MODULUS      ;  R0 := R0 MOD R1                      ;DRB001
 909  004136' 010067  175022            MOV     R0,CYL          ;  CYL := R0                            ;DRB001
 910  004142'                    7$:                             ; END                                   ;DRB001
 911  004142'                    5$:                             ;END                                    ;DRB001
 912  004142' 016767  175016  175012    MOV     CYL,CYLNDR      ;CYLNDR := CYL                          ;DRB001
 913  004150' 032767  000020  173640    BIT     #BPORT,SR1      ;IF BPORT AND DUAL PORT THEN            ;DRB001
 914  004156' 001407                    BEQ     8$              ;BEGIN                                  ;DRB001
 915  004160' 032767  000001  175022    BIT     #BITO,FLAG      ;                                       ;DRB001
 916  004166' 001403                    BEQ     8$              ;                                       ;DRB001
 917  004170' 066767  174744  174764    ADD     CYLLIM,CYLNDR   ; INCREMENT TO SECOND HALF OF DISK      ;DRB001
 918  004176'                    8$:                             ;END                                    ;DRB001
 919  004176' 026767  174760  174772    CMP     CYLNDR,LASCYL   ;IF CYLNDR GT LASCYL THEN               ;DRB001
```



```
 920  004204' 003431                    BLE     20$             ;BEGIN          ;REV. 00
 921  004206' 016767  174764  174746    MOV     LASCYL,CYLNDR   ; CYLNDR := LASCYL (* TRUNCATE *)       ;DRB001
 922  004214' 032767  000040  173574    BIT     #NORAND,SR1     ; IF SEQUENTIAL THEN                    ;DRB001
 923  004222' 001404                    BEQ     10$             ; BEGIN                                 ;DRB001
 924  004224' 005067  174734            CLR     CYL             ;  RESET CYLINDER                       ;DRB001
 925  004230' 005067  174732            CLR     DSKADR          ;  TRACK := 0, SECTOR := 0              ;DRB001
 926  004234'                    10$:                            ; END                                   ;DRB001
 927  004234' 126767  174727  174736    CMPB    TRACK,LASTRK    ; IF TRACK GT LASTRK THEN               ;DRB001
 928  004242' 003412                    BLE     11$             ; BEGIN                                 ;DRB001
 929  004244' 116767  174730  174715    MOVB    LASTRK,TRACK    ;  TRACK := LASTRK                      ;DRB001
 930  004252' 126767  174710  174722    CMPB    SECTOR,LASSEC   ;  IF SECTOR GT LASSEC THEN             ;DRB001
 931  004260' 003403                    BLE     12$             ;  BEGIN                                ;DRB001
 932  004262' 116767  174714  174676    MOVB    LASSEC,SECTOR   ;   SECTOR := LASSEC                    ;DRB001
 933  004270'                    12$:                            ;  END                                  ;DRB001
 934  004270'                    11$:                            ; END                                   ;DRB001
 935
 936  004270'                    20$:                                                                    ;DRB001
 937                                                                                                     ;DRB001
 938                                    ;(* NOW, CHECK BADSPOT TABLE *)                                  ;DRB001
 939  004270' 012700  000252'            MOV     #BADSPT,R0      ;R0 := ADDRESS OF TABLE                 ;DRB001
 940  004274' 016701  174636            MOV     NXTBAD,R1       ;R1 := END OF TABLE                     ;DRB001
 941  004300' 116702  174662            MOVB    SECTOR,R2       ;R2 := SECTOR                           ;DRB001
 942  004304' 116703  174657            MOVB    TRACK,R3        ;R3 := TRACK                            ;DRB001
 943  004310' 016704  174646            MOV     CYLNDR,R4       ;R4 := CYLNDR                           ;DRB001
 944                                    ;(* CALCULATE UPPER LIMIT OF THIS WRITE *)                       ;DRB001
 945  004314' 066702  174666            ADD     SIZSEC,R2       ;R2 := R2 + # OF SECTORS IN WRITE        ;DRB001
 946  004320' 020267  174624            CMP     R2,SECLIM       ;IF R2 GT SECTOR LIMIT THEN             ;DRB001;WMRMRP
 947  004324' 003401                    BLE     13$             ;BEGIN                                  ;DRB001
 948  004326' 005203                    INC     R3              ; R3 := R3 + 1 (INCR. UPPER TRACK)      ;DRB001
 949                                                                                                             ;RMDB
 950  004330'                    13$:                            ;END                                    ;DRB001
 951  004330' 066703  174650            ADD     SIZTRK,R3       ;R3 := R3 + TRKSIZ                      ;DRB001
 952  004334' 020367  174604            CMP     R3,TRKLIM       ;IF R3 GT TRACK LIMIT THEN              ;DRB001;WMRMRP
 953  004340' 003401                    BLE     14$             ;BEGIN                                  ;DRB001
 954  004342' 005204                    INC     R4              ; R4 := R4 + 1 (INCR. UPPER CYLINDER)   ;DRB001
 955                                                                                                             ;RMDB
 956  004344'                    14$:                            ;END                                    ;DRB001
 957  004344' 020001                    15$:    CMP     R0,R1           ;WHILE R0 LT R1 DO              ;DRB001
 958  004346' 002030                    BGE     16$             ;BEGIN                                  ;DRB001
 959  004350' 026710  174606            CMP     CYLNDR,(R0)     ; IF CYLNDR LE (R0) AND R4 GE (R0)      ;DRB001
 960  004354' 003022                    BGT     17$             ;  AND TRACK LE 3(R0) AND R3 GE 3(R0)   ;DRB001
 961  004356' 020410                    CMP     R4,(R0)         ;  AND SECTOR LE 2(R0) AND R2 GE 2(R0) THEN;DRB001
 962  004360' 002420                    BLT     17$             ;                                       ;DRB001
 963  004362' 126760  174601  000003    CMPB    TRACK,3(R0)     ;                                       ;DRB001
 964  004370' 003014                    BGT     17$                                                     ;DRB001
 965  004372' 120360  000003            CMPB    R3,3(R0)        ;                                       ;DRB001
 966  004376' 002411                    BLT     17$             ;                                       ;DRB001
 967  004400' 126760  174562  000002    CMPB    SECTOR,2(R0)    ;                                       ;DRB001
 968  004406' 003005                    BGT     17$             ;                                       ;DRB001
 969  004410' 120260  000002            CMPB    R2,2(R0)        ;                                       ;DRB001
 970  004414' 002402                    BLT     17$             ; BEGIN                                 ;DRB001
 971  004416' 000167  177256            JMP     PICKBK          ;  TRY ANOTHER SECTOR                   ;DRB001
 972  004422'                    17$:                            ; END                                   ;DRB001
 973  004422' 062700  000004            ADD     #4,R0           ; INCREMENT TO NEXT BAD SPOT            ;DRB001
 974  004426' 000746                    BR      15$             ;END                                    ;DRB001
 975  004430'                    16$:                            ;                                       ;DRB001
```

```
 976  004430' 000207                         RTS     PC              ;CPU GO HOME!!!!                                ;DRB001
 977                                                                                                                    ;DRB001
 978                                                                                                                    ;DRB001
 979  004432'                                WRTLIM:                                                                    ;DRB001
 980                                                 ;(* WE ALLOW USER TO CHANGE WBUFSZ "AT WILL"--THEREFORE *)      ;DRB001
 981                                                 ;(* WE DO NOT AUTOMATICALLY KNOW THE TRANSFER SIZE.    *)      ;DRB001
 982                                                 ;(* COMPUTATION IS MADE EASIER, FURTHERMORE, IF WE HAVE *)      ;DRB001
 983                                                 ;(* THE SIZE BROKEN INTO TRACKS AND SECTORS (SINCE      *)      ;DRB001
 984                                                 ;(* WBUFSZ IS ONE WORD, THE TRANSFER CAN'T BE GREATER   *)      ;DRB001
 985                                                 ;(* THAN 12 TRACKS).  THIS ROUTINE ALSO CALCULATES THE  *)      ;DRB001
 986                                                 ;(* UPPER LIMITS ON DISK WRITES SO AS NOT TO OVERFLOW   *)      ;DRB001
 987                                                 ;(* HIGH CYLINDER.                                      *)      ;DRB001
 988                                                                                                                    ;DRB001
 989  004432' 016700  173504                        MOV     WBUFSZ,R0       ;R0 := ALLOCATED WRITE BUFFER           ;DRB001
 990  004436' 000300                                 SWAB    R0              ;R0 := R0 / 256.                        ;DRB001
 991  004440' 042700  177400                        BIC     #177400,R0      ; (* R0 IS WBUFSZ IN SECTORS *)           ;DRB001
 992  004444' 005067  174534                        CLR     SIZTRK          ;SIZTRK := 0                            ;DRB001
 993  004450' 020067  174476              1$:       CMP     R0,SECLM1       ;WHILE R0 GT SEC LIMIT DO               ;DRB001;WMRMRP
 994  004454' 002405                                BLT     2$              ;BEGIN                                  ;DRB001
 995  004456' 105267  174522                        INCB    SIZTRK          ; SIZTRK := SIZTRK + 1                  ;DRB001
 996  004462' 166700  174464                        SUB     SECLM1,R0       ; DECREMENT SECTORS BY 1 TRACK          ;DRB001;WMRMRP
 997  004466' 000770                                BR      1$              ;END                                    ;DRB001
 998  004470' 010067  174512              2$:       MOV     R0,SIZSEC       ;SIZSEC := REMAINDER                    ;DRB001
 999  004474' 016701  174444                        MOV     TRKLIM,R1       ;R1 := HIGH TRACK                       ;DRB001;WMRMRP
1000  004500' 016702  174444                        MOV     SECLIM,R2       ;R2 := HIGH SECTOR                      ;DRB001;WMRMRP
1001  004504' 160002                                SUB     R0,R2           ;R2 := R2 - SIZSEC                      ;DRB001
1002  004506' 005702                                TST     R2              ;IF R2 LT 0 THEN                        ;DRB001
1003  004510' 002003                                BGE     3$              ;BEGIN                                  ;DRB001
1004  004512' 066702  174434                        ADD     SECLM1,R2       ; INCREMENT BACK UP                     ;DRB001;WMRMRP
1005  004516' 005301                                DEC     R1              ; CAN'T FIT THAT LAST TRACK...          ;DRB001
1006  004520'                              3$:                               ;END                                    ;DRB001
1007  004520' 166701  174460                        SUB     SIZTRK,R1       ;R1 := R1 - SIZTRK                      ;DRB001
1008  004524' 005701                                TST     R1              ;IF R1 LT 0 THEN                        ;DRB001
1009  004526' 002004                                BGE     4$              ;BEGIN                                  ;DRB001
1010  004530' 066701  174412                        ADD     TRKLM1,R1       ; INCREMENT R1 BACK UP                  ;DRB001;WMRMRP
1011  004534' 005367  174436                        DEC     LASCYL          ; LASCYL := LASCYL - 1                  ;DRB001
1012  004540'                              4$:                               ;END                                    ;DRB001
1013  004540' 010167  174434                        MOV     R1,LASTRK       ;LASTRK := HIGHEST TRACK WE CAN WRITE   ;DRB001
1014  004544' 010267  174432                        MOV     R2,LASSEC       ;LASSEC := HIGHEST SECTOR WE CAN WRITE  ;DRB001
1015  004550' 000207                                RTS     PC              ;END OF ROUTINE                         ;DRB001
1016  004552'                              GETDVT:                                                                   ;DRB001
1017                                                                                                                   ;DRB001
1018                                                ;(* FIND WHICH TYPE OF DRIVE THIS IS.  IF RP04/5, SET CYLLIM *) ;DRB001
1019                                                ;(* TO 206.  IF RP06, SET CYLLIM TO 408.  DROP THE DRIVE IF  *) ;DRB001
1020                                                ;(* IT IS NOT A RP04/5/6 OR RM02/3                   *)    ;DRB001
1021  004552' 016777  174440  175550                MOV     UNITNO,@#RHCS2  ;SELECT THE DRIVE                       ;DRB001
1022                                                                                                                   ;DRB001
1023                                                                        ;                                                   ;RMDB5
1024                                                                        ; THE FOLLOWING RPDT CHECKS WERE        ;WMRMRP ;RMDB5
1025                                                                        ; INCLUDED FOR RM05 SUPPORT             ;WMRMRP ;RMDB5
1026  004560' 027727  175562  024027        525$:   CMP     @#RPDT,#24027   ;                                       ;WMRMRP ;RMDB5
1027  004566' 001012                                BNE     512$            ; CHECK FOR RM05 DUAL PORT              ;WMRMRP ;RMDB5
1028  004570' 052767  000001  174412        511$:   BIS     #B110,FLAG      ; NO                                    ;WMRMRP ;RMDB5
1029  004576' 012767  000633  174334                MOV     #411.,CYLLIM    ; SET DUAL PORT BIT                     ;WMRMRP ;RMDB5
1030  004604' 012767  001465  174364                MOV     #821.,LASCYL    ; SET CYLINDER LIMIT = 411.              ;RMDB5
1031  004612' 000415                                BR      514$            ; SET LASCYL = 821. (DISK HIGH LIMIT)    ;RMDB5
                                                                              ; NOW GO FINISH UP                       ;RMDB5
```



```
1032  004614' 027727  175526  020027        512$:   CMP     @#RPDT,#20027   ; CHECK FOR RM05 SINGLE PORT              ;RMDB5
1033  004622' 001042                                BNE     515$            ; NO                                    ;RMDB5
1034  004624' 042767  000001  174356        513$:   BIC     #B110,FLAG      ; SET TO SINGLE PORT                    ;RMDB5
1035  004632' 012767  001466  174300                MOV     #822.,CYLLIM    ; SET CYLINDER TO 822.                  ;RMDB5
1036  004640' 012767  001465  174330                MOV     #821.,LASCYL    ; SET DISK HIGH LIMIT TO 821.           ;RMDB5
1037  004646' 012767  000004  174266        514$:   MOV     #4,TYPFLG       ; SET DRIVE TYPE FLAG TO RM05           ;RMDB5
1038  004654' 012767  176000  174272                MOV     #176000,CYLMSK  ; SET MASK FOR RM05 CYLINDER             ;RMDB5
1039  004662' 012767  177740  174266                MOV     #177740,TRKMSK  ; SET MASK FOR RM05 TRACK                ;RMDB5
1040  004670' 012767  177740  174262                MOV     #177740,SECMSK  ; SET MASK FOR RM05 SECTOR MASK          ;RMDB5
1041  004676' 012767  000022  174240                MOV     #18.,TRKLIM     ; SET TRACK LIMIT FOR RM05              ;RMDB5
1042  004704' 012767  000023  174234                MOV     #19.,TRKLM1     ; SET TRACK LIMIT +1 FOR RM05           ;RMDB5
1043  004712' 012767  000037  174230                MOV     #31.,SECLIM     ; SET SECTOR LIMIT FOR RM05             ;RMDB5
1044  004720' 012767  000040  174224                MOV     #32.,SECLM1     ; SET SECTOR LIMIT FOR RM05             ;RMDB5
1045  004726' 000473                                BR      52$             ; GOTO 2$ IN 2 BR'S, 1 IS TOO FAR       ;RMDB5
1046                                                                          ;                                       ;RMDB5
1047  004730'                              515$:                             ; END OF THE RPDT CHECK ADDED CODE FOR RM05 ;RMDB5
1048                                                                        ;                                                   ;RMDB5
1049                                                                        ; THE FOLLOWING RPDT CHECKS WERE        ;WMRMRP
1050                                                                        ; INCLUDED FOR RM02/3 SUPPORT           ;WMRMRP
1051                                                                        ;                                       ;WMRMRP
1052  004730' 027727  175412  024024        25$:    CMP     @#RPDT,#24024   ; CHECK FOR RM03 DUAL PORT              ;WMRMRP
1053  004736' 001404                                BEQ     11$             ; YES                                   ;WMRMRP
1054  004740' 027727  175402  024025                CMP     @#RPDT,#24025   ; CHECK FOR RM02 DUAL PORT              ;WMRMRP
1055  004746' 001012                                BNE     12$             ; NO                                    ;WMRMRP
1056  004750' 052767  000001  174232        11$:    BIS     #B110,FLAG      ; SET DUAL PORT BIT                     ;WMRMRP
1057  004756' 012767  000633  174154                MOV     #411.,CYLLIM    ; SET CYLINDER LIMIT = 411.             ;WMRMRP
1058  004764' 012767  001465  174204                MOV     #821.,LASCYL    ; SET LASCYL = 821. (DISK HIGH LIMIT)   ;WMRMRP
1059  004772' 000421                                BR      14$             ; NOW GO FINISH UP                      ;WMRMRP
1060  004774' 027727  175346  020024        12$:    CMP     @#RPDT,#20024   ; CHECK FOR RM03 SINGLE PORT            ;WMRMRP
1061  005002' 001404                                BEQ     13$             ; YES                                   ;WMRMRP
1062  005004' 027727  175336  020025                CMP     @#RPDT,#20025   ; CHECK FOR RM02 SINGLE PORT            ;WMRMRP
1063  005012' 001042                                BNE     15$             ; NO                                    ;WMRMRP
1064  005014' 042767  000001  174166        13$:    BIC     #B110,FLAG      ; SET TO SINGLE PORT                    ;WMRMRP
1065  005022' 012767  001466  174110                MOV     #822.,CYLLIM    ; SET CYLINDER TO 822.                  ;WMRMRP
1066  005030' 012767  001465  174140                MOV     #821.,LASCYL    ; SET DISK HIGH LIMIT TO 821.           ;WMRMRP
1067  005036' 012767  000002  174076        14$:    MOV     #2,TYPFLG       ; SET DRIVE TYPE FLAG TO RM02/3          ;RMDB5
1068  005044' 012767  176000  174102                MOV     #176000,CYLMSK  ; SET MASK FOR RM02/3 CYLINDER           ;WMRMRP
1069  005052' 012767  177740  174076                MOV     #177740,TRKMSK  ; SET MASK FOR RM02/3 TRACK              ;WMRMRP
1070  005060' 012767  177770  174072                MOV     #177770,SECMSK  ; SET MASK FOR RM02/3 SECTOR MASK        ;WMRMRP
1071  005066' 012767  000004  174050                MOV     #4.,TRKLIM      ; SET TRACK LIMIT FOR RM02/3            ;WMRMRP
1072  005074' 012767  000005  174044                MOV     #5.,TRKLM1      ; SET TRACK LIMIT +1 FOR RM02/3         ;WMRMRP
1073  005102' 012767  000037  174040                MOV     #31.,SECLIM     ; SET SECTOR LIMIT FOR RM02/3           ;WMRMRP
1074  005110' 012767  000040  174034                MOV     #32.,SECLM1     ; SET SECTOR LIMIT FOR RM02/3           ;WMRMRP
1075  005116' 000537                        52$:    BR      2$              ; WE GOT THE STUFF TO DO IT NOW - RETURN ;WMRMRP
1076                                                                        ;                                       ;WMRMRP
1077  005120'                              15$:                              ; END OF THE RPDT CHECK ADDED CODE      ;WMRMRP
1078                                                                        ;                                       ;WMRMRP
1079  005120' 027727  175222  024022                CMP     @#RPDT,#24022   ;IF IT IS AN RP06 DUAL PORT THEN        ;DRB001
1080  005126' 001012                                BNE     1$              ;BEGIN                                  ;DRB001
1081  005130' 052767  000001  174052                BIS     #B110,FLAG      ; DUAL PORT                             ;DRB001
1082  005136' 012767  000627  173774                MOV     #407.,CYLLIM    ; CYLLIM := 408.                        ;DRB001
1083  005144' 012767  001455  174024                MOV     #813.,LASCYL    ; LASCYL := DISK HIGH LIM               ;DRB001
1084  005152' 000471                                BR      5$              ; GO FINISH UP                          ;DRB001;WMRMRP
1085  005154'                              1$:                               ;ELSE                                   ;DRB001
1086  005154' 027727  175166  024021                CMP     @#RPDT,#24021   ;IF IT IS RP04/5 DUAL PORT THEN         ;DRB001
1087  005162' 001404                                BEQ     3$              ;                                       ;DRB001
```

```
1098  005164' 027727  175156  024020          CMP     @RPDT,#24020    ;                               ;DRB001
1099  005172' 001012                          BNE     4$              ;BEGIN                          ;DRB001
1100  005174'                           3$:                                                           ;DRB001
1101  005174' 052767  000001  174006          BIS     #BIT0,FLAG      ; DUAL PORT                     ;DRB001
1102  005202' 012767  000315  173730          MOV     #205.,CYLLIM    ; CYLLIM := 206.                ;DRB001
1103  005210' 012767  000631  173760          MOV     #409.,LASCYL    ; LASCYL := DISK HIGH LIM       ;DRB001
1104  005216' 000447                          BR      5$              ;END                            ;DRB001
1105  005220'                           4$:                           ;ELSE                           ;DRB001
1106  005220' 027727  175122  020022          CMP     @RPDT,#20022    ;IF RP06 SINGLE PORT THEN       ;DRB001
1107  005226' 001012                          BNE     6$              ;BEGIN                          ;DRB001
1108  005230' 042767  000001  173752          BIC     #BIT0,FLAG      ; SINGLE PORT                   ;DRB001
1109  005236' 012767  001456  173674          MOV     #814.,CYLLIM    ; CAN USE ENTIRE DISK           ;DRB001
1110  005244' 012767  001455  173724          MOV     #813.,LASCYL    ; LASCYL := DISK HIGH LIMIT     ;DRB001
1111  005252' 000431                          BR      5$              ; GO FINISH UP                  ;DRB001;WMRMRP
1112  005254'                           6$:                           ;ELSE                           ;DRB001
1113  005254' 027727  175066  020021          CMP     @RPDT,#20021    ;IF RP04/5 SINGLE PORT THEN     ;DRB001
1114  005262' 001404                          BEQ     8$              ;                               ;DRB001
1115  005264' 027727  175056  020020          CMP     @RPDT,#20020    ;                               ;DRB001
1116  005272' 001012                          BNE     9$              ;                               ;DRB001
1117  005274'                           8$:                           ;BEGIN                          ;DRB001
1118  005274' 042767  000001  173706          BIC     #BIT0,FLAG      ; SINGLE PORT                   ;DRB001
1119  005302' 012767  000632  173630          MOV     #410.,CYLLIM    ; USE ENTIRE DISK               ;DRB001
1120  005310' 012767  000631  173660          MOV     #409.,LASCYL    ; LASCYL := DISK HIGH LIMIT     ;DRB001
1121  005316' 000407                          BR      5$              ; GO FINISH UP                  ;DRB001;WMRMRP
1122  005320'                           9$:                           ;ELSE                           ;DRB001
1123                                                                  ;BEGIN                          ;DRB001
1124  005320' 004767  000130                  JSR     PC,DROP         ; DROP THE DRIVE                ;DRB001
1125  005324' 104403  000000' 012306'        MSGN$,BEGIN,BADTYP      ;ASCII MESSAGE CALL WITH COMMON HEADER
1126  005332' 000261                          SEC                     ; ERROR RETURN                  ;DRB001
1127  005334' 000431                          BR      7$              ;                               ;DRB001
1128  005336'                           5$:                           ;                               ;WMRMRP
1129                                                                  ; THE FOLLOWING WERE ADDED WHEN  ;WMRMRP
1130                                                                  ; SUPPORT FOR THE RM02/3 WAS INCLUDED ;WMRMRP
1131                                                                  ;                               ;WMRMRP
1132  005336' 012767  000001  173576          MOV     #1,TYPFLG       ; SET DRIVE TYPE FLAG TO RP     ;WMRMRP ;RMDB5
1133  005344' 012767  176000  173602          MOV     #176000,CYLMSK  ; SET MASK FOR RP04/5/6 CYL     ;WMRMRP
1134  005352' 012767  177740  173576          MOV     #177740,TRKMSK  ; SET MASK FOR RP04/5/6 TRACK   ;WMRMRP
1135  005360' 012767  177740  173572          MOV     #177740,SECMSK  ; SET MASK FOR RP04/5/6 SECTOR  ;WMRMRP
1136  005366' 012767  000022  173550          MOV     #18.,TRKLIM     ; SET THE TRACK LIMIT FOR THE RP'S ;WMRMRP
1137  005374' 012767  000023  173544          MOV     #19.,TRKLM1     ; SET THE TRACK LIMIT +1 FOR THE RP'S ;WMRMRP
1138  005402' 012767  000025  173540          MOV     #21.,SECLIM     ; SET THE SECTOR LIMIT FOR THE RP'S ;WMRMRP
1139  005410' 012767  000026  173534          MOV     #22.,SECLM1     ; SET THE SECTOR LIMIT +1 FOR THE RP'S ;WMRMRP
1140                                                                  ;                               ;WMRMRP
1141                                                                  ; END OF ADDED CODE             ;WMRMRP
1142                                                                  ;                               ;WMRMRP
1143  005416'                           2$:                           ;                               ;DRB001
1144  005416' 000241                          CLC                     ;NO ERROR                       ;DRB001
1145  005420'                           7$:                           ;                               ;DRB001
1146  005420' 000207                          RTS     PC              ;RETURN                         ;DRB001
1147  005422'                           MODLUS:                                                       ;DRB001
1148                                        ;(* THIS LITTLE ROUTINE COMPUTES R0 MODULUS R1      *)   ;DRB001
1149                                        ;(*  I.E., THE REMAINDER OF R0/R1, AND RETURNS THE  *)   ;DRB001
1150                                        ;(* ANSWER IN R0.                                  *)   ;DRB001
1151  005422' 020001                    1$:   CMP     R0,R1           ;WHILE R0 GE R1 DO              ;DRB001
1152  005424' 002402                          BLT     2$              ;BEGIN                          ;DRB001
1153  005426' 160100                          SUB     R1,R0           ; R0 := R0-R1                   ;DRB001
```



