.REM _

IDENTIFICATION

PRODUCT CODE: AC-E971D-MC

PRODUCT NAME: CXRMAD0 RH11,70/RM03,02 S P M

PRODUCT DATE: FEB 1979

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

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

2. REQUIREMENTS

HARDWARE: 1 TO 8 RM03/RM02 WITH AN RH11/RH70 CONTROLLER

STORAGE:: RMA REQUIRES:
1. DECIMAL WORDS: 1825
2. OCTAL WORDS: 03441
3. OCTAL BYTES: 7102

3. PASS DEFINITION

ONE PASS OF THE RMA MODULE CONSISTS OF 1300 CYCLES OF THE BASIC TEST SEQUENCE (WRITE, WRITE-CHECK, READ, DATA-CHECK). THE TEST SEQUENCE WRITES 1024 WORDS, WRITE-CHECKS SAME, READS THE FIRST 256 WORDS, AND DATA-CHECKS SAME.

4. EXECUTION TIME

ONE PASS OF RMA RUNNING ALONE ON A PDP-11/40 TAKES APPROXIMATELY 0.9 MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 176700, VECTOR: 254, BR1: 5, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP

MAKE CERTAIN THAT ALL DRIVES ARE POWERED UP, WRITE ENABLED, AND READY

THIS MODULE ALSO SUPPORTS RP04/5/6 ON THE SAME MASSBUS CONTROLLER. HOWEVER,THIS MODULE IS NOT USED TO EXERCISE RP04/5/6 ALONE.

7. MODULE OPERATION

TEST SEQUENCE:

A. SETUP DEVICE REGISTER ADDRESSES AND MODULE VARIABLES
B. RESET ALL DRIVES ON-LINE AND DROP ALL THAT ARE NOT
C. GET A STARTING SECTOR ADDRESS
D. GET A DRIVE ADDRESS
E. DO A WRITE -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
F. DO A WRITE-CHECK -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
G. DO A READ -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
H. DO A DATA-CHECK -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
I. IF END OF PASS, REPORT AND GO TO C
J. IF END OF DRIVES, GO TO C ELSE GO TO D

8. OPERATION OPTIONS

SR1 BIT0 SET(1):
IF THE RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, A HARD ERROR IS ASSUMED AND THE DRIVE IS DROPPED

SR1 BIT0 CLEAR(0):
IF THE RETRY LIMIT IS EXCEEDED, THE FUNCTION IS ABORTED AND THE TESTING CONTINUES

SR1 BIT2 SET(1):
COUNT DATA LATE ERRORS BUT DO NOT TYPE THEM OUT

SR1 BIT2 CLEAR(0):
TYPE OUT DATA LATE ERRORS AND COUNT THEM

```
SR1 BIT5 CLEAR (0)      ;NORMAL FOR RM03 PACKS THAT ARE FORMATED
                        ;FOR 16 BIT MODE (PDP-11)
SR1 BIT5 SET (1)        ;FOR RM03 18 BIT FORMATED PACKS

SR1 BIT15 SET (1)       ;32 REGISTERS ON RH70
SR1 BIT15 CLEAR (0)     ;22 REGISTERS ON RH70
```

9. NON-STANDARD PRINTOUTS

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

B. ERROR MESSAGES DUMP THE CONTENTS OF THE 20 RH11/RM03 REGISTERS IN THE FOLLOWING ORDER:

RMCS1 RMWC RMBA RMDA RMCS2 RMDS RMER1 RMAS
RMLA RMDB RMMR1 RMDT RMSN RMOF RMDC RMHR
RMMR2 RMER2 RMEC1 RMEC2 RMBAE RMCS3 XFER CNT

10. BAD SPOTS

A. LOCATION 256 THROUGH 452 CONTAIN ROOM FOR 32. BAD SPOTS.
EACH BAD SPOT TAKES TWO WORDS.
THE FIRST WORD SPECIFIES THE CYLINDER ADDRESS.
THE LOW BYTE OF THE SECOND WORD SPEFICIES THE SECTOR ADDRESS.
THE HIGH BYTE OF THE SECOND WORD SPECIFIES THE TRACK ADDRESS.

B. THE RMA MODULE DOES NOT ACCESS THESE SPOTS ON ANY OF THE DRIVE UNDER TEST.

C. THE BAD SPOT FILE IS RETRIEVED FROM ALL DRIVES ASSIGNED IN THE BIT MAP "DVID1".

(CYLINDER 822, TRACK 4, SECTOR 0 - MANUFACTUR BAD SPOT FILE)
(CYLINDER 822, TRACK 4, SECTOR 12 - USER BAD SPOT FILE)
LOCATION 1760 MUST BE SET TO 10 TO SEARCH USER BAD SPOT FILE

```
                        IOMODX  <RMAD  >,176700,254,5,0,0,1300.,144,BUFIN,256.,1024.
                        MODULE  150000,RMAD  ,176700,254,5,0,0,1300.,144,BUFIN,256.,1024.
                        .TITLE  RMAD DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6      23-MAY-78
                                .LIST   BIN
                                ;*****************************************************************************
000000'                         BEGIN:
000000' 046522  042101     040  MODNAM: .ASCII  /RMAD / ;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' 001662'                 INIT:   START                   ;MODULE START ADDR.
000032' 000252'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                       ;PASS COUNTER.
000036' 002424                  ICONT:  1300.                   ;# OF ITERATIONS PER PASS=1300.
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                  AS8:    OPEN                    ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN                    ;ACTUAL DATA.
000112' 002534'                 RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114' 002000                  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' 000144                  IDNUM:  144                     ;MODULE IDENTIFICATION NUMBER=144
000124' 000600'                 RBUFVA: BUFIN                   ;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' 002000                  WBUFRQ: 1024.                   ;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:
                                ;*****************************************************************************
```

```
000252' 000254'                 FILE:   .WORD   BADSPT          ;BAD SECTOR TABLE ADDRESS
000254'                         BADSPT:                         ;BAD SPOT TABLE
                                                                ;FIRST WORD = CYLINDER ADDRESS
                                                                ;SECOND WORD = TRACK + SECTOR ADDRESS
000254' 177777                          .WORD   -1
000256' 177777                          .WORD   -1
000260' 177777                          .WORD   -1
000262' 177777                          .WORD   -1
000264' 177777                          .WORD   -1
000266' 177777                          .WORD   -1
000270' 177777                          .WORD   -1
000272' 177777                          .WORD   -1
000274' 177777                          .WORD   -1
000276' 177777                          .WORD   -1
000300' 177777                          .WORD   -1
000302' 177777                          .WORD   -1
000304' 177777                          .WORD   -1
000306' 177777                          .WORD   -1
000310' 177777                          .WORD   -1
000312' 177777                          .WORD   -1
000314' 177777                          .WORD   -1
000316' 177777                          .WORD   -1
000320' 177777                          .WORD   -1
000322' 177777                          .WORD   -1
000324' 177777                          .WORD   -1
000326' 177777                          .WORD   -1
000330' 177777                          .WORD   -1
000332' 177777                          .WORD   -1
000334' 177777                          .WORD   -1
000336' 177777                          .WORD   -1
000340' 177777                          .WORD   -1
000342' 177777                          .WORD   -1
000344' 177777                          .WORD   -1
000346' 177777                          .WORD   -1
000350' 177777                          .WORD   -1
000352' 177777                          .WORD   -1
000354' 177777                          .WORD   -1
000356' 177777                          .WORD   -1
000360' 177777                          .WORD   -1
000362' 177777                          .WORD   -1
000364' 177777                          .WORD   -1