```
1154  005430' 000774                          BR      1$              ;END                            ;DRB001
1155  005432'                           2$:                           ;                               ;DRB001
1156  005432' 000207                          RTS     PC              ;THAT'S ALL                     ;DRB001
1157
1158  005434' 012700  001242'         CLRRB:  MOV     #RBUF,R0        ;CLEAR RBUF BUFFER
1159  005440' 016701  172466                  MOV     RBUFSZ,R1       ;GET ITS ADDR AND SIZE
1160  005444' 005020                  CLRCOM: CLR     (R0)+           ;CLEAR ANOTHER
1161  005446' 005301                          DEC     R1              ;COUNT ANOTHER
1162  005450' 001375                          BNE     CLRCOM          ;BR BACK TILL DONE
1163  005452' 000207                          RTS     PC
1164
1165  005454' 012701  000001          DROP:   MOV     #1,R1           ; INITIALIZE DROP PICKER
1166  005460' 016700  173532                  MOV     UNITNO,R0       ; GET THE DRIVE NUMBER
1167  005464' 001403                          BEQ     2$              ; IF DRIVE 0 GO DROP IT
1168  005466' 006301                  1$:     ASL     R1              ; POINT TO NEXT DRIVE
1169  005470' 005300                          DEC     R0              ; IS THIS THE ONE ?
1170  005472' 001375                          BNE     1$              ; NO, LOOK AGAIN
1171  005474' 040167  173512          2$:     BIC     R1,DVICE        ; DROP THE DRIVE
1172                                          ;*****************************************************************
1173                                                                  ;CONVERT UNITNO TO ASCII AND
1174                                                                  ;STORE AT ADR1
1175  005500' 104420  000000' 001216'        OTOA$,BEGIN,UNITNO,ADR1
1176  005506' 012516'
1177                                          ;*****************************************************************
1178  005510' 000207                          RTS     PC              ; RETURN
1179                                  ;       ------------------------
1180
1181
1182
1183
1184                                          ;\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
1185                                          ;\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
1186
1187                                                  ;R E V  D       ;MCP
1188
1189
1190
1191
1192
1193  005512' 005067  000464          ERROR:  CLR     DLTFLG          ;INITIALIZE DLT FLAG
1194  005516' 104403  000000' 012470'        MSGN$,BEGIN,ASTER       ;ASCII MESSAGE CALL WITH COMMON HEADER
1195  005524' 005067  173510                  CLR     HRDERR                  ;RESET ERROR INDICATOR
1196  005530' 032767  100000  173324          BIT     #BIT15,S+10             ;
1197  005536' 001417                          BEQ     14$                     ;
1198  005540' 005267  173454                  INC     DLTCNT                  ;
1199  005544' 052767  000001  000430          BIS     #1,DLTFLG               ;SET FLAG FOR DATA LATE
1200  005552' 032767  000004  172236          BIT     #DLATE,SR1              ;
1201  005560' 001006                          BNE     14$                     ;
1202  005562' 052767  177777  173450          BIS     #-1,HRDERR              ;
1203  005570' 104403  000000' 012222'        MSGN$,BEGIN,DLTERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
1204  005576' 032777  020000  174514  14$:   BIT     #BIT13,@RHCS1           ;MASSBUS CONTROL PARITY ERR?
1205  005604' 001406                          BEQ     5$
1206  005606' 052767  177777  173424          BIS     #-1,HRDERR              ;SET HARD ERR
1207  005614' 104403  000000' 012174'        MSGN$,BEGIN,MCPERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
1208  005622' 032767  137400  173232  5$:     BIT     #HRDCS2,S+10            ;ANY ERRORS ?
1209  005630' 001402                          BEQ     1$
```

```
 1200  005632' 004767  000346                JSR     PC,CHKCS2               ;GO SEE THE BITS
 1201  005636' 032767  040000  173220  1$:    BIT     #BIT14,S+12             ;ANY SET IN THE ERROR REG'S ?
 1202  005644' 001451                        BEQ     4$
 1203  005646' 032767  177777  173212        BIT     #HRDER1,S+14
 1204  005654' 001402                        BEQ     2$
 1205  005656' 004767  000464                JSR     PC,CHKER1
 1206  005662' 032767  000006  173252  2$:    BIT     #6,TYPFLG               ; TEST TO SEE IF THIS IS AN RM02/3/05   ;RMDB5
 1207  005670' 001413                        BEQ     30$                     ; NO - THEN CHECK RPER2 & RPER3         ;RMDB5
 1208  005672' 032767  176210  173214        BIT     #176210,S+42            ; RM - CHECK RMER2 HARD ERROR BITS      ;WMRMRP
 1209  005700' 001433                        BEQ     4$                      ; NO ERRORS IN RMER2 - CONTINUE         ;WMRMRP
 1210  005702' 052767  177777  173330        BIS     #-1,HRDERR              ; SET THE HARD ERROR FLAG               ;WMRMRP
 1211  005710' 104403  000000' 012320'        MSGNS,BEGIN,ER2ERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
 1212  005716' 000424                        BR      4$                      ; CONTINUE                              ;WMRMRP
 1213  005720' 032767  167777  173164  30$:   BIT     #HRDER2,S+40            ;ANY ERRORS IN ER2
 1214  005726' 001406                        BEQ     3$
 1215  005730' 052767  177777  173302        BIS     #-1,HRDERR
 1216  005736' 104403  000000' 012320'        MSGNS,BEGIN,ER2ERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
 1217  005744' 032767  140153  173142  3$:    BIT     #HRDER3,S+42            ;TEST ER3
 1218  005752' 001406                        BEQ     4$
 1219  005754' 052767  177777  173256        BIS     #-1,HRDERR              ;SET HARD ERRROR INDICATOR
 1220  005762' 104403  000000' 012324'        MSGNS,BEGIN,ER3ERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
 1221  005770' 032767  040000  173064  4$:    BIT     #BIT14,S+10             ; WCE ?
 1222  005776' 001406                        BEQ     6$
 1223  006000' 052767  000002  000174        BIS     #2,DLTFLG               ;SET FLAG FOR SOFT TO BE USED IF DLT ALSO OCCURE
 1224  006006' 104403  000000' 012330'        MSGNS,BEGIN,WCEERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
 1225  006014' 032767  040000  173030  6$:    BIT     #BIT14,S                ;IRE  ?
 1226  006022' 001406                        BEQ     7$
 1227  006024' 052767  000002  000150        BIS     #2,DLTFLG               ;SET SOFT FLAG
 1228  006032' 104403  000000' 012166'        MSGNS,BEGIN,IKERR       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1229  006040' 032767  100000  173004  7$:    BIT     #BIT15,S                ;SC ?
 1230  006046' 001406                        BEQ     8$
 1231  006050' 052767  000002  000124        BIS     #2,DLTFLG               ;SET FLAG
 1232  006056' 104403  000000' 012334'        MSGNS,BEGIN,SCERR       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1233  006064' 005767  172772        8$:     TST     S+10
 1234  006070' 100405                        BMI     10$
 1235  006072' 016767  174244  173136        MOV     RMDB,STORE
 1236  006100' 005067  174236                CLR     RMDB
 1237  006104' 005767  173130        10$:    TST     HRDERR                  ;ANNY HARD ERROR'S ?
 1238  006110' 100414                        BMI     12$                     ;IF TRUE THEN BR
 1239  006112' 032767  000001  000062        BIT     #1,DLTFLG               ;DLT ?  #DLATE IN SR1 MUST BE SET IF TRUE
 1240  006120' 001404                        BEQ     22$
 1241  006122' 022767  000003  000052        CMP     #3,DLTFLG               ;I; (SOFT & DLT NOT COUNTED )  ?
 1242  006130' 001007                        BNE     11$
 1243  006132'                        22$:
 1244                                        ;*****************************************************************
 1245  006132' 104406  000000' 002242'        SOFERS,BEGIN,TABLE      ;
 1246                                        ;*****************************************************************
 1247  006140' 000403                        BR      11$                     ;
 1248  006142'                        12$:
 1249                                        ;*****************************************************************
 1250  006142' 104405  000000' 002242'        HRDERS,BEGIN,TABLE      ;
 1251                                        ;*****************************************************************
 1252  006150' 005767  174166        11$:    TST     RMDB
 1253  006154' 001003                        BNE     9$
 1254  006156' 016767  173054  174156        MOV     STORE,RMDB
 1255  006164' 004767  001126        9$:     JSR     PC,NOTRDY
```



```
 1256
 1257  006170' 104403  000000' 012470'        MSGNS,BEGIN,ASTER       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1258  006176' 000261                        SEC
 1259  006200' 000207                        RTS     PC
 1260
 1261  006202' 000000                DLTFLG: .WORD   0                       ;DLT FLAG TO BE USED FOR SOFT ERROR
 1262                                                                        ; & INHIBIT DLT COMPARISONS
 1263
 1264  006204' 052767  177777  173026  CHKCS2: BIS     #-1,HRDERR              ;SET HARD ERROR
 1265  006212' 104403  000000' 012340'        MSGNS,BEGIN,CS2HD       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1266  006220' 032767  020000  172634  1$:    BIT     #BIT13,S+10             ;UPE ?
 1267  006226' 001403                        BEQ     2$
 1268  006230' 104403  000000' 012344'        MSGNS,BEGIN,UPER        ;ASCII MESSAGE CALL WITH COMMON HEADER
 1269  006236' 032767  010000  172616  2$:    BIT     #BIT12,S+10             ;NED ?
 1270  006244' 001403                        BEQ     3$
 1271  006246' 104403  000000' 012350'        MSGNS,BEGIN,NEDER       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1272  006254' 032767  004000  172600  3$:    BIT     #BIT11,S+10             ;NXM ?
 1273  006262' 001403                        BEQ     4$
 1274  006264' 104403  000000' 012354'        MSGNS,BEGIN,NXM ;ASCII MESSAGE CALL WITH COMMON HEADER
 1275  006272' 032767  002000  172562  4$:    BIT     #BIT10,S+10
 1276  006300' 001403                        BEQ     5$
 1277  006302' 104403  000000' 012360'        MSGNS,BEGIN,PGERR       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1278  006310' 032767  001000  172544  5$:    BIT     #BIT9,S+10              ;MXF ?
 1279  006316' 001403                        BEQ     6$
 1280  006320' 104403  000000' 012364'        MSGNS,BEGIN,MXFER       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1281
 1282  006326' 032767  000400  172526  6$:    BIT     #BIT8,S+10
 1283  006334' 001403                        BEQ     7$
 1284  006336' 104403  000000' 012370'        MSGNS,BEGIN,MDPER       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1285  006344' 000207                7$:     RTS     PC
 1286
 1287  006346'                       CHKER1:
 1288  006346' 104403  000000' 012374'        MSGNS,BEGIN,ER1HD       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1289  006354' 032767  100000  172504        BIT     #BIT15,S+14             ;DATA CHECK ERROR ?
 1290  006362' 001406                        BEQ     1$
 1291  006364' 052767  000002  177610        BIS     #2,DLTFLG               ;SET SOFT ERR
 1292  006372' 104403  000000' 012400'        MSGNS,BEGIN,DCKER       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1293  006400' 032767  000020  172460  1$:    BIT     #BIT4,S+14              ;FORMAT ERROR ?
 1294  006406' 001423                        BEQ     2$
 1295  006410' 104403  000000' 012404'        MSGNS,BEGIN,FERER       ;ASCII MESSAGE CALL WITH COMMON HEADER
 1296  006416' 016767  172444  172612        MOV     S+14,STORE      ;SAVE ER1
 1297  006424' 042767  077757  172604        BIC     #77757,STORE
 1298  006432' 022767  100020  172576        CMP     #BIT15 ! BIT4,STORE
 1299  006440' 001003                        BNE     21$
 1300  006442' 052767  177777  172570        BIS     #-1,HRDERR              ;SET HARD ERROR INDICATOR
 1301  006450' 052767  000002  177524  21$:   BIS     #2,DLTFLG               ;SET SOFT ERR
 1302
 1303  006456' 032767  000400  172402  2$:    BIT     #BIT8,S+14              ;HCRC ?
 1304  006464' 001406                        BEQ     3$
 1305  006466' 052767  000002  177506        BIS     #2,DLTFLG               ;SET SOFT ERR
 1306  006474' 104403  000000' 012410'        MSGNS,BEGIN,HCRC        ;ASCII MESSAGE CALL WITH COMMON HEADER
 1307  006502' 032767  000200  172356  3$:    BIT     #BIT7,S+14              ;HEADER COMP ERR ?
 1308  006510' 001420                        BEQ     5$
 1309  006512' 104403  000000' 012414'        MSGNS,BEGIN,HCE ;ASCII MESSAGE CALL WITH COMMON HEADER
 1310  006520' 016767  172342  172510        MOV     S+14,STORE
 1311  006526' 042767  177177  172502        BIC     #177177,STORE           ;SAVE ONLY HCRC & HCE
```

```
1312  006534' 022767  000600  172474          CMP     #600,STORE
1313  006542' 001403                          BEQ     5$                      ;SOFT ERROR IF BOTH SET
1314  006544' 052767  177777  172466          BIS     #-1,HRDERR              ;SET HRD ERR .
1315  006552' 032767  077157  172306  5$:     BIT     #77157,S+14             ;ONLY HARD'S LEFT
1316  006560' 001001                          BNE     6$
1317  006562' 000207                          RTS     PC
1318
1319  006564' 052767  177777  172446  6$:     BIS     #-1,HRDERR              ;SET HRD ERR .
1320  006572' 032767  040000  172266          BIT     #BIT14,S+14             ;UNS ?
1321  006600' 001403                          BEQ     7$
1322  006602' 104403  000000' 012420'        MSGN$,BEGIN,UNSER       ;ASCII MESSAGE CALL WITH COMMON HEADER
1323  006610' 032767  020000  172250  7$:     BIT     #BIT13,S+14             ;OPI ?
1324  006616' 001403                          BEQ     8$
1325  006620' 104403  000000' 012424'        MSGN$,BEGIN,OPIER       ;ASCII MESSAGE CALL WITH COMMON HEADER
1326  006626' 032767  010000  172232  8$:     BIT     #BIT12,S+14             ;DTE ?
1327  006634' 001403                          BEQ     9$
1328  006636' 104403  000000' 012430'        MSGN$,BEGIN,DTER        ;ASCII MESSAGE CALL WITH COMMON HEADER
1329  006644' 032767  004000  172214  9$:     BIT     #BIT11,S+14             ;WLE ?
1330  006652' 001403                          BEQ     10$
1331  006654' 104403  000000' 012434'        MSGN$,BEGIN,WLER        ;ASCII MESSAGE CALL WITH COMMON HEADER
1332  006662' 032767  002000  172176  10$:    BIT     #BIT10,S+14             ;IAE ?
1333  006670' 001403                          BEQ     20$
1334  006672' 104403  000000' 012440'        MSGN$,BEGIN,IAER        ;ASCII MESSAGE CALL WITH COMMON HEADER
1335  006700' 032767  001000  172160  20$:    BIT     #BIT9,S+14              ;                       >
1336  006706' 001403                          BEQ     17$                     ;                       >
1337  006710' 104403  000000' 012474'        MSGN$,BEGIN,AUE ;ASCII MESSAGE CALL WITH COMMON HEADER
1338  006716' 032767  000040  172142  17$:    BIT     #BIT5,S+14              ;
1339  006724' 001403                          BEQ     11$                     ;
1340  006726' 104403  000000' 012500'        MSGN$,BEGIN,WCF ;ASCII MESSAGE CALL WITH COMMON HEADER
1341  006734' 032767  000100  172124  11$:    BIT     #BIT6,S+14              ;ECH ?
1342  006742' 001405                          BEQ     12$
1343  006744' 104403  000000' 012444'        MSGN$,BEGIN,ECHR        ;ASCII MESSAGE CALL WITH COMMON HEADER
1344  006752' 004767  000202                  JSR     PC,BAD
1345  006756' 032767  000010  172102  12$:    BIT     #BIT3,S+14              ;PAR  ?
1346  006764' 001403                          BEQ     13$
1347  006766' 104403  000000' 012450'        MSGN$,BEGIN,PAER        ;ASCII MESSAGE CALL WITH COMMON HEADER
1348  006774' 032767  000004  172064  13$:    BIT     #BIT2,S+14              ;RMR ?
1349  007002' 001403                          BEQ     14$
1350  007004' 104403  000000' 012454'        MSGN$,BEGIN,RMRR        ;ASCII MESSAGE CALL WITH COMMON HEADER
1351  007012' 032767  000002  172046  14$:    BIT     #BIT1,S+14              ;ILR ?
1352  007020' 001403                          BEQ     15$
1353  007022' 104403  000000' 012460'        MSGN$,BEGIN,ILRR        ;ASCII MESSAGE CALL WITH COMMON HEADER
1354  007030' 032767  000001  172030  15$:    BIT     #BIT0,S+14              ;ILF ?
1355  007036' 001403                          BEQ     16$
1356  007040' 104403  000000' 012464'        MSGN$,BEGIN,ILFR        ;ASCII MESSAGE CALL WITH COMMON HEADER
1357  007046' 000207                  16$:    RTS     PC
1358
1359                                          ;       R E V  D   END OF UPDATE
1360
1361
1362                                          ;////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\
1363                                          ;////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\
1364
1365
1366
1367  007050'                         ERRORS:                                                         ;DRB001
```



```
1368  007050' 005777  173244                  TST     @#HCS1          ;IF NO ERROR THEN                       ;DRB001
1369  007054' 100402                          BMI     1$              ;BEGIN                                  ;DRB001
1370  007056' 000241                          CLC                     ; SET NO ERROR                          ;DRB001
1371  007060' 000207                          RTS     PC
1372
1373                                                                  ;ELSE                                   ;DRB001
1374  007062' 032767  000006  172052  1$:     BIT     #6,TYPFLG       ; IF DRIVE TYPE IS RM02/3/ OR RM05       ;RMDB5
1375  007070' 001413                          BEQ     4$              ; (BR IF NOT RM)                        ;RMDB
1376  007072' 032777  100000  173262          BIT     #100000,@RPER3  ; AND IF BIT 15 (MSE) SET IN RMER2       ;RMDB1
1377                                                                  ; (NOTE: RPER3 = RMER2 SEE DOCUMENTATION) ;RMDB1
1378  007100' 001407                          BEQ     4$              ; (BR IF NOT MSE)                       ;RMDB
1379  007102' 000241                          CLC                     ; THEN DON'T COUNT THIS AS AN ERROR      ;RMDB
1380  007104' 004767  000050                  JSR     PC,BAD          ; THIS IS A BAD SECTOR SO ENTER II       ;RMDB1
1381  007110' 062706  000004                  ADD     #4,R6           ; ADJUST SP FOR RETURN TO LOOP2          ;RMDB1
1382  007114' 000167  173504                  JMP     LOOP2           ; GO DO NEXT COMPLETE CYCLE              ;RMDB1
1383
1384
1385  007120' 000261                  4$:     SEC                     ;SET C BIT TO INDICATE ERROR            ;RMDB
1386  007122' 005000                          CLR     R0              ;INIT INDEX
1387  007124' 016701  173170                  MOV     @#HCS1,R1       ; BEGIN OF REGISTER POINTERS            ;DRB001
1388  007130'                         3$:                             ; REPEAT                                ;DRB001
1389                                                                  ; BEGIN                                 ;DRB001
1390  007130' 012160  001052'                 MOV     (R1)+,S(R0)     ;  MOV REGISTER TO TEMPORARY            ;DRB001
1391  007134' 005720                          TST     (R0)+           ;  ADVANCE POINTER                      ;DRB001
1392  007136' 022760  177777  002242'         CMP     #-1,TABLE(R0)   ;END ?    REV D                         ;WRRMRP
1393  007144' 001401                          BEQ     33$             ;YES END OF TABLE                       ;DJM01
1394  007146' 000770                          BR      3$              ;  NO GO DO IT AGAIN                    ;DJM01
1395  007150' 004767  000476          33$:    JSR     PC,ERSUB1       ; GET SOME TRIVIAL INFO                 ;DRB001
1396
1397
1398
1399  007154' 000167  176332                  JMP     EROR            ;               REV. DO  ;RCF
1400
1401  007160'                         BAD:
1402  007160'                         11$:                            ;  BEGIN                                ;DRB001
1403  007160' 116767  172002  172004          MOVB    SECTOR,BADSEC   ;   SET CURRENT SECTOR                  MK002
1404  007166' 116767  171775  172000          MOVB    TRACK,BADTRK    ;   SET CURRENT TRACK                   ;MK002
1405  007174' 016767  171762  171760          MOV     CYLNDR,BADCYL   ;   SET CURRENT CYLINDER                ;MK002
1406  007202' 032767  000100  170606          BIT     #BDSPI,SR1      ;IF OPERATOR WANTS TYPEOUT      ;MK002
1407  007210' 001423                          BEQ     12$             ;ELSE BRANCH AROUND MSG ;MK002
1408                                          ;*****************************************************************
1409                                                                  ;CONVERT UNITNO TO ASCII AND
1410                                                                  ;STORE AT ADR1
1411  007212' 104420  000000' 001216'         OTOA$,BEGIN,UNITNO,ADR1
1412  007220' 012516'
1413                                          ;*****************************************************************
1414                                          ;*****************************************************************
1415
1416                                                                  ;CONVERT BADCYL TO ASCII AND
1417  007222' 104420  000000' 001170'         OTOA$,BEGIN,BADCYL,CYLNO        ;STORE AT CYLNO
1418  007230' 010562'
1419                                          ;*****************************************************************
1420                                          ;*****************************************************************
1421
1422                                                                  ;CONVERT BADTRK TO ASCII AND
1423  007232' 104420  000000' 001174'         OTOA$,BEGIN,BADTRK,TRKNO        ;STORE AT TRKNO
```

```
 1424  007240' 010600'
 1425                                   ;*************************************************************
 1426                                   ;*************************************************************
 1427
 1428                                                           ;CONVERT BADSEC TO ASCII AND
 1429  007242' 104420  000000' 001172'         OTOAS,BEGIN,BADSEC,SECNO  ;STORE AT SECNO
 1430  007250' 010616'
 1431                                   ;*************************************************************
 1432  007252' 104403  000000' 012232'         MSGNS,BEGIN,BADMES        ;ASCII MESSAGE CALL WITH COMMON HEADER
 1433  007260'                          12$:                             ;  END                          ;DRB001
 1434  007260' 026727  171652  001052'         CMP     NXTBBK,#ENDBBK    ;  IF MORE SPACE IN TABLE THEN  ;DRB001
 1435  007266' 002012                          BGE     13$               ;  BEGIN                        ;DRB001
 1436  007270' 016700  171642                  MOV     NXTBBK,R0         ;   R0 POINTS TO NEXT FREE SLOT IN TABLE;DRB001
 1437  007274' 016720  171670                  MOV     BADCYL,(R0)+      ;   STUFF CYLINDER NUMBER       ;DRB001
 1438  007300' 116720  171666                  MOVB    BADSEC,(R0)+      ;   STUFF REST OF DISK ADDRESS  ;MK002
 1439  007304' 116720  171664                  MOVB    BADTRK,(R0)+      ;                               ;MK002
 1440  007310' 010067  171622                  MOV     R0,NXTBBK         ;   UPDATE POINTER              ;DRB001
 1441  007314'                          13$:                             ;  END                          ;DRB001
 1442  007314' 000207                          RTS     PC
 1443
 1444
 1445
 1446
 1447
 1448  007316' 012767  077777  171706  NOTRDY: MOV     #77777,CLK        ; SET THE TIMER
 1449  007324'                          4$:
 1450  007324' 032777  004000  172766          BIT     #BIT11,@RHCS1     ;DO I HAVE THE DRIVE DVA?
 1451  007332' 001406                          BEQ     6$                ;NO
 1452  007334' 010046                          MOV     R0,-(SP)
 1453  007336' 012600                          MOV     (SP)+,R0          ;JUST WASTE A LITTLE TIME
 1454  007340' 032777  004000  172752          BIT     #BIT11,@PHCS1             ;STILL GOT IT?
 1455  007346' 001017                          BNE     2$
 1456  007350'                          6$:
 1457  007350' 104407  000000'                BREAKS,BEGIN              ;TEMPORARY RETURN TO MONITOR....
 1458  007354' 104407  000000'                BREAKS,BEGIN              ;THEN CONTINUE AT NEXT INSTRUCTION.
 1459  007360' 104407  000000'                BREAKS,BEGIN              ;TEMPORARY RETURN TO MONITOR....
 1460  007364' 104407  000000'                BREAKS,BEGIN              ;THEN CONTINUE AT NEXT INSTRUCTION.
 1461  007370' 005367  171636                  DEC     CLK               ;COUNT # OF TRIES
 1462  007374' 001353                          BNE     4$                ;NOT DONE YET
 1463  007376' 104403  000000' 012256'         MSGNS,BEGIN,NOT ;ASCII MESSAGE CALL WITH COMMON HEADER
 1464  007404' 000427                          BR      7$                ;COULD NOT GET DRIVE
 1465  007406' 004567  173626          2$:     JSR     R5,CLEAR          ;SET THE CONTROLLER AND DRIVE
 1466  007412' 004567  000076                  JSR     R5,READY          ; IS DRIVE READY ?
 1467  007416' 000435                          BR      1$                ; YES, CONTINUE
 1468  007420' 004767  000226          5$:     JSR     PC,ERSUB1         ; LOAD ERROR INFORMATION
 1469  007424' 005000                          CLR     R0                ;MOVE DRIVE REG INTO TABLE
 1470  007426' 016701  172666                  MOV     RHCS1,R1
 1471  007432' 012160  001052'          22$:    MOV     (R1)+,S(R0)
 1472  007436' 005720                          TST     (R0)+
 1473  007440' 022760  177777  002242'         CMP     #177777,TABLE(R0)                                 ;WMRMRP
 1474  007446' 001371                          BNE     22$
 1475  007450' 012767  000006  170430          MOV     #6,ERRTYP         ;DRIVE NOT READY
 1476                                   ;*************************************************************
 1477  007456' 104405  000000' 002242'         ERRORS,BEGIN,TABLE        ; DRIVE  NOT READY
 1478                                   ;*************************************************************
 1479                                                                     ;**-1                        ;DRB001
```