000366' 177777                          .WORD   -1
000370' 177777                          .WORD   -1
000372' 177777                          .WORD   -1
000374' 177777                          .WORD   -1
000376' 177777                          .WORD   -1
000400' 177777                          .WORD   -1
000402' 177777                          .WORD   -1
000404' 177777                          .WORD   -1
000406' 177777                          .WORD   -1
000410' 177777                          .WORD   -1
000412' 177777                          .WORD   -1
000414' 177777                          .WORD   -1
000416' 177777                          .WORD   -1
000420' 177777                          .WORD   -1
000422' 177777                          .WORD   -1
```



```
000424' 177777                          .WORD   -1
000426' 177777                          .WORD   -1
000430' 177777                          .WORD   -1
000432' 177777                          .WORD   -1
000434' 177777                          .WORD   -1
000436' 177777                          .WORD   -1
000440' 177777                          .WORD   -1
000442' 177777                          .WORD   -1
000444' 177777                          .WORD   -1
000446' 177777                          .WORD   -1
000450' 177777                          .WORD   -1
000452' 177777                          .WORD   -1
000454' 000000                  DLTCNT: 0                       ;DATA LATE ERROR COUNTER
000456' 000000                  CLK:    0
000460' 000000                  BLK1:   0                       ;DISK BLOCK COUNTER
000462' 000000                  CYL:    0                       ;CALCULATED CYLINDER ADDRESS
000464' 000000                  CYLSAV: 0                       ;MODIFIED CYLINDER ADDRESS FOR RM03
000466' 000000                  SEC:    0
000470' 000000                  CNT:    0                       ;CYCLE COUNTER
000472' 000000                  CNT1:   0                       ;CYCLE COUNTER1 - DT03 SUPPORT
000474' 000000                  FUNC:   0
000476' 000000                  ONCEE:  0
000500' 000000                  BADSEC: 0               ;HARDWARE DETECTOR BAD SECTOR
000502' 000000                  FERADR: 0

                                ;DO NOT CHANGE THE ORDER OF THE NEXT 15 LOCATIONS
                                ;NEEDED FOR MAP22 ROUTINE

000504' 000000                  PA18:   0
000506' 000000                  XMEM:   0
000510' 000000                  PA22:   0
000512' 000000                  EA22:   0
000514' 000000                  MBLKRV: 0                       ;HOLDS 1024 WD BLKS PER TRACK
000516' 000000                  MBLKTR: 0                       ;HOLDS BLKS PER CYLINDER
000520' 000000                  MODE:   0                       ;HOLDS LAST SECTOR # THAT WILL FIT PER TRACK
000522' 000000                  MLOWCY: 0                       ;BLKS THAT FIT IN FIRST 410 CYLINDERS (0-409)
000524' 000000                  MHICY:  0                       ;BLKS THAT FIT IN LAST 413 CYLINDERS (410-822)
                                ;

                                                                ;1 BLOCK =4 SECTORS = 1024 (256 X 4 ) WORD.
000526' 000010                  T16:    8.                      ;THESE 5 LOC. GET MOVED TO ABOVE 5 IF 16 BIT MODE
000530' 000050                          40.                     ;1024 WORD BLOCKS/CYLINDER
000532' 000040                          32.                     ;SECTORS/TRACK 8. X 4.
000534' 040020                          410.*40.                        ;410. X 40. (CYLINDER 0 - 409)
000536' 040157                          413.*40.-25.                    ;413. X 40. (CYLINDER 410 - 822)
                                                                        ;EXCLUDING THE LAST 25 BLOCKS


                                ;****************
000540' 000007                  T18:    7.                      ;THESE 5 LOC GET MAPPED INTO FIRST 5 ABOVE IF 18 BIT MOD
000542' 000043                          35.                     ;1024 WORD (18 BIT) BLOCKS/CYLINDER
000544' 000034                          28.                     ;SECTORS/TRACK 7. X 4.
                                                                ;ACTUAL SECTORS ON A TRACK
                                                                ;FOR 18 BIT WORD FORMAT ARE 30.
000546' 034016                          410.*35.                        ;410. X 35. (CYLINDER 0 - 409)
000550' 034136                          413.*35.-25.                    ;413. X 35. (CYLINDER 410 - 822)
                                                                        ;EXCLUDING THE LAST 25 BLOCKS
```

```
                                ;*****************
                                ZERO:   0
000552' 000000                  DSKADR: 0
000554' 000000                  DVICE:  0
000556' 000000                  DRIVE:  0
000560' 000000                  BLKSAV: 0
000562' 000000                  TRUF:   0
000564' 000000                  WCNT1:  0
000566' 000000                  WCNT2:  0
000570' 000000                  UNITNO: 0
000572' 000000                  MIXDV:  0               ;MIXDV=-1,IF NOT A RM03 OR RM02
000574' 000000                  MOD1:   0
000576' 000000                  BUFIN:  .BLKW   256.
```



```
001600'                         TABLE:
001600' 000000                  RMCS1:  0
001602' 000000                  RMWC:   0
001604' 000000                  RMBA:   0
001606' 000000                  RMDA:   0
001610' 000000                  RMCS2:  0
001612' 000000                  RMDS:   0
001614' 000000                  RMER1:  0
001616' 000000                  RMAS:   0
001620' 000000                  RMLA:   0
001622' 000000                  RMDR:   0
001624' 000000                  RMMR1:  0
001626' 000000                  RMDT:   0
001630' 000000                  RMSN:   0
001632' 000000                  RMOF:   0
001634' 000000                  RMDC:   0
001636' 000000                  RMHR:   0
001640' 000000                  RMMR2:  0
001642' 000000                  RMER2:  0
001644' 000000                  RMEC1:  0
001646' 000000                  RMEC2:  0
001650' 000000                  RMBAE:  0
001652' 000000                  RMCS3:  0
001654' 000502'                 XFERAD: FERADR
001656' 000470'                 XFERCT: CNT
001660' 177777                          177777
```

```
  400  001662' 012767  002000  176226  START:  MOV     #1024.,WDER     ;1024. WORDS FROM MEM/ITERATION
  401  001670' 012767  000400  176216          MOV     #256.,@DIO      ;256 WORDS TO MEM/ITERATION
  402  001676' 012767  000003  176214          MOV     #3,INTR         ;3 INTERRUPTS/ITERATION
  403  001704' 005067  176560                  CLR     CNT             ; ZERO END OF PASS TESTER
  404  001710' 005067  176540                  CLR     DLTCNT          ;CLEAR DATA LATE ERROR COUNTER
  405  001714' 012767  010000  176654          MOV     #BIT12,MOD1     ;SETUP FOR 16 BIT MODE (NORMAL)
  406  001722' 012700  000526'                 MOV     #T16,R0         ;GET TABLE OF VALUES FOR 16 BIT MODE
  407  001726' 013767  000040  176062          BIT     #BIT5,SR1       ;16 BIT MODE?
  408  001734' 001404                          BEQ     2$              ;YES
  409  001736' 005067  176634                  CLR     MOD1            ;NO 18 BIT MODE. CLEAR FMT BIT
  410  001742' 012700  000540'                 MOV     #T18,R0         ;GET TABLE OF VALUES FOR 18 BIT MODE
  411  001746' 012701  000514'          2$:    MOV     #MBLKRV,R1      ;BEGIN OF TABLE ENTRIES
  412  001752' 012702  000005                  MOV     #5,R2           ;NO. OF ENTRIES
  413  001756' 012021                  3$:     MOV     (R0)+,(R1)+     ;STORE AN ENTRY
  414  001760' 005302                          DEC     R2              ;COUNT IT
  415  001762' 001375                          BNE     3$              ;DO MORE
  416  001764' 012767  000007  176504  1$:     MOV     #7,ONCEE        ;SET ONE TIME ONLY FLAGS
  417  001772' 105067  005102                  CLRB    FLAG            ; CLEAR FLAGS
  418  001776' 016767  176012  176552          MOV     DVID1,DVICE     ; GET DRIVE INDICATOR
  419  002004' 122737  000016  000041          CMPB    #16,@#41        ;IF RM IS LOAD MEDIUM THEN
  420  002012' 001021                          BNE     13$             ;BEGIN
  421  002014' 113700  000040                  MOVB    @#40,R0         ;  GET LOAD-DRIVE NUMBER