```
 1480  007464' 004767  173522          7$:     JSR     PC,RELESE         ;RELEASE THE DRIVE              ;DRB001
 1481  007470' 004767  175760                  JSR     PC,DROP           ; NO, DROP THE DRIVE
 1482  007474' 104403  000000' 012274'         MSGNS,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
 1483  007502' 016706  170324                  MOV     SPDINT,R6
 1484  007506' 000167  173112                  JMP     LOOP2
 1485  007512' 000207                  1$:     RTS     PC                ;RETURN
 1486                                   ;       ------------------------
 1487
 1488
 1489  007514' 016777  171476  172606  READY:  MOV     UNITNO,@RHCS2     ; LOAD UNIT ADDRESS
 1490  007522' 017700  172572                  MOV     @RHCS1,R0
 1491  007526' 032700  004000                  BIT     #BIT11,R0
 1492  007532' 001433                          BEQ     1$
 1493  007534' 017700  172572                  MOV     @RHDS,R0          ; SAVE STATUS IN R0
 1494  007540' 105700                          TSTB    R0                ; DRIVE READY ?
 1495  007542' 100027                          BPL     1$                ; NO
 1496  007544' 032700  000100                  BIT     #BIT6,R0          ; VOLUME VALID ?
 1497  007550' 001424                          BEQ     1$                ; NO
 1498  007552' 032700  000400                  BIT     #BIT8,R0          ; DRIVE PRESENT ?
 1499  007556' 001421                          BEQ     1$                ; NO
 1500  007560' 032700  004000                  BIT     #BIT11,R0         ; WRITE LOCKED ?
 1501  007564' 001016                          BNE     1$                ; YES
 1502  007566' 032700  010000                  BIT     #BIT12,R0         ; MEDIUM ON LINE ?
 1503  007572' 001413                          BEQ     1$                ; NO
 1504  007574' 032700  040000                  BIT     #BIT14,R0         ; ANY ERRORS ?
 1505  007600' 001010                          BNE     1$                ; YES
 1506  007602' 005700                          TST     R0                ; ATTENTION SET ?
 1507  007604' 100406                          BMI     1$                ; YES
 1508  007606' 032777  004000  172504          BIT     #BIT11,@RHCS1     ;DVA SET?
 1509  007614' 001402                          BEQ     1$                ;BR IF NOT
 1510                                                                     ;**-3                        ;DRB001
 1511  007616' 000261                          SEC                       ; READY                      ;DRB001
 1512  007620' 000401                          BR      2$                ; RETURN                     ;DRB001
 1513  007622' 000241                  1$:     CLC                       ; NOT READY                  ;DRB001
 1514  007624' 000205                  2$:     RTS     R5                ;RETURN                      ;DRB001
 1515                                   ;       ------------------------
 1516
 1517
 1518  007626' 014167  170254          ERSUB2: MOV     -(R1),ASB         ; LOAD THE DATA
 1519  007632' 010167  170244                  MOV     R1,SHADR          ; LOAD ADDRESS OF DATA WRITTEN
 1520  007636' 014267  170246                  MOV     -(R2),AWAS        ; LOAD THE DATA
 1521  007642' 010267  170236                  MOV     R2,WASADR         ; LOAD ADDRESS OF DATA READ
 1522  007646' 005721                          TST     (R1)+             ; RESET REG. 1
 1523  007650' 005722                          TST     (R2)+             ; RESET REG. 2
 1524
 1525  007652' 016767  172442  170220  ERSUB1: MOV     RHCS1,CSRA        ; LOAD ADR OF CURRENT CSR
 1526  007660' 017767  172434  170214          MOV     @RHCS1,ACSR       ; LOAD CONTENTS OF CURRENT CSR
 1527  007666' 000207                          RTS     PC                ; RETURN
 1528
 1529
 1530  007670'                  REZET:
 1531  007670' 012777  000040  172432  3$:     MOV     #BIT5,@RHCS2      ; ISSUE AN RH11 INIT
 1532  007676' 012767  077777  171326          MOV     #77777,CLK        ; SET THE TIMER
 1533  007704' 105777  172410          1$:     TSTB    @RHCS1            ; CONTROLLER READY ?
 1534  007710' 100417                          BMI     2$                ; YES, CONTINUE
 1535  007712' 104407  000000'                BREAKS,BEGIN              ;TEMPORARY RETURN TO MONITOR....
```

```
1536  007716' 104407  000000'                 BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
1537  007722' 005367  171304                 DEC     CLK             ; WAIT SOME MORE ?
1538  007726' 001366                         BNE     1$              ; YES
1539  007730' 012767  000003  170150         MOV     #3,ERRTYP       ;CONTROLLER NOT READY
1540                                          ;*****************************************************************
1541  007736' 104405  000000' 000000         ERDER$,BEGIN,NULL       ; CONTROLLER NOT READY
1542                                          ;*****************************************************************
1543  007744' 104410  000000'                 ENDS,BEGIN              ;
1544  007750' 017746  172362         2$:     MOV     @#PAS,-(SP)
1545  007754' 012677  172356                 MOV     (SP)+,@#PAS
1546  007760' 000207                         RTS     PC              ; RETURN
1547                                  ;       -------------------------
1548
1549
1550  007762' 016700  170020         SETUP:  MOV     ADDR,R0         ; GET DEVICE ADDRESS
1551  007766' 010067  172326                 MOV     R0,RHCS1        ; GENERATE REGISTER ADDRESSES
1552  007772' 062700  000002                 ADD     #2,R0
1553  007776' 010067  172320                 MOV     R0,RHWC
1554  010002' 062700  000002                 ADD     #2,R0
1555  010006' 010067  172312                 MOV     R0,RHBA
1556  010012' 062700  000002                 ADD     #2,R0
1557  010016' 010067  172304                 MOV     R0,RPDA
1558  010022' 062700  000002                 ADD     #2,R0
1559  010026' 010067  172276                 MOV     R0,RHCS2
1560  010032' 062700  000002                 ADD     #2,R0
1561  010036' 010067  172270                 MOV     R0,RPDS
1562  010042' 062700  000002                 ADD     #2,R0
1563  010046' 010067  172262                 MOV     R0,RPER1
1564  010052' 062700  000002                 ADD     #2,R0
1565  010056' 010067  172254                 MOV     R0,RPAS
1566  010062' 062700  000002                 ADD     #2,R0
1567  010066' 010067  172246                 MOV     R0,RPLA
1568  010072' 062700  000002                 ADD     #2,R0
1569  010076' 010067  172240                 MOV     R0,RHDR
1570  010102' 062700  000002                 ADD     #2,R0
1571  010106' 010067  172232                 MOV     R0,RPMR
1572  010112' 062700  000002                 ADD     #2,R0
1573  010116' 010067  172224                 MOV     R0,RPDT
1574  010122' 062700  000002                 ADD     #2,R0
1575  010126' 010067  172216                 MOV     R0,RPSN
1576  010132' 062700  000002                 ADD     #2,R0
1577  010136' 010067  172210                 MOV     R0,RPOF
1578  010142' 062700  000002                 ADD     #2,R0
1579  010146' 010067  172202                 MOV     R0,RPDC
1580  010152' 062700  000002                 ADD     #2,R0
1581  010156' 010067  172174                 MOV     R0,RPCC
1582  010162' 062700  000002                 ADD     #2,R0
1583  010166' 010067  172166                 MOV     R0,RPER2
1584  010172' 062700  000002                 ADD     #2,R0
1585  010176' 010067  172160                 MOV     R0,RPER3
1586  010202' 062700  000002                 ADD     #2,R0
1587  010206' 010067  172152                 MOV     R0,RPEC1
1588  010212' 062700  000002                 ADD     #2,R0
1589  010216' 010067  172144                 MOV     R0,RPEC2
1590                                                                          ;**-7           ;DRB001
1591  010222' 032767  001000  167626         BIT     #ADDR22,RES1    ;DO WE HAVE 22-BIT ADDR SUPPORT?  ;DRB001
```



```
1592  010230' 001415                         BEQ     1$              ;NO                             ;DRB001
1593  010232' 032767  000200  167556         BIT     #RH70,SR1       ;IS THIS AN RH70                ;DRB001
1594  010240' 001411                         BEQ     1$              ;NO                             ;DRB001
1595  010242' 062700  000002                 ADD     #2,R0
1596  010246' 010067  172116                 MOV     R0,RHBAE
1597  010252' 062700  000002                 ADD     #2,R0
1598  010256' 010067  172110                 MOV     R0,RHCS3
1599
1600  010262' 000403                         BR      2$                                              ;DRB001
1601  010264' 042767  001000  167564  1$:    BIC     #ADDR22,RES1    ; CAN'T USE 22 BIT ADDRESSING    ;DRB001
1602  010272' 016700  167512          2$:    MOV     VECTOR,R0       ; GET VECTOR ADDRESS
1603  010276' 012720  003452'                 MOV     #NTRUPT,(R0)+   ; SET POINTER JUST IN CASE
1604  010302' 116710  167504                 MOVB    BR1,(R0)        ; SET PRIORITY
1605  010306' 000207                         RTS     PC              ; RETURN
1606
```

```
                                        .SBTTL  MODULE MESSAGES                                         ;DRB001
                                                                                ;**-40                  ;DRB001
010310' 042440  051122  051117  MES1:   .ASCIZ  ' ERROR%'                                               ;DRB001
010316' 000045
010320' 050040  051101  052111  MES2:   .ASCIZ  ' PARITY'                                              ;DRB001
010326' 000131
010330' 042040  044522  042526  MES3:   .ASCIZ  ' DRIVE '                                               ;DRB001
010336' 000040
010340' 046440  051501  041123  MES4:   .ASCIZ  ' MASSBUS'                                            ;DRB001
010346' 051525     000
010351'     040  040504  040524  MES5:   .ASCIZ  ' DATA'                                              ;DRB001
010356'     000
010357'     040  042522  051124  MES6:   .ASCIZ  ' RETRY EXCEEDED%'                                    ;DRB001
010364' 020131  054105  042503
010372' 042105  042105  000045
010400' 046040  052101  000105  MES7:   .ASCIZ  ' LATE'                                              ;DRB001
010406' 047516  020124  042522  MES8:   .ASCIZ  'NOT READY%'                                         ;DRB001
010414' 042101  022531     000
010421'     040  047503  046125  MES9:   .ASCIZ  ' COULD NOT GET'                                     ;DRB001
010426' 020104  047516  020124
010434' 042507  000124
010440' 042040  047522  050120  MES10:  .ASCIZ  ' DROPPED'                                           ;DRB001
010446' 042105     000
010451'     072  047040  052117  MES11:  .ASCIZ  ': NOT AN RP04/5/6 OR AN RM02/3/5%'                  ;DRB001;WMRMRP
010456' 040440  020116  050122
010464' 032060  032457  033057
010472' 047440  020122  047101
010500' 051040  030115  027462
010506' 027463  022465     000
010513'     040  051124  047101  MES12:  .ASCIZ  ' TRANSFER'                                          ;DRB001
010520' 043123  051105     000
010525'     045     000         MES13:  .ASCIZ  '%'                                                 ;DRB001
010527'     072  041040  042101  MES15:  .ASCIZ  ': BADSPOT AT (OCTAL) CYL: '                         ;DRB001
010534' 050123  052117  040440
010542' 020124  047450  052103
010550' 046101  020051  054503
010556' 035114  000040
010562' 000002              CYLNO:  .BLKB   2                                                       ;DRB001
010564' 020040  020040  020054  MES16:  .ASCIZ  '    , TRK: '                                       ;DRB001
010572' 051124  035113  000040
010600' 000004              TRKNO:  .BLKB   4                                                       ;DRB001
010604' 020040  020054  042523  MES17:  .ASCIZ  '  , SEC: '                                         ;DRB001
010612' 035103  000040
010616' 000004              SECNO:  .BLKB   4                                                       ;DRB001
010622' 020040  000045      MES18:  .ASCIZ  '  %'                                                  ;DRB001
010626' 052440  045516  047516  MES19:  .ASCIZ  ' UNKNOWN'                                         ;DRB001
010634' 047127     000


                                        ;////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\
                                        ;////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\////\\\\


                                        ;REV D  UPDATES         ;MCP
```



```
010637'     110  051101  020104  MES20:  .ASCIZ  'HARD ERROR(S) SET IN RPER2  %'
010644' 051105  047522  024122
010652' 024523  051440  052105
010660' 044440  020116  050122
010666' 051105  020062  022440
010674'     000
010675'     110  051101  020104  MES21:  .ASCIZ  'HARD ERROR(S) SET IN RPER3  % '
010702' 051105  047522  024122
010710' 024523  051440  052105
010716' 044440  020116  050122
010724' 051105  020063  022440
010732' 000040
010734' 051127  052111  020105  MES22:  .ASCIZ  'WRITE CHECK ERROR  % '
010742' 044103  041505  020113
010750' 051105  047522  020122
010756' 022440  000040
010762' 041523  051440  052105  MES23:  .ASCIZ  'SC SET  %'
010770' 020040  000045
010774' 044124  020105  047506  MES24:  .ASCIZ  'THE FOLLOWING ERROR BITS ARE SET IN RPCS2 % % '
011002' 046114  053517  047111
011010' 020107  051105  047522
011016' 020122  044502  051524
011024' 040440  042522  051440
011032' 052105  044440  020116
011040' 050122  051503  020062
011046' 020045  020045     000
011053'     125  044516  052502  MES25:  .ASCIZ  'UNIBUS PARITY ERROR % '
011060' 020123  040520  044522
011066' 054524  042440  051122
011074' 051117  022440  000040
011102' 047516  020116  054105  MES26:  .ASCIZ  'NON EXISTANT DRIVE % '
011110' 051511  040524  052116
011116' 042040  044522  042526
011124' 022440  000040
011130' 054116  020115  022440  MES27:  .ASCIZ  'NXM  %'
011136'     000
011137'     120  047522  051107  MES28:  .ASCIZ  'PROGRAM ERROR  PGE  % '
011144' 046501  042440  051122
011152' 051117  020040  043520
011160' 020105  022440  000040
011166' 044515  051523  042105  MES29:  .ASCIZ  'MISSED TRANSFER  %  '
011174' 052040  040522  051516
011202' 042506  020122  022440
011210' 020040     000
011213'     115  051501  041123  MES30:  .ASCIZ  'MASSBUS DATA LINE PARITY ERROR  % '
011220' 051525  042040  052101
011226' 020101  044514  042516
011234' 050040  051101  052111
011242' 020131  051105  047522
011250' 020122  022440  000040
011256' 044124  020105  047506  MES31:  .ASCIZ  'THE FOLLOWING BITS ARE SET IN RPER1  % % '
011264' 046114  053517  047111
011272' 020107  044502  051524
011300' 040440  042522  051440
011306' 052105  044440  020116
011314' 050122  051105  020061
```

```
 1719  011322' 022440  022440  000040
 1720  011330' 040504  040524  041440  MES32:  .ASCIZ  'DATA CHECK ERROR  % '
 1721  011336' 042510  045503  042440
 1722  011344' 051122  051117  020040
 1723  011352' 020045     000
 1724  011355'    106  051117  040515  MES33:  .ASCIZ  'FORMAT ERROR  %  '
 1725  011362' 020124  051105  047522
 1726  011370' 020122  022440  020040
 1727  011376'    000
 1728  011377'    040  042510  042101  MES34:  .ASCIZ  ' HEADER CRC ERROR % '
 1729  011404' 051105  041440  041522
 1730  011412' 042440  051122  051117
 1731  011420' 022440  000040
 1732  011424' 042510  042101  051105  MES35:  .ASCIZ  'HEADER COMPARE ERROR  % '
 1733  011432' 041440  046517  040520
 1734  011440' 042522  042440  051122
 1735  011446' 051117  020040  020045
 1736  011454'    000
 1737  011455'    104  044522  042526  MES36:  .ASCIZ  'DRIVE UNSAFE  % '
 1738  011462' 052440  051516  043101
 1739  011470' 020105  022440  000040
 1740  011476' 050117  051105  052101  MES37:  .ASCIZ  'OPERATION INCOMPLETE % '
 1741  011504' 047511  020116  047111
 1742  011512' 047503  050115  042514
 1743  011520' 042524  022440  000040
 1744  011526' 051104  053111  020105  MES38:  .ASCIZ  'DRIVE TIMING ERROR % '
 1745  011534' 044524  044515  043516
 1746  011542' 042440  051122  051117
 1747  011550' 022440  000040
 1748  011554' 051127  052111  020105  MES39:  .ASCIZ  'WRITE LOCK ERROR % '
 1749  011562' 047514  045503  042440
 1750  011570' 051122  051117  022440
 1751  011576' 000040
 1752  011600' 047111  040526  044514  MES40:  .ASCIZ  'INVALID ADDRESS ERROR % '
 1753  011606' 020104  042101  051104
 1754  011614' 051505  020123  051105
 1755  011622' 047522  020122  020045
 1756  011630'    000
 1757  011631'    105  041503  044040  MES41:  .ASCIZ  'ECC HARD ERROR % '
 1758  011636' 051101  020104  051105
 1759  011644' 047522  020122  020045
 1760  011652'    000
 1761  011653'    120  051101  052111  MES42:  .ASCIZ  'PARITY ERROR % '
 1762  011660' 020131  051105  047522
 1763  011666' 020122  020045     000
 1764  011673'    122  043505  051511  MES43:  .ASCIZ  'REGISTER MODIFICATION REFUSED  % '
 1765  011700' 042524  020122  047515
 1766  011706' 044504  044506  040503
 1767  011714' 044524  047117  051040
 1768  011722' 043105  051525  042105
 1769  011730' 020040  020045     000
 1770  011735'    111  046114  043505  MES44:  .ASCIZ  'ILLEGAL REGISTER % '
 1771  011742' 046101  051040  043505
 1772  011750' 051511  042524  020122
 1773  011756' 020045     000
 1774  011761'    111  046114  043505  MES45:  .ASCIZ  'ILLEGAL FUNCTION % '
```



```
 1775  011766' 046101  043040  047125
 1776  011774' 052103  047511  020116
 1777  012002' 020045     000
 1778  012005'    040  022445  020045  MES46:  .ASCIZ  ' %%%   ********************************** %% '
 1779  012012' 020040  025052  025052
 1780  012020' 025052  025052  025052
 1781  012026' 025052  025052  025052
 1782  012034' 025052  025052  025052
 1783  012042' 025052  025052  025052
 1784  012050' 025052  025052  025052
 1785  012056' 025052  022440  020045
 1786  012064'    000
 1787  012065'    101  042104  042522  MES47:  .ASCIZ  'ADDRESS OVERFLOW ERROR % '
 1788  012072' 051523  047440  042526
 1789  012100' 043122  047514  020127
 1790  012106' 051105  047522  020122
 1791  012114' 020045     000
 1792  012117'    127  044522  042524  MES48:  .ASCIZ  'WRITE LOCK FAILURE % '
 1793  012124' 046040  041517  020113
 1794  012132' 040506  046111  051125
 1795  012140' 020105  020045     000
 1796  012145'    040  020055  047514  MES49:  .ASCIZ  ' - LOAD MEDIA % '  ;RMDB1
 1797  012152' 042101  046440  042105
 1798  012160' 040511  022440  000040
 1799
 1800
 1801                                        ; R E V  D   END OF  UPDATE
 1802
 1803                                        ;/////\\\\\/////\\\\\/////\\\\\/////\\\\\/////\\\\\
 1804                                        ;/////\\\\\/////\\\\\/////\\\\\/////\\\\\/////\\\\\
 1805
 1806
 1807                                .EVEN                                                   ;DRB001
 1808                                                                                        ;DRB001
 1809  012166' 010513'               TRERR:  MES12                                           ;DRB001
 1810  012170' 010310'                       MES1                                            ;DRB001
 1811  012172' 177777                       177777                                          ;DRB001
 1812                                                                                        ;DRB001
 1813  012174' 010340'               MCPERR: MES4                                            ;DRB001
 1814  012176' 010320'                       MES2                                            ;DRB001
 1815  012200' 010310'                       MES1                                            ;DRB001
 1816  012202' 177777                       177777                                          ;DRB001
 1817                                                                                        ;DRB001
 1818  012204' 010340'               MDPERR: MES4                                            ;DRB001
 1819  012206' 010351'                       MES5                                            ;DRB001
 1820  012210' 010320'                       MES2                                            ;DRB001
 1821  012212' 010310'                       MES1                                            ;DRB001
 1822  012214' 177777                       177777                                          ;DRB001
 1823                                                                                        ;DRB001
 1824  012216' 010357'               EXCED:  MES6                                            ;DRB001
 1825  012220' 177777                       177777                                          ;DRB001
 1826                                                                                        ;DRB001
 1827  012222' 010351'               DLTERR: MES5                                            ;DRB001
 1828  012224' 010400'                       MES7                                            ;DRB001
 1829  012226' 010310'                       MES1                                            ;DRB001
 1830  012230' 177777                       177777                                          ;DRB001
```

```
 1831
 1832  012232' 010330'                BADMES: MES3                                    ;DRB001
 1833  012234' 012523'                        NUMB                                    ;DRB001
 1834  012236' 010527'                        MES15                                   ;DRB001
 1835  012240' 010564'                        MES16                                   ;DRB001
 1836  012242' 010604'                        MES17                                   ;DRB001
 1837  012244' 010622'                        MES18                                   ;DRB001
 1838  012246' 177777                         177777                                  ;DRB001
 1839                                                                                  ;DRB001
 1840  012250' 010626'                UNKNWN: MES19                                   ;DRB001
 1841  012252' 010310'                        MES1                                    ;DRB001
 1842  012254' 177777                         177777                                  ;DRB001
 1843                                                                                  ;DRB001
 1844  012256' 010330'                NOT:    MES3                                    ;DRB001
 1845  012260' 010406'                        MES8                                    ;DRB001
 1846  012262' 177777                         177777                                  ;DRB001
 1847                                                                                  ;DRB001
 1848  012264' 010421'                TOUT:   MES9                                    ;DRB001
 1849  012266' 010330'                        MES3                                    ;DRB001
 1850  012270' 010525'                        MES13                                   ;DRB001
 1851  012272' 177777                         177777                                  ;DRB001
 1852                                                                                  ;DRB001
 1853  012274' 010330'                DRP:    MES3                                    ;DRB001
 1854  012276' 012523'                        NUMB                                    ;DRB001
 1855  012300' 010440'                        MES10                                   ;DRB001
 1856  012302' 010525'                        MES13                                   ;DRB001
 1857  012304' 177777                         177777                                  ;DRB001
 1858                                                                                  ;DRB001
 1859  012306' 010330'                BADTYP: MES3                                    ;DRB001
 1860  012310' 012523'                        NUMB                                    ;DRB001
 1861  012312' 010440'                        MES10                                   ;DRB001
 1862  012314' 010451'                        MES11                                   ;DRB001
 1863  012316' 177777                         177777                                  ;DRB001
 1864
 1865
 1866                                         ;/////\\\\\/////\\\\\/////\\\\\/////\\\\\
 1867                                         ;/////\\\\\/////\\\\\/////\\\\\/////\\\\\
 1868
 1869                                         ;R E V  D  UPDATES
 1870
 1871  012320' 010637'                ER2ERR: MES20
 1872  012322' 177777                         177777
 1873
 1874  012324' 010675'                ER3ERR: MES21
 1875  012326' 177777                         177777
 1876
 1877  012330' 010734'                WCEERR: MES22
 1878  012332' 177777                         177777
 1879
 1880  012334' 010762'                SCERR:  MES23
 1881  012336' 177777                         177777
 1882
 1883  012340' 010774'                CS2HD:  MES24
 1884  012342' 177777                         177777
 1885
 1886  012344' 011053'                UPER:   MES25
```



```
 1887  012346' 177777                         177777
 1888
 1889  012350' 011102'                NEDER:  MES26
 1890  012352' 177777                         177777
 1891
 1892  012354' 011130'                NXM:    MES27
 1893  012356' 177777                         177777
 1894
 1895  012360' 011137'                PGERR:  MES28
 1896  012362' 177777                         177777
 1897
 1898  012364' 011166'                MXFER:  MES29
 1899  012366' 177777                         177777
 1900
 1901  012370' 011213'                MDPER:  MES30
 1902  012372' 177777                         177777
 1903
 1904  012374' 011256'                ER1HD:  MES31
 1905  012376' 177777                         177777
 1906
 1907  012400' 011330'                DCKER:  MES32
 1908  012402' 177777                         177777
 1909
 1910  012404' 011355'                FERER:  MES33
 1911  012406' 177777                         177777
 1912
 1913  012410' 011377'                HCRC:   MES34
 1914  012412' 177777                         177777
 1915
 1916  012414' 011424'                HCE:    MES35
 1917  012416' 177777                         177777
 1918
 1919  012420' 011455'                UNSER:  MES36
 1920  012422' 177777                         177777
 1921
 1922  012424' 011476'                UPIER:  MES37
 1923  012426' 177777                         177777
 1924
 1925  012430' 011526'                DTEF:   MES38
 1926  012432' 177777                         177777
 1927  012434' 011554'                WLER:   MES39
 1928  012436' 177777                         177777
 1929  012440' 011600'                IAER:   MES40
 1930  012442' 177777                         177777
 1931
 1932  012444' 011631'                ECHR:   MES41
 1933  012446' 177777                         177777
 1934
 1935  012450' 011653'                PARR:   MES42
 1936  012452' 177777                         177777
 1937
 1938  012454' 011673'                RMRR:   MES43
 1939  012456' 177777                         177777
 1940
 1941  012460' 011735'                ILRR:   MES44
 1942  012462' 177777                         177777
```

```
                                ILFR:   MES45
012464' 011761'                         177777
012466' 177777
                                ASTER:  MES46
012470' 012005'                         177777
012472' 177777

                                AOE:    MES47                   ;       REV. D0
012474' 012065'                         177777
012476' 177777

                                WCF:    MES48                   ;       REV. D0
012500' 012117'                         177777
012502' 177777

                                LODMED: MES3                    ; LOAD MEDIA    ;RMDB
012504' 010330'                         NUMB
012506' 012523'                         MES10
012510' 010440'                         MES49
012512' 012145'                         177777
012514' 177777


                                ;R E V  D  END UPDATE

                                ;////\\\\////\\\\////\\\\////\\\\////\\\\
                                ;////\\\\////\\\\////\\\\////\\\\////\\\\


012516' 000005                  ADR1:   .BLKB   5                                       ;DRB001
012523'    040      000         NUMB:   .ASCIZ  / /                                     ;DRB001
           012526'                         .EVEN                                       ;DRB001
                                                                                       ;DRB001
                                ;-----------------------------                         ;DRB001
                                                                                       ;DRB001
012526' 000024                  PATCH:
012526' 000000                          .WORD   0
012530' 000000                          .WORD   0
012532' 000000                          .WORD   0
012534' 000000                          .WORD   0
012536' 000000                          .WORD   0
012540' 000000                          .WORD   0
012542' 000000                          .WORD   0
012544' 000000                          .WORD   0
012546' 000000                          .WORD   0
012550' 000000                          .WORD   0
012552' 000000                          .WORD   0
012554' 000000                          .WORD   0
012556' 000000                          .WORD   0
012560' 000000                          .WORD   0
012562' 000000                          .WORD   0
012564' 000000                          .WORD   0
012566' 000000                          .WORD   0
012570' 000000                          .WORD   0
012572' 000000                          .WORD   0
012574' 000000                          .WORD   0
000001                          .END                                                   ;DRB001
```



```
ACSR    000102R          273#   1526*
ADDR    000006R          239#   1550
ADDR22= 001000           302#    767   1591    1601
ADR1    012516R         1165    1411   1969#
AOE     012474R         1337    1949#
APTPRE= 000200           302#
ASR     000106R          277#   1518*
ASTAT   000104R          275#
ASTER   012470R         1184    1257   1946#
AUTO  = 000010           302#
AWAS    000110R          278#   1520*
BAD     007160R          809    1344   1380    1401#
BADCYL  001170R          555#   1405*  1417    1437
BADMES  012232R         1432    1832#
BADSEC  001172R          556#   1403*  1429    1438
BADSPT  000252R          305#    528    633     939
BADTRK  001174R          557#   1404*  1423    1439
BADTYP  012306R         1115    1859#
BDSPT = 000100           506#   1406
BEGIN   000000R          236#    655    669     672     684     686     696     747     748     774     782     791     808
                         862    1115   1165    1184    1193    1197    1211    1216    1220    1224    1228    1232    1245
                        1250    1257   1265    1268    1271    1274    1277    1280    1284    1288    1292    1295    1306
                        1309    1322   1325    1328    1331    1334    1337    1340    1343    1347    1350    1353    1356
                        1411    1417   1423    1429    1432    1457    1458    1459    1460    1463    1477    1482    1535
                        1536    1541   1543
BITAB   003670R           785     828    840#
BIT0  = 000001           302#    915   1028    1034    1056    1064    1081    1091    1098    1108    1354
BIT1  = 000002           302#   1351
BIT10 = 002000           302#   1275   1332
BIT11 = 004000           302#   1272   1329    1450    1454    1491    1500    1508
BIT12 = 010000           302#    763   1269    1326    1502
BIT13 = 020000           302#   1194   1266    1323
BIT14 = 040000           302#    762   1201    1221    1225    1320    1504
BIT15 = 100000           302#   1186   1229    1289    1298
BIT2  = 000004           302#    503   1348
BIT3  = 000010           302#   1345
BIT4  = 000020           302#    504   1293    1298
BIT5  = 000040           302#    505   1338    1531
BIT6  = 000100           302#    506   1341    1496
BIT7  = 000200           302#    507   1307
BIT8  = 000400           302#   1282   1303    1498
BIT9  = 001000           302#   1276   1335
BPORT = 000020           504#    913
BREAK$= 104407           302#    747    748    1457    1458    1459    1460    1535    1536
BR1     000012R          241#   1604
BR2     000013R          242#
BTODS = 104421           302#
CAPRES= 000004           302#
CDATA$= 104412           302#    696
CDERCT  000144R          292#
CDMDCT  000146R          293#
CHKCS2  006204R         1200    1264#
CHKER1  006346R         1205    1287#
CKHNG$= 000001           302#
CLEAR   003240R          752#   1465
CLK     001232R          575#   1448*  1461*   1532*   1537*
```

```
CLKPRE= 000001          302#
CLKSPS= 104422          302#
CLRCOM  005444R        1150#   1152
CLRRB   005434R        1148#
CONFIG  000056R         261#
CSRA    000100R         271#   1525#
CS2HD   012340R        1265    1883#
CYCLE   002670R         679     690#
CYL     001164R         550#    657#   865#   892#   898    902#   906    909#   912    924#
CYLLIM  001140R         530#    898    907    917   1029#  1035#  1057#  1065#  1082#  1092#  1099#  1109#
CYLMSK  001154R         543#    864   1038#  1068#  1123#
CYLNDR  001162R         549#    722    732    742    912#   917#   919    921#   943    959   1405
CYLNO   010562R        1417    1644#
DATCKS= 104411          302#
DATERS= 104404          302#
DCRER   012400R        1292    1907#
DLATE = 000004          503#   1190
DLTCNT  001220R         570#   1188#
DLTERR  012222R        1193    1827#
DLTFLG  006202R        1183#   1189#  1223#  1227#  1231#  1239   1241   1261#  1291#  1301#  1305#
DRIVE   001214R         567#
DROP    005454R         653    1114   1155#  1481
DRP     012274R        1482    1853#
DSKADR  001166R         551#    658#   721    731    741    903#   925#
DTER    012430R        1328    1925#
DVICE   001212R         566#    638#   650    682    828   1161#
DVID1   000014R         243#    638
EA22    001134R         526#    776    777#   778#   779
ECCMEM= 000100          302#
ECHR    012444R        1343    1932#
ENDBBK  001052R         501#   1434
END1TS= 104413          302#    686
ENDS  = 104410          302#    684   1543
ERDR    005512R        1183#   1399
ERRORS  007050R         794    1367#
ERRTYP  000106R         276#   1475#  1539#
ERSUB1  007652R        1395    1468   1525#
ERSUB2  007626R        1518#
ER1HD   012374R        1288    1904#
ER2ERR  012320R        1211    1216   1871#
ER3ERR  012324R        1220    1874#
EXCED   012216R         808    1824#
EXITS = 104400          302#    782
FERADR  001230R         574#    716#   726#   736#   807
FERER   012404R        1295    1910#
FLAG    001210R         565#    915   1028#  1034#  1056#  1064#  1081#  1091#  1098#  1108#
FREE    000150R         294#
FUNC    001222R         571#    715#   725#   735#   780#   781
GETDVT  004552R         836    1016#
GETPAS= 104415          302#    669
GO      003324R         724     734    744    766#
GWBUFS= 104414          302#    672
HCE     012414R        1309    1916#
HCRC    012410R        1306    1913#
HRDCNT  000044R         256#
HRDCS2= 137400          508#   1198
```



```
HRDERR  001240R         578#    799   1185#  1192#  1196#  1210#  1215#  1219#  1237   1264#  1300#  1314#  1319#
HRDERS= 104405          302#   1250   1477   1541
HRDER1= 177777          509#   1203
HRDER2= 167777          510#   1213
HRDER3= 140153          511#   1217
HRDPAS  000050R         258#
IAER    012440R        1334    1929#
ICONT   000036R         253#
ICOUNT  000040R         254#
IDNUM   000122R         283#
ILER    012464R        1356    1944#
ILRR    012460R        1353    1941#
IMODX.= 000000          295#    673
INDPAR= 000040          302#
INIT    000030R         250#
INTR    000120R         282#    632#
KTPRES= 000400          302#
KTXINT= 040000          302#
LASCYL  001176R         559#    919    921   1011#  1030#  1036#  1058#  1066#  1083#  1093#  1100#  1110#
LASSEC  001202R         561#    930    932   1014#
LASTRK  001200R         560#    927    929   1013#
LODMED  012504R         655    1955#
LOOP1   002612R         671#    688
LOOP2   002624R         674#    680    814   1382   1481
MAP22S= 104416          302#    774
MCPERR  012174R        1197    1613#
MDPER   012370R        1284    1901#
MDPERP  012204R        1818#
MES1    010310R        1609#   1810   1815   1821   1829   1841
MES10   010440R        1628#   1855   1861   1957
MES11   010451R        1630#   1862
MES12   010513R        1636#   1809
MES13   010525R        1638#   1850   1856
MES15   010527R        1639#   1834
MES16   010564R        1645#   1835
MES17   010604R        1648#   1836
MES18   010622R        1651#   1837
MES19   010626R        1652#   1840
MES2    010320R        1611#   1814   1820
MES20   010637R        1663#   1871
MES21   010675R        1669#   1874
MES22   010734R        1675#   1877
MES23   010762R        1679#   1880
MES24   010774R        1681#   1883
MES25   011053R        1689#   1886
MES26   011102R        1693#   1889
MES27   011130R        1697#   1892
MES28   011137R        1699#   1895
MES29   011166R        1703#   1898
MES3    010330R        1613#   1832   1844   1849   1853   1859   1955
MES30   011213R        1707#   1901
MES31   011256R        1713#   1904
MES32   011330R        1720#   1907
MES33   011355R        1724#   1910
MES34   011377R        1728#   1913
MES35   011424R        1732#   1916
```

```
MES36   011455R          1737#   1919
MES37   011476R          1740#   1922
MES38   011526R          1744#   1925
MES39   011554R          1748#   1927
MES4    010340R          1615#   1813    1818
MES40   011600R          1752#   1929
MES41   011631R          1757#   1932
MES42   011653R          1761#   1935
MES43   011673R          1764#   1938
MES44   011735R          1770#   1941
MES45   011761R          1774#   1944
MES46   012005R          1778#   1946
MES47   012065R          1787#   1949
MES48   012117R          1792#   1952
MES49   012145R          1796#   1958
MES5    010351R          1617#   1819    1827
MES6    010357R          1619#   1824
MES7    010400R          1622#   1828
MES8    010406R          1623#   1845
MES9    010421R          1625#   1848
MODLUS  005422R           886     895     908    1137#
MODNAM  000000R           237#
MODSP   000252R           251     300#
MSGNS = 104403            302#    655     808    1115   1184   1193   1197   1411   1216   1220   1224   1228   1232
                         1257    1265    1268    1271   1274   1277   1280   1284   1288   1292   1295   1306   1309
                         1322    1325    1328    1331   1334   1337   1340   1343   1347   1350   1353   1356   1432
                         1463    1482
MSGSS = 104402            302#
MSGS  = 104401            302#
MXFER   012364R          1280    1898#
NCPUOP= 000020            302#
NEDER   012350R          1271    1889#
NUAPTY= 000002            302#
NORAND= 000040            505#    855     900    922
NOT     012256R          1463    1844#
NOTRDY  007316R           806     835    1255   1448#
NTRUPT  003452R           783#    1603
NULL  = 000000            302#    1541
NUMH    012523R          1833    1854    1860   1956   1970#
NXM     012354R          1274    1892#
NXTBBK  001136R           528#    637#    940    1434   1436   1440#
OPEN  = 000000            238     244     245    246    247    264    265    266    267    268    269    270    271
                          273     275     277    278    280    281    282    285    286    288    289    291    292
                          293     294     302#
OPIER   012424R          1325    1922#
OTOAS = 104420            302#    1165    1411   1417   1423   1429
PARPRE= 002000            302#
PARR    012450R          1347    1935#
PASCNT  000034R           252#
PATCH   012526R          1975#
PA1H    001126R           523#    769#    774
PA22    001132R           525#    775
PDPF11= 000002            302#
PDPLSI= 020000            302#
PDP44 = 100000            302#
PDP60 = 004000            302#
```



```
PDP70 = 010000            302#
PGERR   012360R          1277    1895#
PICKHK  003700R           692     812     846#   971
PICKOR  003606R           675     821#
PIRQS = 000004            302#    791
PUPSP = 005726            302#
POPSP2= 022626            302#
PRHMSS= 000002            302#
PRTY  = 000000            302#
PRTY0 = 000000            242     302#
PRTY1 = 000040            302#
PRTY2 = 000100            302#
PRTY3 = 000140            302#
PRTY4 = 000200            302#
PRTY5 = 000240            241     302#
PRTY6 = 000300            302#
PRTY7 = 000340            302#
PS    = 177776            302#
PSW   = 177776            302#
PUSH  = 005746            302#
PUSH2 = 024646            302#
PWRFLG= 000002            302#
QMON22= 000010            302#
RANDS = 104417            302#    862
RANNUM  000054R           260#    863     866    870
RBUF    001242R           284     579#   1148
RBUFEA  000130R           286#    744
RBUFPA  000126R           285#    696     740
RBUFSZ  000132R           287#    737    1149
RBUFVA  000124R           284#    669
READ    003116R           695     735#    736
READY   007514R           804     833    1466   1489#
RELESE  003212R           698     745#   1480
RESTRT  002604R           279     665     667#
RES1    000056R           262#    767    1591   1601#
RES2    000060R           263#
RESET   007670R           661    1530#
RHBA    002324R           608#    720*    730*   740*   769    775*   1555*
RHBAE   002370R           626#    776*   1596*
RHCS1   002320R           606#    746*    753*   756*   757    762*   781*   1194   1368   1387   1450   1454   1470
                         1490    1508    1525   1526   1533   1551*
RHCS2   002330R           610#    745*    752*   766*   1021*  1489*  1531*  1559*
RHCS3   002372R           627#   1598*
RHDB    002342R           615#   1235    1236*  1252   1254*  1569*
RHWC    002322R           607#    719*    729*   739*   1553*
RH70  = 000200            302#    507#    662    1593
RMRR    012454R          1350   1938#
RPAS    002336R           613#    760     761*   786    788*   1544   1545*  1565*
RPCC    002356R           621#   1581*
RPDA    002326R           609#    721*    731*   741*   1557*
RPDC    002354R           620#    722*    732*   742*   1579*
RPDS    002332R           611#   1493    1561*
RPDT    002346R           617#   1026    1032   1052   1054   1060   1062   1079   1086   1088   1096   1103   1105
                         1573#
RPEC1   002364R           624#   1587*
RPEC2   002366R           625#   1589*
```

```
RPER1   002334R          612#   1563*
RPER2   002360R          622#   1583*
RPER3   002362R          623#   1376   1585*
RPLA    002340R          614#   1567*
RPMR    002344R          616#   1571*
RPOF    002352R          619#    763*  1577*
RPSN    002350R          618#   1575*
RSTR1   000112R          279#
S       001052R          513#    582    583    584    585    586    587    588    589    590    591    592    593
                         594     595    596    597    598    599    600    601    602    603    666   1186   1198
                        1201    1203   1208   1213   1217   1221   1225   1229   1233   1266   1269   1272   1275
                        1278    1282   1289   1293   1296   1303   1307   1310   1315   1320   1323   1326   1329
                        1332    1335   1338   1341   1345   1348   1351   1354   1390*  1471*
SBADR   000102R          272#   1519*
SCERR   012334R         1232    1880#
SECLIM  001150R          541#    861    946   1000   1043*  1073*  1128*
SECLM1  001152R          542#    885    993    996   1004   1044*  1074*  1129*
SECMSK  001160R          545#    876   1040*  1070*  1125*
SECNO   010616R         1429    1650#
SECTOR  001166R          552#    857*   869*   881    884    887*   930    932*   941    967   1403
SETUP   007762R          660    1550#
SIZSEC  001206R          564#    945    998*
SIZTRK  001204R          563#    951    992*   995*  1007
SOFCNT  000042R          255#
SOFERR= 104406           302#   1245
SOFPAS  000046R          257#
SPOINT  000032R          251#   1483
SPSIZ = 000040             1#    295
SR1     000016R          244#    662    855    900    913    922   1190   1406   1593
SR2     000020R          245#
SR3     000022R          246#
SR4     000024R          247#
START   002374R          250     630#
STAT    000026R          249#
STOPE   001236R          577#   1235*  1254   1296*  1297*  1298   1310*  1311*  1312
SVR0    0000b2R          264#
SVR1    000064R          265#
SVR2    000066R          266#
SVR3    000070R          267#
SVR4    000072R          268#
SVR5    000074R          269#
SVR6    000076R          270#
SYSCNT  000052R          259#
TABLE   002242R          581#    664*   666*  1245   1250   1392   1473   1477
TIMER   001224R          572#
TOUT    012264R         1848#
TRACK   001167R          553#    879*   883*   890    893    896*   927    929*   942    963   1404
TRERR   012166R         1228    1809#
TRKLIM  001144R          539#    890    952    999   1041*  1071*  1126*
TRKLM1  001146R          540#    894   1010   1042*  1072*  1127*
TRKMSK  001156R          544#    868   1039*  1069*  1124*
TRKNO   010600R         1423    1647#
TRPDFD= 000023           302#
TRY     001234R          576#    678*   801*   802
TYPFLG  001142R          536#   1037*  1067*  1122*  1206   1374
UNITNO  001216R          568#    652*   671*   745    752    766    784    813*   822*   823   827   1021   1156
```



```
                        1165    1411   1489
UNKNWN  012250R         1840#
UNSER   012420R         1322    1919#
UPER    012344R         1268    1886#
USTACK= 000001           302#
VECTOR  000010R          240#   1602
WASADR  000104R          274#   1521#
WBUFEA  000136R          289#    724    734
WBUFPA  000134R          288#    720    730
WBUFRQ  000140R          290#
WBUFSZ  000142R          291#    717    727    989
WCEERR  012330R         1224    1877#
WCF     012500R         1340    1952#
WDER    000116R          281#    631*
WDTO    000114R          280#    630*
WLER    012434R         1331    1927#
WRITCK  003022R          694     725#   726
WRITE   002726R          693     715#   716
WRTLIM  004432R          677     979#
XFLAG   000005R          238#
XMEM    001130R          524#    724*   734*   744*   770*   771*   772*   773*   779*   780
ZERO    001226R          573#
.     = 012576R          579#    697   1644#  1647#  1650#  1969#  1971#
```

```
BKMOD       1#
BREAK       1#     747    1456    1459    1535
BTOD        1#
CKDATA      1#     696
CLKSP       1#
DATACK      1#
DATERR      1#
DFSEVN      1#     302
DSEVNT      1#     302
END         1#     684    1543
ENDIT       1#     685
ENDMOD      1#
EQUATS      1#     302
EXIT        1#     782
GETPA       1#     669
GWBUFF      1#     672
HRDER       1#    1248    1476    1540
IOMOD       1#
IOMODP      1#
IOMODR      1#
IOMODX      1#     230
LINEUP    702#     723     733     743
MAP22       1#     774
MODULE      1#     231
MSG         1#
MSGN        1#     655    1115    1184    1193    1197    1211    1216    1220    1224    1228    1232    1257    1265    1268
          1271    1274    1277    1280    1284    1287    1292    1295    1306    1309    1322    1325    1328    1331    1334
          1337    1340    1343    1347    1350    1353    1356    1432    1463    1482
MSGS        1#
NBKMOD      1#
OTOA        1#    1162    1408    1414    1420    1426
PIRQ        1#     790
RAND        1#     862
SBKMOD      1#
SOFER       1#    1244


. ABS.  000000     000
        012576     001

ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

XRMDB0,XRMDB0.SEQ/SOL/CRF/DOC=DDXCOM,XRMDB0
RUN-TIME:  2  5  .8 SECONDS
RUN-TIME RATIO: 38/9=4.1
CORE USED:  11K  (22 PAGES)

DOCUMENT PAGES:  47
```

```
.REM    %
```

IDENTIFICATION

PRODUCT CODE: AC-S077A-MC

PRODUCT NAME: CXRMCA0 RM05/3/2 MODULE

DATE CREATED: APRIL 1980

MAINTAINER: CX DIAGNOSTIC GROUP

AUTHOR: MIKE LEAVITT

1. ABSTRACT

CXRMCA0 IS AN IOMODX THAT EXERCISES RM05, RM03 AND RM02 DISK DRIVES ON AN RH70 OR RH11 CONTROLLER. IT EXERCISES THE DRIVES BY DOING SEEKS, WRITES, WRITE-CHECKS, READS, AND IN-CORE COMPARISONS. ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE TTY.

2. REQUIREMENTS

PDP-11 PROCESSOR
PROGRAM LOADING DEVICE
TERMINAL
RH11 OR RH70 CONTROLLER
1 TO 8 DISK DRIVES (ANY COMBINATION OF RM05'S, RM03'S OR RM02'S)

TEST MEDIA:

THE RM05, RM03 OR RM02 DISK PACK MUST BE FORMATTED IN 16 BIT DATA FORMAT.

LOAD DEVICE:

ANY DEVICE THAT IS SUPPORTED BY THE XXDP SOFTWARE PACKAGE.

3. PASS DEFINITION

ONE PASS OF THE CXRMCA0 MODULE CONSISTS OF 200 CYCLES OF THE FOLLOWING TEST SEQUENCE FROM STEP 1 TO STEP 9.

1. SELECT A STARTING BLOCK ADDRESS. (SEE SECTION 10.)
2. SELECT A DRIVE FOR TEST.
3. IF NO DRIVE IS SELECTED, DROP THE CXRMCA0 MODULE.
4. DO AN EXPLICIT SEEK TO A CYLINDER. IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT.
5. DO A WRITE DATA - IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT.
6. DO A WRITE CHECK DATA - IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT.
7. DO A READ DATA - IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT.
8. DO AN IN-CORE DATA COMPARE - IF ERRORS, REPORT.
9. IF COMPLETED CYCLE OF ALL DRIVES, GO TO STEP 1, ELSE STEP 2.

NOTE: THE RETRY LIMIT IS 3 TIMES. THE MODULE CLEARS THE ERROR, BEFORE EACH RETRY. IF THE MODULE FAILS TO CLEAR THE ERROR, THE MODULE WILL BE DROPPED.

4. EXECUTION TIME

```
58
59                      ONE PASS OF CXRMCA0 RUNNING ALONE ON A PDP-11/70 TAKES APPROXIMATELY
60                      20 SECONDS.
61
62
63
64              5.      CONFIGURATION REQUIREMENTS
65                      --------------------------
66
67                      DEFAULT PARAMETERS:
68
69                      DEVADR: 176700, VECTOR: 254, BR1: 5, DEVCNT: 1
70
71                      REQUIRED PARAMETERS:
72
73                      NONE
74
75
76
77              6.      DEVICE/OPTION SETUP
78                      -------------------
79
80                      MAKE CERTAIN THAT ALL DRIVES ARE POWERED UP, WRITE ENABLED, AND READY
81
82
83
84              7.      OPERATION OPTIONS
85                      -----------------
86
87                      MEANING OF SR1:
88
89                              BIT02 = 1 COUNT DATA LATE ERRORS, BUT DON'T TYPE THEM.
90                              BIT02 = 0 COUNT AND TYPE DATA LATE ERRORS.
91
92                              BIT04 = 1 PORT B SELECTED
93                              BIT04 = 0 PORT A SELECTED
94
95                              BIT10 = 1 SELECT RANDOM BLOCK ADDRESSING.
96                              BIT10 = 0 SELECT SEQUENTIAL BLOCK ADDRESSING.
97
98                              BIT12 = 1 DUAL PORT OPERATION.
99                              BIT12 = 0 SINGLE PORT OPERATION.
100
101                             BIT15 = 1 32 REGISTERS ON RH70
102                             BIT15 = 0 22 REGISTERS ON RH70
103
104                     NOTE:   BIT04 HAS NO MEANING IN SINGLE PORT OPERATION.
105
106
107             8.      NON-STANDARD PRINTOUTS
108                     ----------------------
109
110                     A.  MOST PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN
111                         THE DEC/X11 DOCUMENT.
112
113                     B.  ERROR MESSAGES DUMP THE CONTENTS OF THE 20 OR 22 REGISTERS.
114                         (DEPENDING IF AN RH11 OR RH70 IS USED)
```

```
115
116                         IN THE FOLLOWING ORDER:
117
118                         RMCS1  RMWC   RMBA   RMDA   RMCS2  RMDS   RMER1  RMAS
119                         RMLA   RMDB   RMMR1  RMDT   RMSN   RMOF   RMDC   RMHR
120                         RMMR2  RMER2  RMEC1  RMEC2 *RMBAE *RMCS3
121
122                         * PRINTED ON ERROR REPORT IF USING AN RH70 CONTROLLER
123
124
125             9.      DUAL PORT SETUP:
126                     ----------------
127                     TO RUN A DUAL PORT SYSTEM, SR1 HAS TO BE MODIFIED TO INDICATE TO
128                     THE MODULE, WHICH PORT THE MODULE IS LOCATED ON. SEE SECTION 7.
129                     FOR SR1 OPTIONS.
130
131                     THE CONTROLLER SELECT SWITCH ON THE RM DRIVE MUST BE IN THE A/B
132                     POSITION.  THIS SWITCH IS ACTIVATED WHEN THE DRIVE IS CYCLED UP.
133                     IF THE SWITCH WAS NOT IN THIS POSITION WHEN DRIVE WAS POWERED UP,
134                     THE FOLLOWING STEPS MUST BE TAKEN.  PLACE THE SWITCH IN THE A/B
135                     POSITION, TURN THE "ON-LINE" SWITCH TO OFF-LINE, THEN TURN THE
136                     "ON-LINE" SWITCH TO ON-LINE.  THIS WILL PUT THE RM DRIVE IN THE
137                     DUAL PORT MODE.
138
139
140
141             10.     BLOCK ADDRESS
142                     -------------
143
144                     EACH SECTOR ON THE RM05, RM03 OR RM02 PACK IS CALLED A BLOCK.
145
146                     AT THE BEGINNING OF EACH TEST CYCLE, THE CXRMCA0 MODULE SELECTS
147                     A STARTING BLOCK ADDRESS SEQUENTIALLY OR RANDOMLY FOR EACH
148                     DATA TRANSFER.
149                     WHEN BIT10 OF SR1 IS RESET(0), CXRMCA0 MODULE SELECTS THE BLOCK
150                     ADDRESS SEQUENTIALLY FROM BLOCK 0 TO THE LAST BLOCK.
151                     WHEN BIT10 OF SR1 IS SET(1), CXRMCA0 MODULE SELECTS THE BLOCK
152                     ADDRESS RANDOMLY BETWEEN BLOCK 0 AND THE LAST BLOCK.
153
154                     IN SINGLE PORT OPERATION, CXRMCA0 MODULE ACCESSES EVERY BLOCK
155                     ON THE TRACKS AS FOLLOWS:
156
157                             RM03/2    = 0 THRU 4.
158                             RM05      = 0 THRU 18.
159
160                     IN DUAL PORT OPERATION, CXRMCA0 MODULE ON THE PORT A ACCESSES
161                     EVERY BLOCK ON THE TRACKS AS FOLLOWS:
162
163                             RM03/2    = 0 THRU 2.
164                             RM05      = 0 THRU 9.
165
166                     IN DUAL PORT OPERATION, CXRMCA0 MODULE ON THE PORT B ACCESSES
167                     EVERY BLOCK ON THE TRACKS AS FOLLOWS:
168
169                             RM03/2    =  3. THRU 4.
170                             RM05      = 10. THRU 18.
171
```

NOTE: THE MANUFACTURES/USERS BAD SECTOR FILE WILL NEVER BE ACCESSED IN ANY WAY BY THIS MODULE.

11. BAD SECTOR TABLE

32. LOCATIONS STARTING AT THE LABEL "BADSPT" ARE RESERVED FOR 16. BAD SECTOR ENTRIES BY THE USER. EACH BAD SECTOR IS SPECIFIED THROUGH ENTERING ITS CYLINDER ADDRESS TO THE FIRST WORD AND ITS TRACK AND SECTOR ADDRESS TO THE SECOND WORD; THE HIGH BYTE ON THE SECOND WORD IS THE TRACK ADDRESS WHILE THE LOW BYTE ON THE SECOND WORD IS THE SECTOR ADDRESS. WHEN THE MODULE DETECTS A DATA TRANSFER ERROR, ON ANY OF THESE SPOTS, THE MODULE WILL SKIP OVER THE BAD SECTOR AND START ANOTHER TEST CYCLE.

THE MODULE ALSO DETECTS BAD SECTORS THROUGH THE "BSE" ERROR BIT (BIT15 OF RMER2). WHENEVER THE "BSE" BIT IS SET AFTER EXECUTING A DATA TRANSFER COMMAND, THE MODULE WILL IGNORE THE ERROR.

%