  422  002020' 012701  000001                  MOV     #1,R1           ;  INITIALIZE DRIVE POINTER
  423  002024' 105700                  10$:    TSTB    R0              ;  WHILE NOT POINTING AT LOAD-DRIVE DO
  424  002026' 001403                          BEQ     12$             ;  BEGIN
  425  002030' 006301                          ASL     R1              ;    POINT TO NEXT DRIVE
  426  002032' 105300                          DECB    R0              ;    COUNT SHIFTS
  427  002034' 000773                          BR      10$             ;  END
  428  002036' 130167  176514          12$:    BITB    R1,DVICE        ;  IF LOAD-DRIVE SELECTED THEN
  429  002042' 001405                          BEQ     13$             ;  BEGIN
  430  002044' 140167  176506                  BICB    R1,DVICE        ;    DROP THE DEVICE
  431  002050' 104403  000000' 006746'         MSGN$,BEGIN,LDRIVE      ;ASCII MESSAGE CALL WITH COMMON HEADER
  432                                          ;  END
  433  002056'                         13$:                            ;END
  434  002056' 005767  176474                  TST     DVICE           ;IF NO DRIVES ARE SELECTED THEN
  435  002062' 001002                          BNE     DAOST           ;BEGIN
  436  002064' 000167  000752                  JMP     FINI            ;  DROP MODULE
  437                                          ;END
  438  002070' 016767  176462  176462  DAOST:  MOV     DVICE,DRIVE     ; ALSO SAVE IT IN DRIVE
  439  002076' 016706  175730                  MOV     SPOINT,R6       ;RESTORE STACK POINTER
  440  002102' 012767  177777  176350          MOV     #-1,BLK1        ; INITIALIZE BLOCK COUNTER
  441  002110' 012767  177777  176454          MOV     #-1,UNITNO      ; INITIALIZE DEVICE COUNTER
  442  002116' 004767  003602                  JSR     PC,SETUP        ;SET UP CONTROLLER REGISTER
  443  002122' 004767  003504                  JSR     PC,REZET        ;RESET CONTROLLER AND DRIVES
  444  002126' 004767  003070                  JSR     PC,FDUNIT       ;FOUND THE FIRST DRIVE AVAILABLE
  445  002132' 132767  000010  004740          BITB    #BIT3,FLAG      ;NO DRIVE ?
  446  002140' 001402                          BEQ     .+6             ;DRIVE AVAILABLE
  447  002142' 000167  000674                  JMP     FINI            ;DROP THE MODULE
  448  002146' 004567  003330                  JSR     R5,READY        ;DEVICE READY ?
  449  002152' 000402                          BR      CT              ;YES
  450  002154' 004767  003120                  JSR     PC,NOTRDY       ;NO ,TRY NOT READY ROUTNIE
  451  002160'                         CT:
  452  002160' 005767  176372                  TST     DVICE           ;DROP THE MODULE ?
  453                                  ;       BNE     BADFL           ;NO
  454  002164' 001002                          BNE     1$              ;BRANCH IF DRIVE AVAILABLE
  455  002166' 000167  000650                  JMP     FINI            ;YES DROP MODULE
```



```
  456  002172' 012701  000254'  1$:     MOV     #BADSPT,R1              ;TABLE ADDRESS
  457  002176' 012702  000100           MOV     #64.,R2         ;32 WORDS
  458  002202' 012721  177777   2$:     MOV     #-1,(R1)+       ;RESET TO -1,
  459  002206' 005302                   DEC     R2              ;DECREMENT ONE WORD
  460  002210' 001374                   BNE     2$              ;BRANCH IF NOT DONE
  461  002212' 012767  001466  176242  BADFL:  MOV     #822.,CYL       ;RETRIVE THE BAD SPOT FILE
  462  002220' 112767  000004  176327          MOVB    #4,DSKADR+1     ;FROM THE PACK CYL-822, TRK-4
  463  002226' 112767  000000  176320          MOVB    #0,DSKADR       ;SEC-0(SEC-12 FOR USER BAD SPOT FILE)
  464  002234' 005767  176336                  TST     MOD1            ;IN 16 BIT MODE ?
  465  002240' 001002                          BNE     .+6             ;BRANCH IF IT IS
  466  002242' 005267  176306                  INC     DSKADR          ;OTHERWISE START FROM SECTOR 1
  467  002246' 016777  176320  177334          MOV     UNITNO,@RMCS2   ;LOAD DRIVE NUMBER
  468  002254' 012777  000000  177352          MOV     #0,@RMDC        ;SET TO CYL 0
  469  002262' 012777  000000  177316          MOV     #0,@RMDA        ;SET TO TRK 0,SEC 0 AT START TIME
  470  002270'                  BADFL2:
  471  002270' 104415  000000'  000124'         GETPAS,BEGIN, RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
  472  002276' 004767  002636                   JSR     PC,RM03CK       ;CHECK IF RM03 OR RM02
  473  002302' 005767  176266                   TST     MIXDV           ;NOT THEM ?
  474  002306' 100455                           BMI     4$              ;BRANCH IF NOT
  475  002310' 016767  175616  176252           MOV     RBUFSZ,WCNT2    ;WORD CTR
  476  002316' 005467  176246                   NEG     WCNT2           ;2'S COMPLEMENT
  477  002322' 004567  001104                   JSR     R5,READ         ;READ ONE SECTOR AT A TIME
  478  002326' 000167  000150                   JMP     RETRX           ;BRANCH IF ERROR
  479  002332' 012701  000600'                  MOV     #BUFIN,R1       ;BUFFER ADDRESS
  480  002336' 012702  000254'                  MOV     #BADSPT,R2              ;TABLE ADDRESS
  481  002342' 012703  000200                   MOV     #128.,R3                ;MAX 16 BAD SECTORS, EACH SECTOR DEFINED BY TWO
  482  002346' 060203                           ADD     R2,R3           ;LAST ADDRESS
  483  002350' 012704  000400                   MOV     #256.,R4        ;END OF BAD SPOT FILE
  484  002354' 060104                           ADD     R1,R4           ;LAST ADDRESS OF BAD SPOT FILE
  485  002356' 062701  000010                   ADD     #10,R1          ;BAD SECTOR STARTS AT 5TH WORD
  486  002362' 022711  177777           1$:     CMP     #-1,(R1)        ;TABLE IS EMPTY ?
  487  002366' 001425                           BEQ     4$              ;BRANCH IF IT IS
  488  002370' 022712  177777                   CMP     #-1,(R2)        ;ENTRY FOR BAD SPOT TABLE ?
  489  002374' 001006                           BNE     2$              ;BRANCH IF NOT
  490  002376' 011112                           MOV     (R1),(R2)       ;LOAD CYLINDER ADDRESS
  491  002400' 016162  000002  000002           MOV     2(R1),2(R2)     ;LOAD THE TRACK AND SECTOR ADDRESSES
  492  002406' 062701  000004                   ADD     #4,R1           ;ASJUST POINTER
  493  002412' 062702  000004           2$:     ADD     #4,R2           ;ADJUST POINTER
  494  002416' 020104                           CMP     R1,R4           ;END OF TABLE ?
  495  002420' 103010                           BHIS    4$              ;BRANCH IF SO
  496  002422' 020203                           CMP     R2,R3           ;OVER 32 BAD SECTORS RECORDED ?
  497  002424' 103001                           BHIS    3$              ;BRANCH IF SO
  498  002426' 000755                           BR      1$              ;LOOP BACK
  499  002430' 104403  007044'  000000'  3$:     MSGN$,MES14,BEGIN       ;MESSAGE: OVER 32 BAD SECTORS
  500  002436' 000167  000400                   JMP     FINI            ;DROP THE MODULES
  501  002442' 004767  002554           4$:     JSR     PC,FDUNIT       ;FOUND NEXT DRIVE
  502  002446' 132767  000010  004424           BITB    #BIT3,FLAG      ;DRVIE AVAILABLE ?
  503  002454' 001006                           BNE     5$              ;BRANCH IF NOT
  504  002456' 004567  003020                   JSR     R5,READY        ;LET DRIVE READY
  505  002462' 000653                           BR      BADFL           ;BRANCH IF DRIVE READY
  506  002464' 004767  002610                   JSR     PC,NOTRDY       ;TRY DRIVE NOT READY
  507  002470' 000650                           BR      BADFL           ;BRANCH IF SUCCESSFUL
  508  002472' 012767  177777  176072  5$:     MOV     #-1,UNITNO      ;RESET UNIT NUMBER
  509                                           ;ALL ON LINE DRIVE HAS BEEN RETRIVED THE BAD SPOT FILE
  510  002500' 000425                           BR      RSTRT1          ;+ SUPPORT - DT03
  511
```

```
002502' 105267  176046          RETRX:  INCB    DSKADR          ;ADJUST 2 SECTOR
002506' 105267  176042                  INCB    DSKADR          ;ADJUST TWO SECTOR
002512' 122767  000036  176034          CMPB    #30.,DSKADR     ;SECTORS 0,2,4,.....30, ALL TRIED
002520' 101263                          BHI     BADFL2          ;BRANCH IF NOT
002522' 104403  007050' 000000'         MSGNS,MES15,BEGIN       ;MESSAGE : RETRIEVE FAILS
002530' 000167  000306                  JMP     FINI            ;DROP THE MODULES