```
 5                                      .TITLE  RMCA DEC/X11 SYSTEM EXERCISER MODULE
                                ;       DDXCOM  VERSION  6       23-MAY-78
                                        .LIST   BIN
                                ;*****************************************************************
000000                          BEGIN:
000000  000122     115     103 MODNAM: .ASCII  /RMCA / ;MODULE NAME.
000005  000000                 XFLAG:  .BYTE   0PEN                ;USED TO KEEP TRACK OF RBUFF USAGE
000006  176700                 ADDR:   176700+0                    ;1ST DEVICE ADDR.
000010  000254                 VECTOR: 254+0                       ;1ST DEVICE VECTOR.
000012  000240                 BR1:    .BYTE   PRTY5+0             ;1ST BR LEVEL.
000013  000000                 BR2:    .BYTE   PRTY0+0             ;2ND BR LEVEL.
000014  000001                 DVID1:  0+1                         ;DEVICE INDICATOR 1.
000016  000000                 SR1:    OPEN                        ;SWITCH REGISTER 1
000020  000000                 SR2:    OPEN                        ;SWITCH REGISTER 2
000022  000000                 SR3:    OPEN                        ;SWITCH REGISTER 3
000024  000000                 SR4:    OPEN                        ;SWITCH REGISTER 4
                                ;*****************************************************************
000026  150000                 STAT:   150000                      ;STATUS WORD.
000030  001670'                INIT:   START                       ;MODULE START ADDR.
000032  000252'                SPOINT: MODSP                       ;MODULE STACK POINTER.
000034  000000                 PASCNT: 0                           ;PASS COUNTER.
000036  000310                 ICONT:  200.                        ;# OF ITERATIONS PER PASS=200.
000040  000000                 ICOUNT: 0                           ;LOC TO COUNT ITERATIONS
000042  000000                 SOFCNT: 0                           ;LOC TO SAVE TOTAL SOFT ERRORS
000044  000000                 HRDCNT: 0                           ;LOC TO SAVE TOTAL HARD ERRORS
000046  000000                 SOFPAS: 0                           ;LOC TO SAVE SOFT ERRORS PER PASS
000050  000000                 HRDPAS: 0                           ;LOC TO SAVE HARD ERRORS PER PASS
000052  000000                 SYSCNT: 0                           ;# OF SYS ERRORS ACCUMULATED
000054  000000                 RANNUM: 0                           ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056                          CONFIG:                             ;RESERVED FOR MONITOR USE
000056  000000                 RES1:   0                           ;RESERVED FOR MONITOR USE
000060  000000                 RES2:   0                           ;RESERVED FOR MONITOR USE
000062  000000                 SVR0:   OPEN                        ;LOC TO SAVE R0.
000064  000000                 SVR1:   OPEN                        ;LOC TO SAVE R1.
000066  000000                 SVR2:   OPEN                        ;LOC TO SAVE R2.
000070  000000                 SVR3:   OPEN                        ;LOC TO SAVE R3.
000072  000000                 SVR4:   OPEN                        ;LOC TO SAVE R4.
000074  000000                 SVR5:   OPEN                        ;LOC TO SAVE R5.
000076  000000                 SVR6:   OPEN                        ;LOC TO SAVE R6.
000100  000000                 CSRA:   OPEN                        ;ADDR OF CURRENT CSR.
000102                          SBADR:                              ;ADDR OF GOOD DATA, OR
000102  000000                 ACSR:   OPEN                        ;CONTENTS OF CSR.
000104                          WASADR:                             ;ADDR OF BAD DATA, OR
000104  000000                 ASTAT:  OPEN                        ;STATUS REG CONTENTS.
000106                          ERRTYP:                             ;TYPE OF ERROR
000106  000000                 ASB:    OPEN                        ;EXPECTED DATA.
000110  000000                 AWAS:   OPEN                        ;ACTUAL DATA.
000112  002130'                RSTRT:  RESTRT                      ;RESTART ADDRESS AFTER END OF PASS
000114  000000                 WDTU:   OPEN                        ;WORDS TO MEMORY PER ITERATION
000116  000000                 WDFR:   OPEN                        ;WORDS FROM MEMORY PER ITERATION
000120  000000                 INTR:   OPEN                        ;# OF INTERRUPTS PER ITERATION
000122  000000                 IDNUM:  0                           ;MODULE IDENTIFICATION NUMBER=0
000124  000416'                RBUFVA: RBUF                        ;READ BUFFER VIRTUAL ADDRESS
000126  000000                 RBUFPA: OPEN                        ;READ BUFFER PHYSICAL ADDRESS
000130  000000                 RBUFEA: OPEN                        ;READ BUFFER EA BITS
000132  000400                 RBUFSZ: 256.                        ;SIZE OF THE READ BUFFER
000134  000000                 WBUFPA: OPEN                        ;WRITE BUFFER PHYSICAL ADDRESS
```



```
000136  000000                 WBUFEA: OPEN                        ;WRITE BUFFER EA BITS
000140  000400                 WBUFRQ: 256.                        ;WRITE BUFFER SIZE REQUESTED
000142  000000                 WBUFSZ: OPEN                        ;WRITE BUFFER SIZE AVAILABLE
000144  000000                 CDERCT: OPEN                        ;CDATA/DATCK ERROR COUNT
000146  000000                 CDWDCT: OPEN                        ;CDATA/DATCK WORD COUNT
000150  000000                 FREE:   OPEN                        ;RESERVED FOR FUTURE USE
        000040                          .REPT   SPSIZ               ;MODULE STACK STARTS HERE.
000252                          MODSP:
                                ;*****************************************************************
 6 000252                       BADSPT:
12 000252  177777                       .WORD   177777
   000254  177777                       .WORD   177777
   000256  177777                       .WORD   177777
   000260  177777                       .WORD   177777
   000262  177777                       .WORD   177777
   000264  177777                       .WORD   177777
   000266  177777                       .WORD   177777
   000270  177777                       .WORD   177777
   000272  177777                       .WORD   177777
   000274  177777                       .WORD   177777
   000276  177777                       .WORD   177777
   000300  177777                       .WORD   177777
   000302  177777                       .WORD   177777
   000304  177777                       .WORD   177777
   000306  177777                       .WORD   177777
   000310  177777                       .WORD   177777
   000312  177777                       .WORD   177777
   000314  177777                       .WORD   177777
   000316  177777                       .WORD   177777
   000320  177777                       .WORD   177777
   000322  177777                       .WORD   177777
   000324  177777                       .WORD   177777
   000326  177777                       .WORD   177777
   000330  177777                       .WORD   177777
   000332  177777                       .WORD   177777
   000334  177777                       .WORD   177777
   000336  177777                       .WORD   177777
   000340  177777                       .WORD   177777
   000342  177777                       .WORD   177777
   000344  177777                       .WORD   177777
   000346  177777                       .WORD   177777
   000350  177777                       .WORD   177777
14
15                              ;DO NOT CHANGE THE ORDER OF THE NEXT 4 LOCATIONS, THEY ARE
16                              ;NEEDED FOR MAP22 ROUTINE.
17 000352  000000                 PA18:   0
18 000354  000000                 XMEM:   0
19 000356  000000                 PA22:   0
20 000360  000000                 EA22:   0
21
22 000362  000000                 DRPDRV: 0               ;DROP DRIVE # 1
23 000364  000000                 FLAG70: 0               ;BIT15 SET INDICATES THAT AN 11/70 PRESENT
24 000366  000000                 CYLIND: 0               ;CYLINDER ADDRESS FOR DRIVER
25 000370  000000                 DSKADR: 0               ;COMPOSITE TRACK AND SECTOR ADDRESS FOR DRIVER
26 000372  000000                 DVICE:  0               ;DEVICE NUMBER
27 000374  000000                 UNITNO: 0               ;# OF UNIT BEING TESTED
28 000376  000000                 BADSEC: 0               ;BAD SECTOR FLAG, USER BAD SECTOR DETECTED = -1,
```

```
                                                        ;BSE = 100000
000400  000000                  DLICNT: 0               ;DATA LATE ERROR COUNT
000402  000000                  FUNC:   0               ;FUNCTION TO BE PERFORMED
000404  000000                  ZERO:   0
000406  000000                  FERADR: 0
000410  000000                  CLK:    0               ;USED IN TIMEOUT LOOPS
000412  000000                  TRY:    0
000414  000000                  DRVTYP: 0               ;DRIVE TYPE FLAG, RM02 = 5, RM03 = 4 AND RM05 = 7

000416                          RBUF:   .BLKW   256.

                                ;RH/RM SAVED REGISTER POINTERS

001416  001614'                 TABLE:  S+NCS1
001420  001616'                         S+NWC
001422  001620'                         S+NBA
001424  001622'                         S+NDA
001426  001624'                         S+NCS2
001430  001626'                         S+NDS
001432  001630'                         S+NER1
001434  001632'                         S+NAS
001436  001634'                         S+NLA
001440  001636'                         S+NDB
001442  001640'                         S+NMR1
001444  001642'                         S+NDT
001446  001644'                         S+NSN
001450  001646'                         S+NOF
001452  001650'                         S+NDC
001454  001652'                         S+NMR
001456  001654'                         S+NMR2
001460  001656'                         S+NER2
001462  001660'                         S+NEC1
001464  001662'                         S+NEC2
001466  001664'                         S+NBAE
001470  001666'                         S+NCS3
001472  177777                          177777

001474  000000                  RMCS1:  0               ;CONTROL STATUS REGISTER #1
001476  000000                  RMWC:   0               ;WORD COUNT REGISTER
001500  000000                  RMBA:   0               ;BUS ADDRESS REGISTER
001502  000000                  RMDA:   0               ;DESIRED TRACK/SECTOR ADDRESS REGISTER
001504  000000                  RMCS2:  0               ;CONTROL STATUS REGISTER #2
001506  000000                  RMDS:   0               ;DRIVE STATUS REGISTER
001510  000000                  RMER1:  0               ;ERROR REGISTER #1
001512  000000                  RMAS:   0               ;ATTENTION SUMMARY REGISTER
001514  000000                  RMLA:   0               ;LOOK AHEAD REGISTER
001516  000000                  RMDB:   0               ;DATA BUFFER REGISTER
001520  000000                  RMMR1:  0               ;MAINTENANCE REGISTER #1
001522  000000                  RMDT:   0               ;DRIVE TYPE REGISTER
001524  000000                  RMSN:   0               ;SERIAL NUMBER REGISTER
001526  000000                  RMOF:   0               ;OFFSET REGISTER
001530  000000                  RMDC:   0               ;DESIRED CYLINDER ADDRESS REGISTER
001532  000000                  RMMR:   0               ;HOLDING REGISTER
001534  000000                  RMMR2:  0               ;MAINTENANCE REGISTER #2
001536  000000                  RMER2:  0               ;ERROR REGISTER #2
001540  000000                  RMEC1:  0               ;ECC POSITION REGISTER
001542  000000                  RMEC2:  0               ;ECC PATTERN REGISTER
```



```
001544  000000                  RMBAE:  0               ;BUS ADDRESS EXTENSION REGISTER (RH70)
001546  000000                  RMCS3:  0               ;CONTROL STATUS REGISTER #3 (RH70)

001550  000406'                 XFERAD: FERADR
001552  177777                          177777

                                ;REGISTER ADDRESS INDEX VALUE

                000000          NCS1    = 0
                000002          NWC     = 2
                000004          NBA     = 4
                000006          NDA     = 6
                000010          NCS2    = 10
                000012          NDS     = 12
                000014          NER1    = 14
                000016          NAS     = 16
                000020          NLA     = 20
                000022          NDB     = 22
                000024          NMR1    = 24
                000026          NDT     = 26
                000030          NSN     = 30
                000032          NOF     = 32
                000034          NDC     = 34
                000036          NMR     = 36
                000040          NMR2    = 40
                000042          NER2    = 42
                000044          NEC1    = 44
                000046          NEC2    = 46
                000050          NBAE    = 50
                000052          NCS3    = 52

                                ;DISK ADDRESS EQUATES FOR LIMITS TABLES

                000000          FT      = 0             ;FIRST TRACK
                000002          LT      = 2             ;LAST TRACK
                000004          LS      = 4             ;LAST SECTOR
                000006          LC      = 6             ;LAST CYLINDER
                000010          BST     = 10            ;BAD SECTOR TRACK (DEC 144 FILE)
                000012          SEC     = 12            ;CURRENT SECTOR
                000014          TRK     = 14            ;CURRENT TRACK
                000016          CYL     = 16            ;CURRENT CYLINDER

                                ;TABLE OF ADDRESS LIMITS FOR RM05 AND RM03/02

                                ;NOTES: (1) WORD 1 (FIRST TRACK) AND WORD 2 (LAST TRACK) WILL BE
                                ;           INITIALIZED AT PROGRAM START.
                                ;
                                ;       (2) DO NOT ALTER THE BAD SECTOR TRACK OR LAST CYLINDER
                                ;           VALUES. IF EITHER VALUE IS CHANGED, THE BAD SECTOR
                                ;           FILE(DEC 144 FILE) MAY BE DESTROYED.

001554  000000                  TAB03:  .WORD   0       ;FIRST TRACK
001556  000000                          .WORD   0       ;LAST TRACK
001560  000037                          .WORD   31.     ;LAST SECTOR
001562  001466                          .WORD   822.    ;LAST CYLINDER
001564  000004                          .WORD   4.      ;BAD SECTOR TRACK (DEC 144 FILE)
001566  000000                          .WORD   0       ;CURRENT SECTOR ADDRESS
```

```
143 001570  000000                      .WORD   0       ;CURRENT TRACK ADDRESS
144 001572  000000                      .WORD   0       ;CURRENT CYLINDER ADDRESS
145
146 001574  000000              TAB05:  .WORD   0       ;FIRST TRACK
147 001576  000000                      .WORD   0       ;LAST TRACK
148 001600  000037                      .WORD   31.     ;LAST SECTOR
149 001602  001466                      .WORD   822.    ;LAST CYLINDER
150 001604  000022                      .WORD   18.     ;BAD SECTOR TRACK (DEC 144 FILE)
151 001606  000000                      .WORD   0       ;CURRENT SECTOR ADDRESS
152 001610  000000                      .WORD   0       ;CURRENT TRACK ADDRESS
153 001612  000000                      .WORD   0       ;CURRENT CYLINDER ADDRESS
154
155 001614                      S:      .BLKW   22.     ;STORE THE RH/RM REGISTER CONTENTS
156
157                             ;CONTROL STATUS REGISTER #1
158
159         100000              SC      = 100000        ;SPECIAL CONDITION
160         040000              TRE     = 40000         ;TRANSFER ERROR
161         020000              MCPE    = 20000         ;CONTROL BUS PARITY ERROR
162         004000              DVA     = 4000          ;DRIVE AVAILABLE
163         000200              RDY     = 200           ;CONTROL READY
164         000001              GO      = 1             ;GO BIT
165
166                             ;CONTROL STATUS REGISTER #2
167
168         000040              CLR     = 40            ;MASSBUS CLEAR COMMAND
169         000100              IR      = 100           ;SILO INPUT READY
170         000200              OR      = 200           ;SILO OUTPUT READY
171         000400              MDPE    = 400           ;MASS BUS PARITY
172         001000              MXF     = 1000          ;MISSED TRANSFER ERROR
173         002000              PGE     = 2000          ;PROGRAM ERROR (RH)
174         004000              NEM     = 4000          ;NON-EXISTENCE MEMORY
175         010000              NED     = 10000         ;NON-EXISTENCE DRIVE
176         020000              UPE     = 20000         ;UNIBUS PARITY ERROR
177         040000              WCE     = 40000         ;WRITE CHECK ERROR
178         100000              DLT     = 100000        ;DATA LATE ERROR
179
180                             ;RM DRIVE STATUS REGISTER
181
182         000001              OM      = 1             ;OFFSET MODE
183         000100              VV      = 100           ;VOLUMNE VALID
184         000200              DRY     = 200           ;DRIVE READY
185         000400              DPR     = 400           ;DRIVE PRESENT
186         001000              PGM     = 1000          ;PROGRAMMABLE STATE
187         002000              LBT     = 2000          ;LAST BLOCK TRANSFERRED
188         004000              WRL     = 4000          ;WRITE LOCK
189         010000              MOL     = 10000         ;MEDIUM ONLINE
190         020000              PIP     = 20000         ;PROSITIONING OPERATION IN PROGRESS
191         040000              ERR     = 40000         ;DRIVE ERROR
192         100000              ATA     = 100000        ;DRIVE ATTENTION
193
194                             ;RM ERROR REGISTER #1
195
196         000001              ILF     = 1             ;ILLEGAL FUNCTION
197         000002              ILR     = 2             ;ILLEGAL REGISTER
198         000004              RMR     = 4             ;REGISTER MODIFICATION REFUSE
199         000010              PAR     = 10            ;PARITY ERROR
```



```
200         000020              FER     = 20            ;FORMAT ERROR
201         000040              WCF     = 40            ;WRITE CLOCK FAIL
202         000100              ECH     = 100           ;HARD ERROR ECC
203         000200              HCE     = 200           ;HEADER COMPARE FAIL
204         000400              HCRC    = 400           ;HEADER CRC ERROR
205         001000              AOE     = 1000          ;ADDRESS OVERFLOW ERROR
206         002000              IAE     = 2000          ;INVALID ADDRESS ERROR
207         004000              WLE     = 4000          ;WRITE LOCK ERROR
208         010000              DTE     = 10000         ;DRIVE TIMING ERROR
209         020000              OPI     = 20000         ;OPERATION INCOMPLETE
210         040000              UNS     = 40000         ;DRIVE UNSAFE
211         100000              DCK     = 100000        ;DATA CHECK ERROR
212
213                             ;RM ERROR REGISTER #2
214
215         000010              OPE     = 10            ;DATA PARITY DURING WRITE
216         000200              DVC     = 200           ;DEVICE CHECK
217         002000              LBC     = 2000          ;LOST BIT CLOCK
218         004000              LSC     = 4000          ;LOST SYSTEM CLOCK
219         040000              SKI     = 40000         ;SEEK INCOMPLETE
220         100000              BSE     = 100000        ;BAD SECTOR ERROR
221
222                             ;RM OFFSET REGISTER
223
224         010000              FMT     = 10000         ;16 BIT DATA FORMAT
225
```

```
                                        .SBTTL  START OF PROGRAM
                                
001670  012767  000400  176216  START:  MOV     #256.,WDTC      ;256 WORDS TO MEM/ITERATION
001676  012767  000400  176212          MOV     #256.,WDFR      ;256 WORDS FROM MEM/ITERATION
001704  012767  000012  176206          MOV     #10.,INTR       ;10 INTERRUPTS/ITERATION
001712  016767  176076  176452          MOV     DVID1,DVICE     ;GET DRIVES TO TEST
001720  016706  176106                  MOV     SPOINT,R6       ;INITIATE STACK POINT
001724  012767  000003  177622          MOV     #3.,TAB03+FT    ;SETUP RM03/2 FIRST TRACK AND
001732  012767  000004  177616          MOV     #4.,TAB03+LT    ;SETUP LAST TRACK, IF THE DUAL PORT AND
                                                                ;PORT B ARE SELECTED.
001740  012767  000012  177626          MOV     #10.,TAB05+FT   ;SETUP RM05 FIRST TRACK AND
001746  012767  000022  177622          MOV     #18.,TAB05+LT   ;SETUP LAST TRACK, IF THE DUAL PORT AND
                                                                ;PORT B ARE SELECTED.
001754  016700  176036                  MOV     SR1,R0          ;GET SWITCH REGISTER #1
001760  042700  167757                  BIC     #^C<BIT12!BIT4>,R0
001764  022700  010020                  CMP     #BIT12!BIT4,R0  ;DUAL PORT AND PORT B ?
001770  001415                          BEQ     1$              ;BR IF YES
001772  005067  177556                  CLR     TAB03+FT        ;NO, SETUP FIRST TRACK TO 0
001776  005067  177572                  CLR     TAB05+FT
002002  032700  010000                  BIT     #BIT12,R0       ;DUAL PORT AND PORT A ?
002006  001406                          BEQ     1$              ;BR IF NO
002010  012767  000002  177540          MOV     #2.,TAB03+LT    ;SETUP RM03/2 LAST TRACK AND
002016  012767  000011  177552          MOV     #9.,TAB05+LT    ;SETUP RM05 LAST TRACK FOR PORT A
002024                          1$:
002024  012767  177777  177534          MOV     #-1,TAB03+SEC   ;INITIALIZE THE SECTOR ADDRESS
002032  012767  177777  177546          MOV     #-1,TAB05+SEC
002040  016767  177510  177522          MOV     TAB03,TAB03+TRK ;INITIALIZE THE TRACK ADDRESS
002046  016767  177522  177534          MOV     TAB05,TAB05+TRK
002054  005067  177512                  CLR     TAB03+CYL       ;INITIALIZE THE CYLINDER ADDRESS
002060  005067  177526                  CLR     TAB05+CYL
002064  005067  176310                  CLR     DLTCNT          ;RESET THE DATA LATE ERROR COUNT
002070  004767  003556                  JSR     PC,SETUP        ;SETUP REGISTER ADDRESSES AND PSEL BIT
002074  012767  177777  176272          MOV     #-1,UNITNO      ;PRE-SET SET UNIT NUMBER
002102                          2$:
002102  004767  001606                  JSR     PC,GETDRV       ;GO GET A DRIVE
002106  000407                          BR      3$              ;RETURN HERE WHEN ALL DRIVES ARE DONE
002110  004567  003116                  JSR     R5,READY        ;SEE IF DRIVE IS AVAILABLE
002114  000772                          BR      2$              ;BR IF NOT AVAILABLE
002116  012777  000023  177350          MOV     #23,@#RMCS1     ;GO ISSUE A PACK ACKNOWLEDGE COMMAND TO
                                                                ;SET "VV" BIT IN DRIVE STATUS.
002124  000766                          BR      2$              ;DO NEXT DRIVE
002126                          3$:
002126  000404                          BR      RSTRT1          ;START THE PASS

002130  005767  175700          RESTRT: TST     PASCNT          ;SUPPORT DT03
002134  001001                          BNE     RSTRT1
002136  000654                          BR      START
002140                          RSTRT1:
002140  104415  000000' 000124'          GETPAS,BEGIN, RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
002146                          LOOP1:
002146  012767  177777  176220          MOV     #-1,UNITNO      ;PRE-SET UNIT NUMBER
002154  004767  001140                  JSR     PC,GENADR       ;GENERATE BLOCK ADDRESS
002160                          LOOP2:
002160  004767  001530                  JSR     PC,GETDRV       ;GO GET A DRIVE
002164  000413                          BR      2$              ;RETURNS HERE IF ALL DRIVES DONE
002166  004567  003040                  JSR     R5,READY        ;SEE IF DRIVE IS READY
002172  000772                          BR      LOOP2           ;LOOP BACK IF NOT READY
```



```
002174  004767  001450                  JSR     PC,LOADR        ;LOAD BLOCK ADDRESS
002200  004767  000022                  JSR     PC,CYCLE        ;COMMAND SEQUENCE ROUTINE
002204  000240                          NOP                     ;ERROR EXIT FOR CYCLE ROUTINE
002206                          1$:
002206  104413  000000'                  ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
002212  000762                          BR      LOOP2           ;DO IT TO NEXT DRIVE
002214                          2$:
002214  005767  176152                  TST     DVICE           ;ARE ALL DEVICES DROPPED ?
002220  001352                          BNE     LOOP1           ;BR IF NO

002222  104410  000000'                  END$,BEGIN              ;
```

```
                                        ;*** SUBROUTINES ***
                                        ;
                                        ;CALL
                                        ;       JSR     PC,CYCLE        ;CALL ROUTINE
                                        ;       BR      ?               ;ERROR EXIT
                                        ;       RET                     ;NORMAL RETURN
                                        ;
002226  005067  176160          CYCLE:  CLR     TRY             ;CLEAR THE RETRY COUNT
002232  104414  000000'                  GWBUFS, BEGIN           ;GET WRITE BUFFER INFORMATION
002236  004767  003132                  JSR     PC,WAIT         ;WAIT FOR DRIVE
002242  000442                          BR      2$              ;EXIT ON ERROR
002244  004567  000554                  JSR     R5,SEEK         ;DO AN EXPLICIT SEEK
002250  000434                          BR      1$              ;EXIT IF SEEK ERROR
002252  004567  000106                  JSR     R5,WRITE        ;WRITE DATA COMMAND
002256  000431                          BR      1$              ;FAILURE EXIT
002260  004767  003110                  JSR     PC,WAIT         ;WAIT FOR DRIVE
002264  000431                          BR      2$              ;EXIT ON ERROR
002266  004567  000532                  JSR     R5,SEEK         ;DO AN EXPLICIT SEEK
002272  000423                          BR      1$              ;EXIT IF SEEK ERROR
002274  004567  000144                  JSR     R5,WRITCK       ;WRITE CHECK DATA COMMAND
002300  000420                          BR      1$              ;FAILURE EXIT
002302  004767  003066                  JSR     PC,WAIT         ;WAIT FOR DRIVE
002306  000420                          BR      2$              ;EXIT ON ERROR
002310  004567  000510                  JSR     R5,SEEK         ;DO AN EXPLICIT SEEK
002314  000412                          BR      1$              ;EXIT IF SEEK ERROR
002316  004567  000202                  JSR     R5,READ         ;READ DATA COMMAND
002322  000407                          BR      1$              ;FAILURE EXIT
002324  104412  000000' 000126'          CDATA$,BEGIN,PBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
002332  002362'                         3$                      ; IF ERROR, RETURN AT TAG 3$
002334  062716  000002                  ADD     #2,(SP)         ;ADJUST GOOD RETURN FOR CYCLE ROUTINE
002340  000410                          BR      3$              ;EXIT
002342                          1$:
002342  005767  176014                  TST     DRPDRV          ;DROP THE DRIVE ?
002346  001405                          BEQ     3$              ;BR IF NO
002350  004767  001406          2$:     JSR     PC,DROP         ;DROP THE DRIVE
002354  104403  000000' 007174'          MSGNS,BEGIN,DRM ;ASCII MESSAGE CALL WITH COMMON HEADER
002362  000207                  3$:     RTS     PC              ;EXIT
```



```
                                        .SBTTL  COMMAND SUBROUTINES
                                        ;
                                        ;CALL
                                        ;
                                        ;       JSR     R5,COMMAND      ;CALL COMMAND ROUTINE
                                        ;       BR      ?               ;ERROR RETURN
                                        ;       RET                     ;NORMAL RETURN
                                        ;
                                        ;THE ROUTINE COMMAND NAMES:
                                        ;WRITE  - WRITE DATA, USING WRITE BUFFER,
                                        ;WRITCK - WRITE CHECK DATA, USING WRITE BUFFER,
                                        ;READ   - READ DATA, USING READ BUFFER
                                        ;WRDCKR - WRITE CHECK DATA, USING READ BUFFER
                                        ;WRHCKR - WRITE CHECK HEADER AND DATA, USING READ BUFFER
                                        ;READHD - READ HEADER AND DATA, USING READ BUFFER
                                        ;SEEK   - SEEK COMMAND
                                        ;
002364  012767  000161  176010  WRITE:  MOV     #161,FUNC       ;LOAD WRITE FUNCTION
002372  012767  002364' 176006          MOV     #WRITE,FERADR   ;SAVE WHERE WE WERE
002400  016746  175536                  MOV     WBUFSZ,-(SP)    ;GET WRITE SIZE
002404  005416                          NEG     (SP)            ;NEGATE IT
002406  012677  177064                  MOV     (SP)+,@RMWC     ;LOAD WORD COUNT
002412  016777  175516  177060          MOV     WBUFPA,@RMBA    ;LOAD BUFFER ADDRESS
002420  016746  175512                  MOV     WBUFEA,-(SP)    ;GET EXTENDED MEMORY BITS
002424  006316                          ASL     (SP)            ;SHIFT 4 PLACES TO THE LEFT
002426  006316                          ASL     (SP)            ;TO LINE UP WITH RMCS1
002430  006316                          ASL     (SP)
002432  006316                          ASL     (SP)
002434  012667  175714                  MOV     (SP)+,XMEM      ;SAVE THE SHIFTED BITS
002440  000167  000400                  JMP     GO1             ;CONTINUE
                                        ;
002444  012767  000151  175730  WRITCK: MOV     #151,FUNC       ;LOAD WRITE-CHECK FUNCTION
002452  012767  002444' 175726          MOV     #WRITCK,FERADR  ;SAVE WHERE WE WERE
002460  016746  175456                  MOV     WBUFSZ,-(SP)    ;GET WRITE SIZE
002464  005416                          NEG     (SP)            ;NEGATE IT
002466  012677  177004                  MOV     (SP)+,@RMWC     ;LOAD WORD COUNT
002472  016777  175436  177000          MOV     WBUFPA,@RMBA    ;LOAD BUFFER ADDRESS
002500  016746  175432                  MOV     WBUFEA,-(SP)    ;GET EXTENDED MEMORY BITS
002504  006316                          ASL     (SP)            ;SHIFT 4 PLACES TO THE LEFT
002506  006316                          ASL     (SP)            ;TO LINE UP WITH RMCS1
002510  006316                          ASL     (SP)
002512  006316                          ASL     (SP)
002514  012667  175634                  MOV     (SP)+,XMEM      ;SAVE THE SHIFTED BITS
002520  000167  000320                  JMP     GO1             ;CONTINUE
                                        ;
002524  012767  000171  175650  READ:   MOV     #171,FUNC       ;LOAD READ FUNCTION
002532  012767  002524' 175646          MOV     #READ,FERADR    ;SAVE WHERE WE WERE
002540  016746  175366                  MOV     RBUFSZ,-(SP)    ;GET READ SIZE
002544  005416                          NEG     (SP)            ;NEGATE IT
002546  012677  176724                  MOV     (SP)+,@RMWC     ;LOAD WORD COUNT
002552  016777  175350  176720          MOV     RBUFPA,@RMBA    ;LOAD BUFFER ADDRESS
002560  016746  175344                  MOV     RBUFEA,-(SP)    ;GET EXTENDED MEMORY BITS
002564  006316                          ASL     (SP)            ;SHIFT 4 PLACES TO THE LEFT
002566  006316                          ASL     (SP)            ;TO LINE UP WITH RMCS1
002570  006316                          ASL     (SP)
002572  006316                          ASL     (SP)
002574  012667  175554                  MOV     (SP)+,XMEM      ;SAVE THE SHIFTED BITS
```

```
 43 002630  000167  000240                 JMP     GO1             ;CONTINUE
 44
 45 002634  012767  000151  175570  WRDCKR: MOV     #151,FUNC       ;LOAD WRITE-CHECK FUNCTION
 46 002642  012767  002604' 175566          MOV     #WRDCKR,FERADR  ;SAVE WHERE WE WERE
 47 002650  016746  175306                  MOV     RBUFSZ,-(SP)    ;GET WRITE SIZE
 48 002654  005416                          NEG     (SP)            ;NEGATE IT
 49 002656  012677  176644                  MOV     (SP)+,@RMWC     ;LOAD WORD COUNT
 50 002662  016777  175270  176640          MOV     RBUFPA,@RMBA    ;LOAD BUFFER ADDRESS
 51 002670  016746  175264                  MOV     RBUFEA,-(SP)    ;GET EXTENDED MEMORY BITS
    002674  006316                          ASL     (SP)            ;SHIFT 4 PLACES TO THE LEFT
    002676  006316                          ASL     (SP)            ;TO LINE UP WITH RHCS1
    002700  006316                          ASL     (SP)
    002702  006316                          ASL     (SP)
    002704  012667  175474                  MOV     (SP)+,XMEM      ;SAVE THE SHIFTED BITS
 52 002710  000167  000160                  JMP     GO1             ;CONTINUE
 53
 54 002714  012767  000153  175510  WRHCKR: MOV     #153,FUNC       ;LOAD WRITE-CHECK-HEADER-AND-DATA FUNCTION
 55 002722  012767  002604' 175506          MOV     #WRHCKR,FERADR  ;SAVE WHERE WE WERE
 56 002730  016746  175226                  MOV     RBUFSZ,-(SP)    ;GET WRITE SIZE
 57 002734  005416                          NEG     (SP)            ;NEGATE IT
 58 002736  012677  176564                  MOV     (SP)+,@RMWC     ;LOAD WORD COUNT
 59 002742  016777  175210  176560          MOV     RBUFPA,@RMBA    ;LOAD BUFFER ADDRESS
 60 002750  016746  175204                  MOV     RBUFEA,-(SP)    ;GET EXTENDED MEMORY BITS
    002754  006316                          ASL     (SP)            ;SHIFT 4 PLACES TO THE LEFT
    002756  006316                          ASL     (SP)            ;TO LINE UP WITH RHCS1
    002760  006316                          ASL     (SP)
    002762  006316                          ASL     (SP)
    002764  012667  175414                  MOV     (SP)+,XMEM      ;SAVE THE SHIFTED BITS
 61 002770  000167  000100                  JMP     GO1             ;CONTINUE
 62
 63 002774  012767  000173  175430  READHD: MOV     #173,FUNC       ;READ HEADER AND DATA COMMAND
 64 003002  012767  002744' 175426          MOV     #READHD,FERADR  ;SAVE THE CALLING ADDRESS
 65 003010  016746  175146                  MOV     RBUFSZ,-(SP)    ;GET THE BUFFER SIZE
 66 003014  005416                          NEG     (SP)            ;THE NEGATIVE WORD COUNT
 67 003016  012677  176504                  MOV     (SP)+,@RMWC     ;LOAD THE WORD COUNT
 68 003022  016777  175130  176500          MOV     RBUFPA,@RMBA    ;LOAD BUFFER ADDRESS
 69 003030  016746  175124                  MOV     RBUFEA,-(SP)    ;GET EXTENDED MEMORY BITS
    003034  006316                          ASL     (SP)            ;SHIFT 4 PLACES TO THE LEFT
    003036  006316                          ASL     (SP)            ;TO LINE UP WITH RHCS1
    003040  006316                          ASL     (SP)
    003042  006316                          ASL     (SP)
    003044  012667  175334                  MOV     (SP)+,XMEM      ;SAVE THE SHIFTED BITS
 70 003050  000167  000020                  JMP     GO1             ;EXECUTE THE COMMAND
 71
 72 003054  012767  000105  175350  SEEK:   MOV     #105,FUNC       ;SEEK COMMAND
 73 003062  012767  003024' 175346          MOV     #SEEK,FERADR    ;CALLING ADDRESS
 74 003070  000167  000104                  JMP     GO2             ;EXECUTE THE SEEK COMMAND
 75
```



```
  1                                 .SBTTL  COMMAND EXECUTE ROUTINE
  2
  3 003074                          GO1:
  4 003074  005767  175314                  TST     FLAG70          ;11/70???
  5 003100  100034                          BPL     1$              ;NO
  6 003102  017767  176422  175272          MOV     @RMBA,PA18      ;GET 18 BIT ADDR
  7 003110  006267  175270                  ASR     XMEM            ;SHIFT EA BITS TO POSITION 4,5
  8 003114  006267  175264                  ASR     XMEM
  9 003120  006267  175260                  ASR     XMEM
 10 003124  006267  175254                  ASR     XMEM
 11 003130  104416  000000' 000352'        MAP22$, BEGIN,PA18              ; GET 22-BIT ADDR FROM 18-BIT ADDR
 12 003136  016777  175244  176364          MOV     PA22,@RMBA      ;LOAD BA REG
 13 003144  016777  175240  176422          MOV     EA22,@RMBAE     ;LOAD BAE REG
 14 003152  042767  000074  175230          BIC     #74,EA22        ;CLEAR UNWANTED BITS
 15 003160  000367  175224                  SWAB    EA22            ;LOAD INTO BITS 8,9
 16 003164  016767  175220  175212          MOV     EA22,XMEM       ;LOAD XMEM TO SET INTO FUNCTION CODE
 17 003172  056767  175206  175232  1$:     BIS     XMEM,FUNC       ;LOAD EXTENDED MEMORY BITS
 18 003200                          GO2:
 19 003200  012777  010000  176350          MOV     #FMT,@RMOF      ;SET 16 BIT FORMAT
 20 003206  016777  175204  176344          MOV     CYLIND,@RMDC    ;LOAD THE CYLINDER ADDRESS
 21 003214  016777  175200  176310          MOV     DSKADR,@RMDA    ;LOAD THE TRACK AND SECTOR ADDRESS
 22 003222  016777  175204  176274          MOV     FUNC,@RMCS1     ;EXECUTE THE FUNCTION
 23 003230  104400  000000'                 EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
 24 003234                          NTRUPT:
 25                                         ;-----------------------------------------------------------------
    003234  000004  000000' 003242'        PIRQS,BEGIN,1S          ; QUEUE UP TO CONTINUE AT 1S AND RTI
                                           ;-----------------------------------------------------------------
 26
 27 003242  004567  000602          1$:     JSR     R5,ERRORS       ;GO CHECK FOR ERRORS
 28 003246  000403                          BR      2$              ;ERRORS DETECTED
 29 003250  062705  000002                  ADD     #2,R5           ;NO ERRORS DETECTED
 30 003254  000427                          BR      3$              ;EXIT
 31 003256                          2$:
 32 003256  005767  175130                  TST     DRPDRV          ;DROP DRIVE ?
 33 003262  001031                          BNE     5$              ;BR IF YES
 34 003264  004567  001772                  JSR     R5,READY        ;GET THE DRIVE AGAIN
 35 003270  000426                          BR      5$              ;IF FAILS, EXIT
 36 003272  022767  177777  175126          CMP     #-1,BADSEC      ;USR DETECTED BAD SPOT ?
 37 003300  001422                          BEQ     5$              ;BR IF YES
 38 003302  032767  100000  175116          BIT     #BSE,BADSEC     ;BAD SECTOR ERROR ?
 39 003310  001016                          BNE     5$              ;BR IF YES
 40 003312  005267  175124                  INC     TRY             ;INCREMENT THE RETRY COUNT
 41 003316  026727  175120  000003          CMP     TRY,#3          ;OVER 3 RETRIES ?
 42 003324  103406                          BLO     4$              ;BR IF NOT
 43 003326  104403  000000' 007050'        MSGNS,BEGIN,EXCED       ;ASCII MESSAGE CALL WITH COMMON HEADER
 44 003334  005067  175102          3$:     CLR     TRY
 45 003340  000402                          BR      5$              ;EXIT
 46 003342  162705  000004          4$:     SUB     #4,R5           ;ADJUST ADDRESS FOR RETRY
 47 003346  000205                  5$:     RTS     R5              ;EXIT
 48
 49
 50                                 ;THIS ROUTINE IS USED TO GENERATOR A DISK ADDRESS ON WHICH THE
 51                                 ;DRIVE WILL SEEK, WRITE, WRITE CHECK AND READ ON.
 52                                 ;
 53                                 ;CALL
 54                                 ;       JSR     PC,GENADR       ;CALL GENERATOR ROUTINE
 55
```

```
                        GENADR:
003320                  
003320  012700  001554'          MOV     #TAB03,R0       ;GET RM03 LIMITER AND STORAGE TABLE
003324                  1$:
003324  032767  002000  174464    BIT     #BIT10,SR1      ;CHECK SR1 FOR RANDOM ACCESS MODE
003332  001476                  BEQ     7$              ;BR IF NO
003334  104417  000000'         RANDS,BEGIN
003340  016760  174510  000016    MOV     RANNUM,CYL(R0)  ;GENERATE THE CYLINDER ADDRESS
003346  042760  176000  000016    BIC     #176000,CYL(R0) ;CHOP OFF HIGH BITS
003354  104417  000000'         RANDS,BEGIN
003360  016760  174470  000014    MOV     RANNUM,TRK(R0)  ;GENERATE THE TRACK ADDRESS
003366  042760  177400  000014    BIC     #177400,TRK(R0) ;CHOP OFF HIGH BYTE
003374  104417  000000'         RANDS,BEGIN
003400  016760  174450  000012    MOV     RANNUM,SEC(R0)  ;GENERATE THE SECTOR ADDRESS
003406  042760  177400  000012    BIC     #177400,SEC(R0) ;CHOP OFF HIGH BYTE
                        
003414  026060  000006  000016  2$:     CMP     LC(R0),CYL(R0)  ;OVER THE CYLINDER LIMIT ?
003422  103003                  BHIS    3$              ;BR IF NOT
003424  006260  000016          ASR     CYL(R0)         ;REDUCE TO HALF
003430  000771                  BR      2$              ;CHECK AGAIN
003432  026060  000002  000014  3$:     CMP     LT(R0),TRK(R0)  ;OVER THE TRACK LIMIT ?
003440  103003                  BHIS    4$              ;BR IF NOT
003442  006260  000014          ASR     TRK(R0)         ;REDUCE TO HALF
003446  000771                  BR      3$
003450  026060  000004  000012  4$:     CMP     LS(R0),SEC(R0)  ;OVER THE SECTOR LIMIT ?
003456  103003                  BHIS    5$              ;BR IF NOT
003460  006260  000012          ASR     SEC(R0)         ;REDUCE TO HALF
003464  000771                  BR      4$              ;CHECK AGAIN
003466  026060  000002  000014  5$:     CMP     LT(R0),TRK(R0)  ;TO ENSURE TRACK IN THE
003474  103005                  BHIS    6$              ;RANGE BETWEEN FT AND LT
003476  166060  000002  000014    SUB     LT(R0),TRK(R0)  ;ADJUST TO THE UPPER PORTION
003504  005360  000014          DEC     TRK(R0)
003510  026060  000014  000000  6$:     CMP     TRK(R0),FT(R0)  ;HIGH THAN THE LOW LIMITER ?
003516  103035                  BHIS    8$              ;EXIT IF SO ,ELSE
003520  066060  000000  000014    ADD     FT(R0),TRK(R0)  ;ELSE,ADJUST TO THE LOWER PORTION
003526  000757                  BR      5$              ;EXIT
                        
                        ;GENERATE DISK ADDRESS BY SEQUENTIAL ADDRESSING
003530                  7$:
003530  005260  000012          INC     SEC(R0)         ;SEQUENTIAL ACCESS TO ALL SECTORS
003534  026060  000004  000012    CMP     LS(R0),SEC(R0)  ;TIME TO ADJUST THE TRACK ADDRESS
003542  103023                  BHIS    8$              ;BR IF NOT
003544  005060  000012          CLR     SEC(R0)         ;RESET THE SECTOR ADDRESS
003550  005260  000014          INC     TRK(R0)         ;INCREMENT THE TRACK ADDRESS
003554  026060  000002  000014    CMP     LT(R0),TRK(R0)  ;OVER LIMIT ?
003562  103013                  BHIS    6$              ;BR IF NOT
003564  016060  000000  000014    MOV     FT(R0),TRK(R0)  ;RESET THE STARTING TRACK ADDRESS
003572  005260  000016          INC     CYL(R0)         ;ADJUST THE CYLINDER ADDRESS
003576  026060  000006  000016    CMP     LC(R0),CYL(R0)  ;OVER LIMIT ?
003604  103002                  BHIS    8$              ;BR IF NOT
003606  005060  000016          CLR     CYL(R0)         ;ELSE, RESET CYLINDER ADDRESS
                        
                        ;CHECK FOR MFG/USR BAD SECTOR FILE IN TRACK ADDRESS
003612                  8$:
003612  026060  000006  000016    CMP     LC(R0),CYL(R0)  ;IS IT THE LAST CYLINDER ?
003620  001004                  BNE     9$              ;BR IF NO
003622  026060  000010  000014    CMP     BST(R0),TRK(R0) ;IS IT THE BAD SECTOR TRACK (DEC 144) ?
003630  001635                  BEQ     1$              ;YES, THIS IS BAD SECTOR FILE
```



```
                                                        ;GET ANOTHER ADDRESS
003632                  9$:
003632  022700  001574'          CMP     #TAB05,R0       ;IS RM05 TABLE GENERATED YET ?
003636  001403                  BEQ     10$             ;BR IF YES
003640  012700  001574'          MOV     #TAB05,R0       ;LOAD INDEX TO RM05 TABLE
003644  000627                  BR      1$              ;AND GENERATE ADDRESSES
003646  000207                  10$:    RTS     PC
                        
                        ;THIS ROUTINE IS USED TO LOAD THE SECTOR, TRACK AND CYLINDER ADDRESSES
                        ;FOR THE DEVICE UNDER TEST.
                        ;
                        ;CALL
                        ;       JSR     PC,LDADR        ;CALL LOAD ADDRESS ROUTINE
                        
003650  012700  001554'          LDADR:  MOV     #TAB03,R0       ;ASSUME RM03/2 TABLE
003654  022767  000007  174532    CMP     #7,DRVTYP       ;IS THIS AN RM05 ?
003662  001002                  BNE     1$              ;BR IF NO
003664  012700  001574'          MOV     #TAB05,R0       ;GET RM05 TABLE
003670  116067  000012  174472  1$:     MOVB    SEC(R0),DSKADR  ;LOAD SECTOR ADDRESS
003676  116067  000014  174465          MOVB    TRK(R0),DSKADR+1        ;LOAD TRACK ADDRESS
003704  016067  000016  174454          MOV     CYL(R0),CYLIND  ;LOAD CYLINDER ADDRESS
003712  000207                  RTS     PC              ;EXIT
                        
                        ;---------------------------
                        ;       JSR     PC,GETDRV       ;GO GET A DRIVE
                        ;       RET                     ;EXIT ADDRESS IF NO DRIVE IS AVAIL
                        ;       RET                     ;NORMAL RET
                        
003714                  GETDRV:
003714  005267  174454  1$:     INC     UNITNO          ;POINT TO NEXT DRIVE
003720  026727  174450  000010          CMP     UNITNO,#8.              ;DONE LOOKING?
003726  002010                  BGE     4$              ;EXIT IF ALL DRIVE SELECTED
003730  016700  174440  2$:     MOV     UNITNO,R0       ;USE AS AN INDEX
003734  136067  003752' 174430          BITB    BITTAB(R0),DVICE        ;TEST THIS DEVICE?
003742  001764                  BEQ     1$              ;LOOP BACK , IF NOT SELECTED
003744  062716  000002          3$:     ADD     #2,(SP)         ;ADJUST FOR DRIVE AVAILABLE RETURN
003750                  4$:
003750  000207                  RTS     PC
                        
003752  001001          BITTAB: 1001
003754  004004                  4004
003756  020020                  20020
003760  100100                  100100
                        
                        ;---------------------------
                        
003762  012701  000001  DROP:   MOV     #1,R1           ;INITIALIZE DROP PICKER
003766  016700  174402          MOV     UNITNO,R0       ;GET THE DRIVE NUMBER
003772  001403                  BEQ     2$              ;IF DRIVE 0 GO DROP IT
003774  006301                  1$:     ASL     R1              ;POINT TO NEXT DRIVE
003776  005300                  DEC     R0              ;IS THIS THE ONE ?
004000  001375                  BNE     1$              ;NO, LOOK AGAIN
004002  040167  174364          2$:     BIC     R1,DVICE        ;DROP THE DRIVE
                                ;*****************************************************************
                                                        ;CONVERT UNITNO TO ASCII AND
                                                        ;STORE AT ADR1
004006  104420  000000' 000374'          OTOA$,BEGIN,UNITNO,ADR1
```

```
                                        ;*****************************************************************
  167 004016  000207                    RTS     PC              ;RETURN
  168
```



```
    1                                   .SBTTL  ERROR HANDLER ROUTINE
    2
    3                                   ;CALL:
    4                                   ;       JSR     R5,ERRORS       ;CHECK IF ANY ERROR ON RM/RM
    5                                   ;       RET                     ;ERROR RET
    6                                   ;       RET2                    ;NORMAL RET
    7
    8 004020                            ERRORS:
    9 004020  005067  174336                    CLR     DRPDRV          ;CLEAR DROP DRIVE FLAG
   10 004024  005067  174346                    CLR     BADSEC          ;CLEAR THE BAD SECTOR FLAG
   11 004030  004767  001600                    JSR     PC,ERSUB1       ;SET UP FOR ERROR REPORT
   12 004034  004767  001442                    JSR     PC,GET          ;READ AND STORE ALL REGISTERS
   13 004040  016700  175550                    MOV     S+NCS1,R0       ;GET CONTROL STATUS
   14 004044  042700  177576                    BIC     #^C<RDY!GO>,R0
   15 004050  022700  000200                    CMP     #RDY,R0         ;RDY = 1 AND GO = 0 ?
   16 004054  001416                            BEQ     1$              ;BR IF YES
   17 004056  012767  000001  174276            MOV     #1,DRPDRV       ;SET DROP DRIVE FLAG
   18 004064  104403  000000' 007164'          MSGNS,BEGIN,NOTRDY      ;ASCII MESSAGE CALL WITH COMMON HEADER
   19 004072  012767  000003  174006            MOV     #3,ERRTYP       ;SAVE THE ERROR TYPE
   20                                           ;*****************************************************************
      004100  104405  000000' 001416'          HRDERS,BEGIN,TABLE      ;CONTROLLER NOT READY
                                               ;*****************************************************************
   21 004106  000167  001116                    JMP     33$
   22
   23 004112  016700  175510            1$:     MOV     S+NDS,R0        ;GET DRIVE STATUS
   24 004116  005100                            COM     R0
   25 004120  032700  010700                    BIT     #DRY!DPR!VV!MOL,R0      ;DRIVE READY ?
   26 004124  001416                            BEQ     2$              ;BR IF YES
   27 004126  012767  000001  174226            MOV     #1,DRPDRV       ;SET DROP DRIVE FLAG
   28 004134  104403  000000' 007154'          MSGNS,BEGIN,NODRDY      ;ASCII MESSAGE CALL WITH COMMON HEADER
   29 004142  012767  000006  173736            MOV     #6,ERRTYP       ;SAVE THE ERROR TYPE
   30                                           ;*****************************************************************
      004150  104405  000000' 001416'          HRDERS,BEGIN,TABLE      ;DRIVE NOT READY
                                               ;*****************************************************************
   31 004156  000167  001046                    JMP     33$
   32
   33 004162  016700  175426            2$:     MOV     S+NCS1,R0       ;CHECK IF SEEK COMMAND IS EXECUTED
   34 004166  042700  177701                    BIC     #177701,R0      ;LEFT ONLY FUNCTION BITS
   35 004172  022700  000004                    CMP     #4,R0           ;A SEEK COMMAND ?
   36 004176  001023                            BNE     3$              ;BR IF NOT
   37 004200  016700  175422                    MOV     S+NDS,R0        ;CHECK THE STATUS
   38 004204  042700  057777                    BIC     #^C<ATA!PIP>,R0
   39 004210  022700  100000                    CMP     #ATA,R0         ;ATA = 1 AND PIP = 0 ?
   40 004214  001423                            BEQ     4$              ;BR IF YES
   41 004216  012777  000013  175250            MOV     #13,@RMCS1      ;RELEASE THE DRIVE
   42 004224  104407  000000'                  BREAKS,BEGIN            ;TEMPORARY RETURN TO MONITOR....
      004230  104407  000000'                  BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
   43 004234  104403  000000' 007100'          MSGNS,BEGIN,SEEKER      ;ASCII MESSAGE CALL WITH COMMON HEADER
   44 004242  000167  000746                    JMP     32$             ;EXIT
   45
   46 004246  012777  000013  175220    3$:     MOV     #13,@RMCS1      ;RELEASE THE DRIVE
   47 004254  104407  000000'                  BREAKS,BEGIN            ;TEMPORARY RETURN TO MONITOR....
      004260  104407  000000'                  BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
   48
   49 004264  032767  060000  175322    4$:     BIT     #MCPE!ITRE,S+NCS1       ;ANY RH ERROR DETECTED ?
   50 004272  001010                            BNE     5$              ;BR IS SO
   51 004274  032767  040000  175324            BIT     #ERR,S+NDS      ;ANY DRIVE ERROR ?
```