```



```
002534' 005767  175732          RFSTRT: TST     CNT1            ;+      / SUPPORT
002540' 001005                          BNE     RSTRT1          ;+     /  FOR
002542' 005767  175722                  TST     CNT             ;+    /   DT03
002546' 001002                          BNE     RSTRT1          ;+   /    BUS
002550' 000167  177106                  JMP     START           ;+  /     SWITCH
002554'                         RSTRT1:                         ;+/       OPTION
002554' 104415  000000' 000124'         GETPAS,BEGIN, RBUFVA    ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
002562' 016767  175344  176000          MOV     RBUFSZ,WCNT2    ; SAVE READ BUFFER SIZE
002570' 005467  175774                  NEG     WCNT2           ; GET THE 2'S COMPLEMENT
002574' 016767  175660  175760  STRT:   MOV     BLK1,RLKSAV     ;INIT BLOCK COUNTER
002602'                         STRT1:
002602' 104414  000000'                 GWBUFS, BEGIN           ;GET WRITE BUFFER INFORMATION
002606' 016767  175330  175752          MOV     WBUFSZ,WCNT1    ; SAVE WRITE BUFFER SIZE
002614' 005467  175746          2$:     NEG     WCNT1           ; GET THE 2'S COMPLEMENT
002620' 004767  002376          NEXT:   JSR     PC,FDUNIT       ; FIND UNIT #
002624' 016777  175742  176756          MOV     UNITNO,@RMCS2   ;GET UNIT # SO BOTH PORTS
                                                                ;WILL BE LOOKING FOR SAME DRIVE
002632' 005767  175720                  TST     DVICE           ; ANY DRIVES LEFT ?
002636' 001002                          BNE     1$              ; YES
002640' 000167  000176                  JMP     FINI            ; NO DROP MODULE
002644' 132767  000010  004226  1$:     BITB    #BIT3,FLAG      ;MORE DRIVES ON SYS?
002652' 001404                          BEQ     ABW             ;YES
002654' 012767  177777  175710          MOV     #-1,UNITNO              ;RESET DRIVE NUMBER
002662' 000744                          BR      STRT            ;YES CORRECT SETUP FOR DUEL PORT RETURN
002664' 004767  002250          ABW:    JSR     PC,RM03CK       ;UPDATE DSK ADDR
002670' 004567  002606                  JSR     R5,READY        ; IS DRIVE READY ?
002674' 000402                          BR      1$              ; YES, CONTINUE
002676' 004767  002376                  JSR     PC,NOTRDY       ; NOT READY, GO WAIT UNTIL IT IS
002702' 005067  004156          1$:     CLR     TRY1            ; ZERO RETRY COUNTERS
002706' 005067  004154                  CLR     TRY3            ;
002712' 005067  004152                  CLR     TRY5
002716' 005067  004150                  CLR     TRY7
002722' 005067  004146                  CLR     TRY9
002726' 005067  004144                  CLR     TRY11
002732' 004567  000466                  JSR     R5,UPDATT               ;TO SEE IF RM03 OR RM02
                                                                        ;FOR THE SELECTED DRIVE
```

```
                                GOC:    JSR     R5,WRITE        ; WRITE SOME DATA
                                        JMP     RETRY1          ; IF ERRORS, TRY IT AGAIN
                                        BITB    #BIT2,FLAG      ; DISK OVERFLOW ?
                                        BEQ     GOA             ; NO, CONTINUE
                                DAOCL:  BICB    #BIT2,FLAG      ; YES, CLEAR OVERFLOW FLAG
                                        MOV     #-1,BLK1        ; RESET BLOCK #
                                        JMP     STRT            ; CONTINUE
                                GOA:    JSR     R5,WRITCK       ; WRITE-CHECK THE DATA
                                        JMP     RETRY2          ; IF ERRORS, TRY AGAIN
                                GOB:    JSR     R5,READ         ; READ THE DATA WRITTEN
                                        JMP     RETRY3          ; IF ERRORS, TRY AGAIN
                                        CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
                                        .+2                     ; IF ERROR, CONTINUE
                                CYCLE:
                                        INC     CNT1            ;+BUMP INDICATOR FOR DT03