```
 52 004302  001004                          BNE     5$              ;BR IS SO
 53 004304  062705  000002                  ADD     #2,R5           ;ELSE,ADJUST FOR GOOD RETURN
 54 004310  000167  000714                  JMP     33$             ;EXIT
 55
 56                                 ;CHECK TO SEE IF USER DEFINED ANY BAD SPOTS
 57 004314                          5$:                             ;CHECK IF ON THE BAD SPOT ?
 58 004314  012700  000252'                 MOV     #BADSPT,R0      ;TABLE ADDRESS
 59 004320  012701  000020                  MOV     #16.,R1         ;TOTAL BAD SECTOR COUNT
 60 004324  021067  174036          6$:     CMP     (R0),CYLIND     ;ON THE SAME CYLINDER ?
 61 004330  001014                          BNE     7$              ;BR IF NOT
 62 004332  126067  000003  174031          CMPB    3(R0),DSKADR+1  ;ON THE SAME TRACK ?
 63 004340  001010                          BNE     7$              ;BR IF NOT
 64 004342  126067  000002  174020          CMPB    2(R0),DSKADR    ;ON THE SAME  SECTOR ?
 65 004350  001004                          BNE     7$              ;BR IF NOT
 66 004352  012767  177777  174016          MOV     #-1,BADSEC      ;SET USR BAD SPOT FLAG
 67 004360  000423                          BR      9$              ;EXIT
 68 004362  062700  000004          7$:     ADD     #4,R0           ;ADJUST TABLE ADDRESS
 69 004366  005301                          DEC     R1              ;DECREMENT ONE BSE COUNT
 70 004370  001355                          BNE     6$              ;BR IF NOT END OF TABLE
 71
 72                                 ;CHECK FOR HARDWARE DETECTED BAD SPOTS
 73 004372                          8$:
 74 004372  022767  000200  175230          CMP     #HCE,S+NER1     ;IS IT ONLY A HEADER COMPARE ERROR ?
 75 004400  001415                          BEQ     10$             ;BR IF YES
 76 004402  032767  000400  175220          BIT     #HCRC,S+NER1    ;HEADER CRC ERROR ?
 77 004410  001011                          BNE     10$             ;BR IF YES
 78 004412  022767  100000  175236          CMP     #BSE,S+NER2     ;IS IT A HARDWARE DETECTED ERROR ?
 79 004420  001005                          BNE     10$             ;BR IF NO
 80 004422  016767  175230  173746          MOV     S+NER2,BADSEC   ;GET RMER2 FOR BSE
 81
 82 004430  000167  000574          9$:     JMP     33$             ;TAKE ERROR EXIT , BUT NOT REPORT THE ERROR
 83
 84                                 ;CHECK FOR OTHER ERRORS
 85 004434                          10$:
 86 004434  032767  020000  175152          BIT     #MCPE,S+NCS1    ;RH CONTROL BUS PARITY ERROR ?
 87 004442  001405                          BEQ     11$             ;BR IF NOT
 88 004444  104403  000000' 007040'         MSGN$,BEGIN,MCPERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
 89 004452  000167  000536                  JMP     32$             ;EXIT
 90 004456  032767  000400  175140  11$:    BIT     #MDPE,S+NCS2    ;MASS BUS DATA PARITY ERROR ?
 91 004464  001405                          BEQ     12$             ;BR IF NOT
 92 004466  104403  000000' 007044'         MSGN$,BEGIN,MDPERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
 93 004474  000167  000514                  JMP     32$             ;EXIT
 94 004500  032767  037000  175116  12$:    BIT     #MXF!PGE!NEM!NED!UPE,S+NCS2     ;RH CONTROLL FAILS ?
 95 004506  001411                          BEQ     13$             ;BR IF NOT
 96 004510  032767  040000  175110          BIT     #ERR,S+NDS      ;ANY DRIVE ERROR ?
 97 004516  001005                          BNE     13$             ;BR IF SO
 98 004520  104403  000000' 007054'         MSGN$,BEGIN,RHFAIL      ;ASCII MESSAGE CALL WITH COMMON HEADER
 99 004526  000167  000462                  JMP     32$             ;EXIT
100 004532  032767  100000  175064  13$:    BIT     #DTE,S+NCS2     ;DATA LATE ERROR ?
101 004540  001423                          BEQ     15$             ;BR IF NOT
102 004542  005267  173632                  INC     DLTCNT          ;INCREMENT DATA LATE ERROR COUNT
103 004546  032767  000004  173242          BIT     #BIT2,SR1       ;ALLOW TO TYPE OUT DATA LATE ERROR ?
104 004554  001402                          BEQ     14$             ;BR IF SO
105 004556  000167  000446                  JMP     33$             ;EXIT
106 004562                          14$:
107 004562  104403  000000' 007150'         MSGN$,BEGIN,DLTERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
108 004570  012767  000002  173310          MOV     #2,ERRTYP       ;SAVE THE ERROR TYPE
```



```
109                                 ;*************************************************************
    004576  104406  000000' 00141e'         SOFERS,BEGIN,TABLE      ;DATA LATE ERROR
                                    ;*************************************************************
110 004604  000167  000420                  JMP     33$             ;EXIT
111 004610  032767  040000  175010  15$:    BIT     #ERR,S+NDS      ;ANY DRIVE ERROR ?
112 004616  001027                          BNE     18$             ;BR IF ANY
113 004620  032767  040000  174776          BIT     #WCE,S+NCS2     ;RH WRITE CHECK ERROR ?
114 004626  001404                          BEQ     16$             ;BR IF NOT
115 004630  104403  000000' 007134'         MSGN$,BEGIN,WRTCK       ;ASCII MESSAGE CALL WITH COMMON HEADER
116 004636  000566                          BR      32$             ;EXIT
117 004640  032767  040000  174746  16$:    BIT     #TRE,S+NCS1     ;THEN RH TRANSFER ERROR ?
118 004646  001404                          BEQ     17$             ;BR IF NOT
119 004650  104403  000000' 007034'         MSGN$,BEGIN,IHERR       ;ASCII MESSAGE CALL WITH COMMON HEADER
120 004656  000556                          BR      32$             ;EXIT
121 004660  012767  000000  173220  17$:    MOV     #0,ERRTYP       ;UNKNOWN ERROR
122                                 ;*************************************************************
    004666  104405  000000' 001416'         HRDERS,BEGIN,TABLE      ;UNKNOWN ERROR
                                    ;*************************************************************
123 004674  000555                          BR      33$             ;EXIT
124 004676  032767  040000  174724  18$:    BIT     #UNS,S+NER1     ;RM DEVICE UNSAFE ?
125 004704  001404                          BEQ     19$             ;BR IF NOT
126 004706  104403  000000' 007060'         MSGN$,BEGIN,UNSAFE      ;ASCII MESSAGE CALL WITH COMMON HEADER
127 004714  000537                          BR      32$             ;EXIT
128 004716  032767  000600  174704  19$:    BIT     #HCRC!HCE,S+NER1        ;HEADER INFORMATION ERROR ?
129 004724  001404                          BEQ     20$             ;BR IF NOT
130 004726  104403  000000' 007064'         MSGN$,BEGIN,HEADER      ;ASCII MESSAGE CALL WITH COMMON HEADER
131 004734  000527                          BR      32$             ;EXIT
132 004736  032767  000020  174664  20$:    BIT     #FER,S+NER1     ;DATA FORMAT ERROR ?
133 004744  001404                          BEQ     21$             ;BR IF NOT
134 004746  104403  000000' 007070'         MSGN$,BEGIN,FORMAT      ;ASCII MESSAGE CALL WITH COMMON HEADER
135 004754  000517                          BR      32$             ;EXIT
136 004756  032767  003000  174644  21$:    BIT     #IAE!AOE,S+NER1 ;ADDRESSING ERROR ?
137 004764  001410                          BEQ     22$             ;BR IF NOT
138 004766  032767  002000  174632          BIT     #LBT,S+NDS      ;LAST BLOCK TRANSFER SET ?
139 004774  001115                          BNE     33$             ;EXIT, IF SO
140 004776  104403  000000' 007074'         MSGN$,BEGIN,ADDRES      ;ASCII MESSAGE CALL WITH COMMON HEADER
141 005004  000503                          BR      32$             ;EXIT
142 005006  032767  020000  174614  22$:    BIT     #UPI,S+NER1     ;POSITIONER FAILURE ?
143 005014  001404                          BEQ     23$             ;BR IF NOT
144 005016  104403  000000' 007144'         MSGN$,BEGIN,POSIT       ;ASCII MESSAGE CALL WITH COMMON HEADER
145 005024  000473                          BR      32$             ;EXIT
146 005026  032767  040000  174622  23$:    BIT     #SKI,S+NER2     ;SEEK ERROR ?
147 005034  001404                          BEQ     24$             ;BR IF NOT
148 005036  104403  000000' 007100'         MSGN$,BEGIN,SEEKER      ;ASCII MESSAGE CALL WITH COMMON HEADER
149 005044  000463                          BR      32$             ;EXIT
150 005046  032767  004000  174602  24$:    BIT     #LSC,S+NER2     ;LOSS OF SYSTEM CLOCK ?
151 005054  001403                          BEQ     25$             ;BR IF NOT
152 005056  104403  000000' 007104'         MSGN$,BEGIN,LOSYSC      ;ASCII MESSAGE CALL WITH COMMON HEADER
153 005064  032767  004000  174536  25$:    BIT     #WLE,S+NER1     ;WRITE LOCK ERROR ?
154 005072  001404                          BEQ     26$             ;BR IF NOT
155 005074  104403  000000' 007110'         MSGN$,BEGIN,WTLOCK      ;ASCII MESSAGE CALL WITH COMMON HEADER
156 005102  000444                          BR      32$             ;EXIT
157 005104  032767  010000  174516  26$:    BIT     #DTE,S+NER1     ;DRIVE TIMING ERROR ?
158 005112  001404                          BEQ     27$             ;BR IF NOT
159 005114  104403  000000' 007140'         MSGN$,BEGIN,TIMERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
160 005122  000434                          BR      32$             ;EXIT
161 005124  032767  100100  174476  27$:    BIT     #DCK!ECH,S+NER1 ;DATA CHECK ERROR ?
```

```
 162 005132  001404                        BEQ     28$             ;BR IF NO1
 163 005134  104403  000000' 007114'       MSGN$,BEGIN,DATERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
 164 005142  000424                        BR      32$             ;EXIT
 165 005144  032767  002000  174504  28$:  BIT     #LBC,S+NER2     ;LOST BIT CLOCK ?
 166 005152  001404                        BEQ     29$             ;BR IF NOT
 167 005154  104403  000000' 007120'       MSGN$,BEGIN,LOSBIT      ;ASCII MESSAGE CALL WITH COMMON HEADER
 168 005162  000414                        BR      32$             ;EXIT
 169 005164  032767  000040  174436  29$:  BIT     #WCF,S+NER1     ;WRITE CLOCK ERROR ?
 170 005172  001404                        BEQ     30$             ;BR IF NOT
 171 005174  104403  000000' 007124'       MSGN$,BEGIN,WTCLCK      ;ASCII MESSAGE CALL WITH COMMON HEADER
 172 005202  000404                        BR      32$             ;EXIT
 173 005204                          30$:
     005204  104403  000000' 007130'       MSGN$,BEGIN,DRVERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
 174 005212  000400                        BR      32$             ;EXIT
 175 005214                  ,       31$:                        ;
 176 005214                          32$:
 177 005214  012767  000000  172664        MOV     #0,ERRTYP       ;DUMMY ERROR NUMBER
 178                                       ;*****************************************************************
     005222  104406  000000' 001416'       SUPER$,BEGIN,TABLE      ;DUMP RH/RM REGISTERS
                                           ;*****************************************************************
 179 005230                          33$:
 180 005230  000205                        RTS     R5              ;EXIT
 181
 182                                 ;THIS ROUTINE IS USED TO WAIT FOR "DVA" TO SET, CLEAR THE MASSBUS
 183                                 ;AND SELECT THE DEVICE.
 184                                 ;
 185                                 ;CALL
 186                                 ;       JSR     R5,READY        ;CALL READY ROUTINE
 187                                 ;       BR      ?               ;ERROR RETURN
 188                                 ;       RETURN                  ;NORMAL EXIT
 189
 190 005232  004767  000136  READY:  JSR     PC,WAIT         ;WAIT FOR "DVA" BIT
 191 005236  000436                        BR      1$              ;EXIT ON ERROR
 192 005240  017700  174256                MOV     @RMDT,R0        ;READ THE DRIVE TYPE
 193 005244  010067  173144                MOV     R0,DRVTYP       ;GET DRIVE TYPE AND
 194 005250  042767  177770  173136        BIC     #^C7,DRVTYP     ;SAVE TYPE BITS
 195 005256  022700  024025                CMP     #24025,R0       ;DUAL PORT RM02 ?
 196 005262  001442                        BEQ     2$              ;BR IS SO
 197 005264  022700  020025                CMP     #20025,R0       ;SINGLE PORT RM02 ?
 198 005270  001437                        BEQ     2$              ;BR IF SO
 199 005272  022700  024024                CMP     #24024,R0       ;DUAL PORT RM03 ?
 200 005276  001434                        BEQ     2$              ;BR IF YES
 201 005300  022700  020024                CMP     #20024,R0       ;SINGLE PORT RM03 ?
 202 005304  001431                        BEQ     2$              ;BR IF YES
 203 005306  022700  024027                CMP     #24027,R0       ;DUAL PORT RM05 ?
 204 005312  001426                        BEQ     2$              ;BR IF YES
 205 005314  022700  020027                CMP     #20027,R0       ;SINGLE PORT RM05 ?
 206 005320  001423                        BEQ     2$              ;BR IF YES
 207 005322  004767  000306                JSR     PC,ERSUB1       ;SETUP FOR ERROR REPORT
 208 005326  104403  000000' 007160'       MSGN$,BEGIN,DEVNO1      ;ASCII MESSAGE CALL WITH COMMON HEADER
 209 005334                          1$:
 210 005334  004767  000142                JSR     PC,GET          ;READ AND STORE ALL REGISTERS
 211 005340  012767  000006  172540        MOV     #6,ERRTYP       ;SAVE THE ERROR TYPE
 212                                       ;*****************************************************************
     005346  104405  000000' 001416'       HRDER$,BEGIN,TABLE      ;DEVICE NOT AVAILABLE OR NOT AN RM05/3/2
                                           ;*****************************************************************
 213 005354  004767  176402                JSR     PC,DROP         ;DROP THE DRIVE
```



```
 214 005360  104403  000000' 007174'       MSGN$,BEGIN,DRM ;ASCII MESSAGE CALL WITH COMMON HEADER
 215 005366  000401                        BR      3$              ;EXIT WITH ERROR
 216 005370  005725                  2$:   TST     (R5)+           ;DRIVE AVAILABLE, ON-LINE, SAFE AND NO ERRORS
 217 005372  000205                  3$:   RTS     R5              ;NORMAL EXIT
 218
 219                                 ;THIS ROUTINE WAITS FOR "DVA" TO SET IN THE RMCS1 REGISTER. IF THE "DVA"
 220                                 ;BIT SETS, THE MBA IS CLEARED AND THE DRIVE IS SELECTED. IF THE "DVA" BIT
 221                                 ;DOES NOT SET WITHIN A CERTAIN PERIOD OF TIME,  AN ERROR IS REPORTED AND
 222                                 ;RETURN IS MADE BACK TO THE MAIN PROGRAM, WHERE THE DRIVE WILL BE DROPPED
 223                                 ;FROM THE TEST.
 224                                 ;
 225                                 ;CALL
 226                                 ;       JSR     PC,WAIT         ;CALL WAIT ROUTINE
 227                                 ;       BR      ???             ;ERROR RETURN
 228                                 ;       RETURN                  ;NORMAL RETURN
 229
 230 005374  016777  172774  174102  WAIT:   MOV     UNITNO,@RMCS2   ;SELECT DRIVE ADDRESS
 231 005402  012767  017777  173000          MOV     #17777,CLK      ;CLOCK COUNT
 232 005410                          1$:
 233 005410  032777  004000  174056          BIT     #DVA,@RMCS1     ;IS DRIVE AVAILABLE ?
 234 005416  001020                        BNE     2$              ;BR IF YES
 235 005420  104407  000000'               BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
     005424  104407  000000'               BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
 236 005430  005367  172754                DEC     CLK             ;DECREMENT CLOCK COUNT
 237 005434  001365                        BNE     1$
 238 005436  004767  000172                JSR     PC,ERSUB1       ;SETUP FOR ERROR REPORT
 239 005442  012767  000000  172436        MOV     #0,ERRTYP       ;UNKNOWN ERROR TYPE
 240 005450  104403  000000' 007170'       MSGN$,BEGIN,NODVA       ;ASCII MESSAGE CALL WITH COMMON HEADER
 241 005456  000410                        BR      3$              ;TIMEOUT ON "DVA" BIT
 242 005460                          2$:
 243 005460  012777  000040  174016        MOV     #CLR,@RMCS2     ;ISSUE A CLEAR COMMAND
 244 005466  016777  172702  174010        MOV     UNITNO,@RMCS2   ;SELECT THE DRIVE ADDRESS
 245 005474  062716  000002                ADD     #2,(SP)         ;ADJUST FOR GOOD RETURN
 246 005500  000207                  3$:   RTS     PC              ;RETURN
 247
 248                                 ;-------------------------
 249                                 ;THIS ROUTINE IS USED TO READ AND STORE ALL REGISTERS FROM THE RH/RM
 250                                 ;WHEN AN ERROR HAS BEEN DETECTED.
 251                                 ;
 252                                 ;CALL
 253                                 ;       JSR     PC,GET          ;CALL GET ROUTINE
 254
 255 005502  005000                  GET:    CLR     R0              ;START WITH RMCS1 REGISTER INDEX 0
 256 005504  016701  173764                  MOV     RMCS1,R1        ;RH11/RH70 BUS ADDRESS
 257 005510  022701  001516'         1$:     CMP     #RMDB,R1        ;IS IT RMDB REGISTER ?
 258 005514  001006                        BNE     2$              ;BR IF NO
 259 005516  032711  000200                BIT     #DR,(R1)        ;IS "DR" BIT SET ?
 260 005522  001003                        BNE     2$              ;BR IF YES
 261 005524  005060  001614'               CLR     S(R0)           ;ELSE, CLEAR RMDB REGISTER
 262 005530  000402                        BR      3$
 263 005532  012160  001614'         2$:   MOV     (R1)+,S(R0)     ;READ ONE REGISTER
 264 005536  062700  000002          3$:   ADD     #2,R0           ;ADJUST TABLE INDEX
 265 005542  022700  000050                CMP     #50,R0          ;DONE READING ALL REGISTERS ?
 266 005546  001360                        BNE     1$              ;BR IF NO
 267 005550  012760  177777  001416'       MOV     #-1,TABLE(R0)   ;TERMINATE DECX REGISTER TABLE FOR AN RH11
 268 005556  005767  172602                TST     FLAG70          ;CHECK FOR ADDITIONAL RH70 REGISTERS
 269 005562  100011                        BPL     4$              ;BR IF NONE
```

```
 270 005564  012760  001664' 001416'          MOV     #S+NBAE,TABLE(R0)       ;EXTEND THE DECX REGISTER TABLE
 271 005572  012160  001614'                  MOV     (R1)+,S(R0)     ;GET RMBAE REGISTER
 272 005576  062700  000002                   ADD     #2,R0           ;ADJUST INDEX TABLE
 273 005602  011160  001614'                  MOV     (R1),S(R0)      ;GET RMCS3 REGISTER
 274 005606  000207                   4$:     RTS     PC              ;RETURN
 275
 276                                  ;-------------------------
 277
 278 005610  014167  172272           ERSUB2: MOV     -(R1),ASB       ;LOAD THE DATA
 279 005614  010167  172262                   MOV     R1,SBADR        ;LOAD ADDRESS OF DATA WRITTEN
 280 005620  014267  172264                   MOV     -(R2),AWAS      ;LOAD THE DATA
 281 005624  010267  172254                   MOV     R2,WASADR       ;LOAD ADDRESS OF DATA READ
 282 005630  005721                           TST     (R1)+           ;RESET REG. 1
 283 005632  005722                           TST     (R2)+           ;RESET REG. 2
 284
 285 005634  016767  173634  172236   ERSUB1: MOV     RMCS1,CSRA      ;LOAD ADR OF CURRENT CSR
 286 005642  017767  173626  172232           MOV     @RMCS1,ACSR     ;LOAD CONTENTS OF CURRENT CSR
 287 005650  000207                           RTS     PC              ;RETURN
 288
 289                                  ;-------------------------
 290
 291 005652  005067  172506           SETUP:  CLR     FLAG70          ;CLEAR 11/70 FLAG
 292 005656  016700  172124                   MOV     ADDR,R0         ;GET DEVICE ADDRESS (DEFAULT = 176700)
 293 005662  012701  001474'                  MOV     #RMCS1,R1       ;GET STARTING REGISTER ADDRESS
 294 005666  010021                   1$:     MOV     R0,(R1)+        ;GENERATE REGISTER ADDRESS
 295 005670  062700  000002                   ADD     #2,R0
 296 005674  022701  001544'                  CMP     #RMBAE,R1       ;LAST REGISTER FOR RH11 ?
 297 005700  001372                           BNE     1$              ;BR IF NO
 298 005702  052777  002000  173564           BIS     #BIT10,@RMCS1   ;TRY SETTING "PSEL" BIT
 299 005710  032777  002000  173556           BIT     #BIT10,@RMCS1   ;IS CPU AN 11/70 ?
 300 005716  001015                           BNE     3$              ;BR IF NO
 301 005720  052767  100000  172436           BIS     #BIT15,FLAG70   ;SET 11/70 FLAG
 302 005726  032767  100000  172062           BIT     #BIT15,SR1      ;SPECIFY 32 REGISTER ON RH70 ?
 303 005734  001402                           BEQ     2$              ;BR IF NO
 304 005736  062700  000024                   ADD     #24,R0          ;YES, ADJUST THE RMBAE ADDRESS
 305 005742  010021                   2$:     MOV     R0,(R1)+        ;SETUP RMBAE REGISTER
 306 005744  062700  000002                   ADD     #2,R0           ;ADJUST INDEX
 307 005750  010011                           MOV     R0,(R1)         ;SETUP RMCS3 REGISTER
 308 005752                           3$:
 309 005752  016700  172032                   MOV     VECTOR,R0       ;GET VECTOR ADDRESS
 310 005756  012720  003204'                  MOV     #NTRUPT,(R0)+   ;SET POINTER JUST IN CASE
 311 005762  116710  172024                   MOVB    BR1,(R0)        ;SET PRIORITY
 312 005766  000207                           RTS     PC              ;RETURN
 313
 314                                  .SBTTL  ASCIZ MESSAGE
 315
 316 005770      045     040     122  MES1:   .ASCIZ  '% RH11/RH70 TRANSFER ERROR'
 317 006023      045     040     115  MES2:   .ASCIZ  '% MASSBUS PARITY ERROR MCPE=1'
 318 006061      045     040     115  MES3:   .ASCIZ  '% MASSBUS DATA PARITY ERROR MDPE=1'
 319 006124      040     104     122  MES4:   .ASCIZ  ' DRIVE '
 320 006134      040     104     122  MES5:   .ASCIZ  ' DROPPED%'
 321 006146      045     040     122  MES6:   .ASCIZ  '% RETRY EXCEEDED'
 322 006167      045     040     122  MES10:  .ASCIZ  '% RM DEVICE UNSAFE'
 323 006212      045     040     110  MES11:  .ASCIZ  '% HEADER INFORMATION ERROR'
 324 006245      045     040     104  MES12:  .ASCIZ  '% DATA FORMAT BIT ERROR'
 325 006275      045     040     104  MES14:  .ASCIZ  '% DRIVE ERROR'
 326 006313      045     040     122  MES15:  .ASCIZ  '% RH11/RH70 CONTROLLER ERROR'
```



```
 327 006350      045     040     101  MES16:  .ASCIZ  '% ADDRESSING ERROR'
 328 006373      045     040     123  MES17:  .ASCIZ  '% SEEK ERROR'
 329 006410      045     040     114  MES20:  .ASCIZ  '% LOST SYSTEM CLOCK'
 330 006434      045     040     127  MES21:  .ASCIZ  '% WRITE LOCK ERROR'
 331 006457      045     040     104  MES22:  .ASCIZ  '% DATA CHECK ERROR'
 332 006502      045     040     114  MES23:  .ASCIZ  '% LOST BIT CLOCK'
 333 006523      045     040     127  MES24:  .ASCIZ  '% WRITE CLOCK ERROR'
 334 006547      045     040     122  MES25:  .ASCIZ  '% RH11/RH70 WRITE CHECK ERROR'
 335 006605      045     040     104  MES26:  .ASCIZ  '% DRIVE TIMING ERROR'
 336 006632      045     040     000  MES27:  .ASCIZ  '% '
 337 006635      045     040     120  MES30:  .ASCIZ  '% POSITIONING ERROR'
 338 006661      045     040     104  MES31:  .ASCIZ  '% DATA LATE ERROR'
 339 006703      045     040     104  MES32:  .ASCIZ  '% DRIVE NOT READY'
 340 006725      045     040     104  MES33:  .ASCIZ  '% DEVICE NOT AN RM05/3/2'
 341 006756      045     040     103  MES34:  .ASCIZ  '% CONTROLLER NOT READY'
 342 007005      045     040     104  MES35:  .ASCIZ  '% DRIVE NOT AVAILABLE'
 343
 344                                  .EVEN
 345 007034  005770'                   TRERR:  MES1
 346 007036  177777                           177777
 347 007040  006023'                   MCPERR: MES2
 348 007042  177777                           177777
 349 007044  006061'                   MDPEPR: MES3
 350 007046  177777                           177777
 351 007050  006146'                   EXCED:  MES6
 352 007052  177777                           177777
 353 007054  006313'                   RHFAIL: MES15
 354 007056  177777                           177777
 355 007060  006167'                   UNSAFE: MES10
 356 007062  177777                           177777
 357 007064  006212'                   HEADER: MES11
 358 007066  177777                           177777
 359 007070  006245'                   FORMAT: MES12
 360 007072  177777                           177777
 361 007074  006350'                   ADDRES: MES16
 362 007076  177777                           177777
 363 007100  006373'                   SEEKER: MES17
 364 007102  177777                           177777
 365 007104  006410'                   LOSYSC: MES20
 366 007106  177777                           177777
 367 007110  006434'                   WILOCK: MES21
 368 007112  177777                           177777
 369 007114  006457'                   DATERR: MES22
 370 007116  177777                           177777
 371 007120  006502'                   LOSBIT: MES23
 372 007122  177777                           177777
 373 007124  006523'                   WICLCK: MES24
 374 007126  177777                           177777
 375 007130  006275'                   DRVERR: MES14
 376 007132  177777                           177777
 377 007134  006547'                   WRTCK:  MES25
 378 007136  177777                           177777
 379 007140  006605'                   TIMERR: MES26
 380 007142  177777                           177777
 381 007144  006635'                   POSIT:  MES30
 382 007146  177777                           177777
 383 007150  006661'                   DLTERR: MES31
```

```
384 007152  177777                        177777
385 007154  006703'             NODRDY: MES32
386 007156  177777                        177777
387 007160  006725'             DEVNOT: MES33
388 007162  177777                        177777
389 007164  006756'             NOTRDY: MES34
390 007166  177777                        177777
391 007170  007005'             NODVA:  MES35
392 007172  177777                        177777
393 007174  006124'             DRM:    MES4
394 007176  007211'                     NUMB
395 007200  006134'                     MES5
396 007202  177777                     177777
397
398 007204                      ADR1:   .BLKB   5               ;STARTING OF THE ASCII NUMBER STRING
399 007211      000             NUMB:   .BYTE   0               ;LEAST SIGNIFICANT ASCII DIGIT
400 007212      000                     .BYTE   0               ;TERMINATOR OF THE ASCII STRING
401                             .EVEN
402
403         000001              .END
```



```
ACSR    000102R     DRPDRV  000362R     LDADR   003650R     NDT    = 000026     READ    002524R
ADDR    000006R     DRVERR  007130R     LOOP1   002146R     NEC1   = 000044     READHD  002744R
ADDRES  007074R     DRVTYP  000414R     LOOP2   002160R     NEC2   = 000046     READY   005232R
ADDR22= 001000      DRY    = 000200     LOSB11  007120R     NED    = 010000     RESTRT  002130R
ADR1    007204R     DSKADR  000370R     LOSYSC  007104R     NEM    = 004000     RES1    000056R
AOE    = 001000     DTE    = 010000     LS     = 000004     NER1   = 000014     RES2    000060R
ASB     000106R     DTL    = 100000     LSC    = 004000     NER2   = 000042     RHFAIL  007054R
ASTAT   000104R     DVA    = 004000     LT     = 000002     NHP    = 000036     RMAS    001512R
ATA    = 100000     DVC    = 000200     MAP22$= 104416      NLA    = 000020     RMHA    001500R
AWAS    000110R     DVICE   000372R     MCPE   = 020000     NMR1   = 000024     RMHAE   001544R
BADSEC  000376R     DVID1   000014R     MCPERR  007040R     NMR2   = 000040     RMCS1   001474R
BADSPT  000252R     EA22    000360R     MDPE   = 000400     NODRDY  007154R     RMCS2   001504R
BEGIN   000000R     ECH    = 000100     MDPERR  007044R     NODVA   007170R     RMCS3   001546R
BITTAB  003752R     ENDIT$= 104413      MES1    005770R     NOF    = 000032     RMDA    001502R
BIT0   = 000001     ENDS   = 104410     MES10   006167R     NOTRDY  007164R     RMDB    001516R
BIT1   = 000002     ERR    = 040000     MES11   006212R     NSN    = 000030     RMDC    001530R
BIT10  = 002000     ERRORS  004020R     MES12   006245R     NTRUPT  003204R     RMDS    001506R
BIT11  = 004000     ERRTYP  000106R     MES14   006275R     NULL   = 000000     RMDT    001522R
BIT12  = 010000     ERSUB1  005634R     MES15   006313R     NUMB    007211R     RMEC1   001540R
BIT13  = 020000     ERSUB2  005610R     MES16   006350R     NWC    = 000002     RMEC2   001542R
BIT14  = 040000     EXCED   007050R     MES17   006373R     OM     = 000001     RMER1   001510R
BIT15  = 100000     EXITS  = 104400     MES2    006023R     OPEN   = 000000     RMER2   001536R
BIT2   = 000004     FER    = 000020     MES20   006410R     OPI    = 020000     RMHR    001532R
BIT3   = 000010     FERADR  000406R     MES21   006434R     OR     = 000200     RMLA    001514R
BIT4   = 000020     FLAG70  000364R     MES22   006457R     OTOA$  = 104420     RMMR1   001520R
BIT5   = 000040     FMT    = 010000     MES23   006502R     PAR    = 000010     RMMR2   001534R
BIT6   = 000100     FORMAT  007070R     MES24   006523R     PASCNT  000034R     RMOF    001526R
BIT7   = 000200     FREE    000150R     MES25   006547R     PA18    000352R     RMR    = 000004
BIT8   = 000400     FT     = 000000     MES26   006605R     PA22    000356R     RMSN    001524R
BIT9   = 001000     FUNC    000402R     MES27   006632R     PGE    = 002000     RMWC    001476R
BREAK$= 104407      GENADR  003320R     MES3    006061R     PGM    = 001000     RSTR1   000112R
BR1     000012R     GET     005502R     MES30   006635R     PIP    = 020000     RSTRT1  002140R
BR2     000013R     GETDRV  003714R     MES31   006661R     PIRQS  = 000004     R6     =%000006
BSE    = 100000     GETPAS= 104415      MES32   006703R     PUPSP  = 005726     R7     =%000007
BST    = 000010     GO     = 000001     MES33   006725R     PUPSP2= 022626      S       001614R
BTODS  = 104421     GO1     003044R     MES34   006756R     POSIT   007144R     SBADR   000102R
CDATAS= 104412      GO2     003150R     MES35   007005R     PRTY   = 000000     SC     = 100000
CDERCT  000144R     GWBUFS= 104414      MES4    006124R     PRTY0  = 000000     SEC    = 000012
CDWDCT  000146R     HCE    = 000200     MES5    006134R     PRTY1  = 000040     SEEK    003024R
CLK     000410R     HCRC   = 000400     MES6    006146R     PRTY2  = 000100     SEEKER  007100R
CLR    = 000040     HEADER  007064R     MODNAM  000000R     PRTY3  = 000140     SETUP   005652R
CONFIG  000056R     HRDCNT  000044R     MODSP   000252R     PRTY4  = 000200     SKI    = 040000
CSRA    000100R     HRDER$= 104405      MUL    = 010000     PRTY5  = 000240     SUFCNT  000042R
CYCLE   002226R     HRDPAS  000050R     MSGNS  = 104403     PRTY6  = 000300     SOFERS= 104406
CYL    = 000016     IAE    = 002000     MSGSS  = 104402     PRTY7  = 000340     SOFPAS  000046R
CYLIND  000366R     ICONT   000036R     MSGS   = 104401     PS     = 177776     SPOINT  000032R
DATCK$= 104411      ICOUNT  000040R     MXF    = 001000     PSW    = 177776     SPSIZ  = 000040
DATERR  007114R     IDNUM   000122R     NAS    = 000016     PUSH   = 005746     SR1     000016R
DATERS= 104404      ILF    = 000001     NBA    = 000004     PUSH2  = 024646     SR2     000020R
DCK    = 100000     ILR    = 000002     NBAE   = 000050     RANDS  = 104417     SR3     000022R
DEVNOT  007160R     IMODX.= 000000      NCS1   = 000000     RANNUM  000054R     SR4     000024R
DLTCNT  000400R     INIT    000030R     NCS2   = 000010     RBUF    000416R     START   001670R
DLTERR  007150R     INTR    000120R     NCS3   = 000052     RBUFEA  000130R     STAT    000026R
DPE    = 000010     IR     = 000100     NDA    = 000006     RBUFPA  000126R     SVR0    000062R
DPR    = 000400     LBC    = 002000     NDB    = 000022     RBUFSZ  000132R     SVP1    000064R
DRM     007174R     LBT    = 002000     NDC    = 000034     RBUFVA  000124R     SVR2    000066R
DROP    003762R     LC     = 000006     NDS    = 000012     RDY    = 000200     SVP3    000070R
```

```
SVR4    000072R         TMERR   007034R         VV     = 000100         wCF    = 000040         wPL    = 004000
SVR5    000074R         TRK     = 000014        wA11    005374R         wDFR    000116R         wRTCK   007134R
SVR6    000076R         TRPDFD= 000022          wASADR  000104R         wDTO    000114R         wTCLCK  007124R
SYSCNT  000052R         TRY     000412R         wBUFEA  000136R         wLE    = 004000         wTLOCK  007110R
TABLE   001416R         UNITNO  000374R         wBUFPA  000134R         wRDCKR  002604R         XFERAD  001550R
TAB03   001554R         UNS     = 040000        wBUFRQ  000140R         wRMCKR  002664R         XFLAG   000005R
TAB05   001574R         UNSAFE  007060R         wBUFSZ  000142R         wRTICK  002444R         XMEM    000354R
TIMERR  007140R         UPE     = 020000        wCE    = 040000         wRTIE   002364R         ZERO    000404R
TRE    = 040000         VECTOR  000010R

. ABS.  000000      000
        007214      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  4872 WORDS  ( 20 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  69 PAGES
XRMCA0,CXRMCA/C=DDACOM,CXRMCA.DOC,CXRMCA
```



```
ACSR      3-5#      8-286*
ADDR      3-5#      8-292
ADDR22    3-5#
ADDRES    8-140     8-361#
ADR1      7-166     8-398#
AOE       3-205#    8-136
ASB       3-5#      8-278*
ASTAT     3-5#
ATA       3-192#    8-38      8-39
AWAS      3-5#      8-280*
BADSEC    3-28#     7-36      7-38      8-10*     8-66*     8-80*
BADSPT    3-6#      8-58
BEGIN     3-5#      4-49      4-63      4-69      5-9       5-28      5-35      7-11      7-23      7-25      7-43      7-61      7-64      7-67
          7-166     8-18      8-20      8-28      8-30      8-42      8-42      8-43      8-47      8-47      8-88      8-92      8-98      8-107
          8-109     8-115     8-119     8-122     8-126     8-130     8-134     8-140     8-144     8-148     8-152     8-155     8-159     8-163
          8-167     8-171     8-173     8-178     8-208     8-212     8-214     8-235     8-235     8-240
BIT0      3-5#
BIT1      3-5#
BIT10     3-5#      7-59      8-298     8-299
BIT11     3-5#
BIT12     3-5#      4-15      4-16      4-20
BIT13     3-5#
BIT14     3-5#
BIT15     3-5#      8-301     8-302
BIT2      3-5#      8-103
BIT3      3-5#
BIT4      3-5#      4-15      4-16
BIT5      3-5#
BIT6      3-5#
BIT7      3-5#
BIT8      3-5#
BIT9      3-5#
BITTAB    7-146     7-152#
BR1       3-5#      8-311
BR2       3-5#
BREAK$    3-5#      8-42      8-42      8-47      8-47      8-235     8-235
BSE       3-220#    7-38      8-78
BST       3-123#    7-111
BTOD$     3-5#
CDATA$    3-5#      5-28
CDERCT    3-5#
CDWDCT    3-5#
CLK       3-34#     8-231*    8-236*
CLR       3-168#    8-243
CONFIG    3-5#
CSRA      3-5#      8-285*
CYCLE     4-60      5-88
CYL       3-126#    4-29*     4-30*     7-62*     7-63*     7-71      7-73*     7-102*    7-103     7-105*    7-109     7-133
CYLIND    3-24#     7-20      7-133*    8-60
DATCK$    3-5#
DATER$    3-5#
DATERR    8-163     8-369#
DCK       3-211#    8-161
DEVNOT    8-208     8-387#
DLTCNT    3-30#     4-31*     8-102*
DLTERR    8-107     8-383#
```

SEQ 0032

```
DPE       3-215#
DPR       3-185#    8-25
DRM       5-35      8-214     8-393#
DROP      5-34      7-159#    8-213
DRPDRV    3-22#     5-32      7-32      8-9*      8-17*     8-27*
DRVERR    8-173     8-375#
DRVTYP    3-36#     7-128     8-193*    8-194*
DRY       3-184#    8-25
DSKADR    3-25#     7-21      7-131*    7-132*    8-62      8-64
DTE       3-208#    8-157
DTL       3-178#    8-100
DVA       3-162#    8-233
DVC       3-216#
DVICE     3-26#     4-6*      4-66      7-146     7-165*
DVID1     3-5#      4-6
EA22      3-20#     7-13      7-14*     7-15*     7-16
ECH       3-202#    8-161
END$      3-5#      4-69
ENDIT$    3-5#      4-63
ERR       3-191#    8-51      8-96      8-111
ERRORS    7-27      8-88
ERRTYP    3-5#      8-19*     8-29*     8-108*    8-121*    8-177*    8-211*    8-239*
ERSUB1    8-11      8-207     8-238     8-285#
ERSUB2    8-278#
EXCED     7-43      8-351#
EXIT$     3-5#      7-23
FER       3-200#    8-132
FERADR    3-33#     3-89      6-19*     6-28*     6-37*     6-46*     6-55*     6-64*     6-73*
FLAG70    3-23#     7-4       8-268     8-291*    8-301*
FMT       3-224#    7-19
FORMAT    8-134     8-359#
FREE      3-5#
FT        3-119#    4-8*      4-11*     4-18*     4-19*     7-87      7-89      7-101
FUNC      3-31#     6-18*     6-27*     6-36*     6-45*     6-54*     6-63*     6-72*     7-17*     7-22
GENADR    4-52      7-56#
GET       8-12      8-210     8-255#
GETDRV    4-35      4-54      7-141#
GETPA$    3-5#      4-49
GO        3-164#    8-14
GO1       6-25      6-34      6-43      6-52      6-61      6-70      7-3#
GO2       6-74      7-18#
G#BUF$    3-5#      5-9
HCE       3-203#    8-74      8-128
HCRC      3-204#    8-76      8-128
HEADER    8-130     8-357#
HRDCNT    3-5#
HRDER$    3-5#      8-20      8-30      8-122     8-212
HRDPAS    3-5#
IAE       3-206#    8-136
ICONT     3-5#
ICOUNT    3-5#
IDNUM     3-5#
ILF       3-196#
ILR       3-197#
IMODX.    3-5#      5-9
INIT      3-5#

INTR      3-5#      4-5*
```

SEQ 0033

SEQ 0034

```
IR          3-169#
LBC         3-217#      8-165
LBT         3-187#      8-138
LC          3-122#      7-71        7-103       7-109
LDADR       4-59        7-127#
LOOP1       4-50#       4-67
LOOP2       4-53#       4-57        4-64
LOSBIT      8-167       8-371#
LOSYSC      8-152       8-365#
LS          3-121#      7-79        7-95
LSC         3-218#      8-150
LT          3-120#      4-9*        4-12*       4-22*       4-23*       7-75        7-83        7-85        7-99
MAP22#      3-5#        7-11
MCPE        3-161#      8-49        8-86
MCPERR      8-88        8-347#
MDPE        3-171#      8-90
MDPERR      8-92        8-349#
MES1        8-316#      8-345
MES10       8-322#      8-355
MES11       8-323#      8-357
MES12       8-324#      8-359
MES14       8-325#      8-375
MES15       8-326#      8-353
MES16       8-327#      8-361
MES17       8-328#      8-363
MES2        8-317#      8-347
MES20       8-329#      8-365
MES21       8-330#      8-367
MES22       8-331#      8-369
MES23       8-332#      8-371
MES24       8-333#      8-373
MES25       8-334#      8-377
MES26       8-335#      8-379
MES27       8-336#
MES3        8-318#      8-349
MES30       8-337#      8-381
MES31       8-338#      8-383
MES32       8-339#      8-385
MES33       8-340#      8-387
MES34       8-341#      8-389
MES35       8-342#      8-391
MES4        8-319#      8-393
MES5        8-320#      8-395
MES6        8-321#      8-351
MODNAM      3-5#
MODSP       3-5         3-5#
MOL         3-189#      8-25
MSG#        3-5#
MSGM#       3-5#        5-35        7-43        8-18        8-28        8-43        8-88        8-92        8-98        8-107       8-115       8-119       8-126       8-130
            8-134       8-140       8-144       8-148       8-152       8-155       8-159       8-163       8-167       8-171       8-173       8-208       8-214       8-240
MSGS#       3-5#
MXF         3-172#      8-94
NAS         3-49        3-101#
NBA         3-44        3-96#
NBAE        3-62        3-114#      8-270
NCS1        3-42        3-94#       8-13        8-33        8-49        8-86        8-117
NCS2        3-46        3-98#       8-90        8-94        8-100       8-113
```

SEQ 0035

SEQ 0036

```
NCS3    3-63    3-115#
NDA     3-45    3-97#
NDB     3-51    3-103#
NDC     3-56    3-108#
NDS     3-47    3-99#   8-23    8-37    8-51    8-96    8-111   8-138
NDT     3-53    3-105#
NEC1    3-60    3-112#
NEC2    3-61    3-113#
NED     3-175#  8-94
NEM     3-174#  8-94
NER1    3-48    3-100#  8-74    8-76    8-124   8-128   8-132   8-136   8-142   8-153   8-157   8-161   8-169
NER2    3-59    3-111#  8-78    8-80    8-146   8-150   8-165
NHR     3-57    3-109#
NLA     3-50    3-102#
NMR1    3-52    3-104#
NMR2    3-58    3-110#
NODRDY  8-28    8-385#
NODVA   8-240   8-391#
NOF     3-55    3-107#
NOTRDY  8-18    8-389#
NSN     3-54    3-106#
NTRUPT  7-24#   8-310
NULL    3-5#
NUMB    8-394   8-399#
NWC     3-43    3-95#
OM      3-182#
OPEN    3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5
        3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5
        3-5#
OPI     3-209#  8-142
OR      3-170#  8-259
OTUA$   3-5#    7-166
PA18    3-17#   7-6#    7-11
PA22    3-19#   7-12
PAR     3-199#
PASCNT  3-5#    4-45
PGE     3-173#  8-94
PGM     3-186#
PIP     3-190#  8-38
PIRQ$   3-5#    7-25
POPSP   3-5#
POPSP2  3-5#
POSIT   8-144   8-381#
PRTY    3-5#
PRTY0   3-5     3-5#
PRTY1   3-5#
PRTY2   3-5#
PRTY3   3-5#
PRTY4   3-5#
PRTY5   3-5     3-5#
PRTY6   3-5#
PRTY7   3-5#
PS      3-5#
PSW     3-5#
PUSH    3-5#
PUSH2   3-5#

R6      3-5#    4-7#
```

SEQ 0037

SEQ 0038

```
R7          3-5#
RAND$       3-5#        7-61        7-64        7-67
RANNUM      3-5#        7-62        7-65        7-68
RBUF        3-5         3-38#
RBUFEA      3-5#        6-42        6-51        6-60        6-69
RBUFPA      3-5#        5-28        6-41        6-50        6-59        6-68
RBUFSZ      3-5#        6-38        6-47        6-56        6-65
RBUFVA      3-5#        4-49
RDY         3-163#      8-14        8-15
READ        5-26        6-36#       6-37
READHD      6-63#       6-64
READY       4-37        4-56        7-34        8-190#
RES1        3-5#
RES2        3-5#
RESTRT      3-5         4-45#
RHFAIL      8-98        8-353#
RMAS        3-73#
RMBA        3-68#       6-23*       6-32*       6-41*       6-50*       6-59*       6-68*       7-6         7-12*
RMBAE       3-86#       7-13*       8-296
RMCS1       3-66#       4-39*       7-22*       8-41*       8-46*       8-233       8-256       8-285       8-286       8-293       8-298*      8-299
RMCS2       3-70#       8-230*      8-243*      8-244*
RMCS3       3-87#
RMDA        3-69#       7-21*
RMDB        3-75#       8-257
RMDC        3-80#       7-20*
RMDS        3-71#
RMDT        3-77#       8-192
RMEC1       3-84#
RMEC2       3-85#
RMER1       3-72#
RMER2       3-83#
RMHR        3-81#
RMLA        3-74#
RMMR1       3-76#
RMMR2       3-82#
RMOF        3-79#       7-19*
RMR         3-198#
RMSN        3-78#
RMWC        3-67#       6-22*       6-31*       6-40*       6-49*       6-58*       6-67*
RSTRT       3-5#
RSTRT1      4-43        4-46        4-48#
S           3-42        3-43        3-44        3-45        3-46        3-47        3-48        3-49        3-50        3-51        3-52        3-53        3-54        3-55
            3-56        3-57        3-58        3-59        3-60        3-61        3-62        3-63        3-155#      8-13        8-23        8-33        8-37        8-49
            8-51        8-74        8-76        8-78        8-80        8-86        8-90        8-94        8-96        8-100       8-111       8-113       8-117       8-124
            8-128       8-132       8-136       8-138       8-142       8-146       8-150       8-153       8-157       8-161       8-165       8-169       8-261*      8-263*
            8-270       8-271*      8-273*
SBADR       3-5#        8-279*
SC          3-159#
SEC         3-124#      4-25*       4-26*       7-68#       7-69#       7-79        7-81#       7-94*       7-95        7-97*       7-131
SEEK        5-12        5-18        5-24        6-72#       6-73
SEEKER      8-43        8-148       8-363#
SETUP       4-32        8-291#
SKI         3-219#      8-146
SOFCNT      3-5#
SOFER#      3-5#        8-109       8-178
SOFPAS      3-5#


SPOINT      3-5#        4-7
```

SEQ 0039

SEQ 0040

```
SPSIZ   1-7#    3-5
SR1     3-5#    4-14    7-59    8-103   8-302
SR2     3-5#
SR3     3-5#
SR4     3-5#
START   3-5     4-3#    4-47
STAT    3-5#
SVR0    3-5#
SVR1    3-5#
SVR2    3-5#
SVR3    3-5#
SVR4    3-5#
SVR5    3-5#
SVR6    3-5#
SYSCNT  3-5#
TAB03   3-137#  4-8*    4-9*    4-18*   4-22*   4-25*   4-27    4-27*   4-29*   7-57    7-127
TAB05   3-146#  4-11*   4-12*   4-19*   4-23*   4-26*   4-28    4-28*   4-30*   7-115   7-117   7-130
TABLE   3-42#   8-20    8-30    8-109   8-122   8-178   8-212   8-267#  8-270*
TIMERR  8-159   8-379#
TRE     3-160#  8-49    8-117
TRERR   8-119   8-345#
TRK     3-125#  4-27#   4-28*   7-65*   7-66*   7-75    7-77*   7-83    7-85*   7-86*   7-87    7-89*   7-98*   7-99
        7-101*  7-111   7-132
TRPDFD  3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5
        3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5
        3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5#    3-5#    3-5#    3-5#    3-5#    3-5#
        3-5#    3-5#    3-5#    3-5#    3-5#    3-5#    3-5#    3-5#    3-5#    3-5#    3-5#    3-5#    3-5#
TRY     3-35#   5-8*    7-40*   7-41    7-44*
UNITNO  3-27#   4-33*   4-51*   7-142*  7-143   7-145   7-160   7-166   8-230   8-244
UNS     3-210#  8-124
UNSAFE  8-126   8-355#
UPE     3-176#  8-94
VECTOR  3-5#    8-309
VV      3-183#  8-25
WAIT    5-10    5-16    5-22    8-190   8-230#
WASADR  3-5#    8-281#
WBUFEA  3-5#    6-24    6-33
WBUFPA  3-5#    6-23    6-32
WBUFRQ  3-5#
WBUFSZ  3-5#    6-20    6-29
WCE     3-177#  8-113
WCF     3-201#  8-169
WDFR    3-5#    4-4*
WDTO    3-5#    4-3*
WLE     3-207#  8-153
WRDCKR  6-45#   6-46
WRMCKR  6-54#   6-55
WRITCK  5-20    6-27#   6-28
WRITE   5-14    6-18#   6-19
WRL     3-188#
WRTCK   8-115   8-377#
WTCLCK  8-171   8-373#
WTLOCK  8-155   8-367#
XFERAD  3-89#
XFLAG   3-5#
XMEM    3-18#   6-24*   6-33*   6-42*   6-51*   6-60*   6-69*   7-7*    7-8*    7-9*    7-10*   7-16*   7-17

ZERO    3-32#
```

SEQ 0041

```
BKMOD   1-95#
BREAK   1-185#  8-42    8-47    8-235
BTOD    1-204#
CKDATA  1-240#  5-28
DATACK  1-249#
DATERR  1-138#
DFSEVN  1-272#  3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5     3-5
DSEVNT  1-282#  3-5     3-5     3-5     3-5     3-5     3-5
END     1-175#
ENDIT   1-166#  4-63
ENDMOD  1-171#  4-69
EQUATS  1-288#  3-5
EXIT    1-120#  7-23
GETPA   1-231#  4-49
GWBUFF  1-219#  5-9
HRDER   1-128#  8-20    8-30    8-122   8-212
IOMOD   1-91#
IOMODP  1-115#
IOMODR  1-111#
IOMODX  1-107#  3-5
LINEUP  5-38#   6-24    6-33    6-42    6-51    6-60    6-69
MAP22   1-235#  7-11
MODULE  1-8#    3-5
MSG     1-154#
MSGN    1-158#  5-35    7-43    8-18    8-28    8-43    8-88    8-92    8-98    8-107   8-115   8-119   8-126   8-130
        8-134   8-140   8-144   8-148   8-152   8-155   8-159   8-163   8-167   8-171   8-173   8-208   8-214   8-240
MSGS    1-162#
NBKMOD  1-103#
OTOA    1-190#  7-166
PIRQ    1-179#  7-25
RAND    1-124#  7-61    7-64    7-67
SBKMOD  1-99#
SOFER   1-144#  8-109   8-178
```