                                        ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
                                        JMP     NEXT
                                FINI:
                                        ENDS$,BEGIN             ; DROP THE MODULE
                                        ;-----------------------
                                NXT1:   JMP     NEXT            ;GET NEXT DRIVE
                                RETRY1: TST     BADSEC          ;BAD SECTOR DETECTED ?
                                        BEQ     .+10.           ;DETORE
                                        CLR     BADSEC          ;CLEAR THE FLAG
                                        JMP     NEXTA           ;EXIT
                                        INCB    TRY1            ; COUNT THE RETRYS
                                        CMPB    #3,TRY1         ; LIMIT EXCEEDED
                                        BEQ     1$              ; YES
                                        JMP     GOC             ; NO RETRY
                                1$:
                                        MSGN$,BEGIN,EXCED1      ;ASCII MESSAGE CALL WITH COMMON HEADER
                                        JMP     NEXTA           ; GO TO NEXT DRIVE
                                RETRY2: INCB    TRY2            ; COUNT THE RETRYS
                                        CMPB    #3,TRY2         ; LIMIT EXCEEDED
                                        BEQ     1$              ; YES
                                        JMP     GOA             ; NO RETRY
                                1$:
                                        MSGN$,BEGIN,EXCED2      ;ASCII MESSAGE CALL WITH COMMON HEADER
                                        JMP     NEXTA           ; GO TO NEXT DRIVE
                                RETRY3: INCB    TRY3            ; COUNT THE RETRYS
                                        CMPB    #3,TRY3         ; LIMIT EXCEEDED
                                        BEQ     1$              ; YES
                                        JMP     GOB             ; NO RETRY
                                1$:
                                        MSGN$,BEGIN,EXCED3      ;ASCII MESSAGE CALL WITH COMMON HEADER
                                        JMP     NEXTA           ; GO TO NEXT DRIVE
```



```
                                NEXTA:  BIT     #BIT0,SR1       ; DROP THE DRIVE
                                        BEQ     1$              ; NO, SKIP TO NEXT DRIVE
                                        JSR     PC,DROP         ; YES, DROP OFFENDING DRIVE
                                        MSGN$,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
                                1$:     JMP     NEXT            ; GO ON TO NEXT DRIVE
                                        ;-----------------------


                                        .MACRO  LINEUP  EABITS  ; LINE UP EA BITS FOR RHCS1
                                ;       LINEUP  EABITS
                                        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
                                        .ENDM   LINEUP
```

```
                                        ;               ----- RM DISK DRIVERS -----
                                        ;
003234' 012767  000161  175232  WRITE:  MOV     #161,FUNC       ; LOAD WRITE FUNCTION
003242' 016777  175320  176332          MOV     WCNT1,@RMWC     ; LOAD WORD COUNT
003250' 016777  174660  176325          MOV     WBUFPA,@RMBA    ; LOAD BUFFER ADDRESS
003256' 016777  175272  176322          MOV     DSKADR,@RMDA    ; LOAD DISK ADDRESS
003264' 004767  001650                  JSR     PC,RM03CK       ; CHECK FOR RM03 DRIVE
003270' 016777  175166  176336          MOV     CYL,@RMDC       ; LOAD CYLINDER ADDRESS
003276' 016767  175162  175156          MOV     CYLSAV,CYL      ; RESTORE CALCULATED CYLINDER ADDRESS
                                        ;       LINEUP  WBUFEA
003304' 016700  174626                  MOV     WBUFEA,R0       ; GET EXTENDED MEMORY BITS
003310' 006300                          ASL     R0              ; SHIFT 4 PLACES TO THE LEFT
003312' 006300                          ASL     R0              ; TO LINE UP WITH RHCS1
003314' 006300                          ASL     R0              ;
003316' 006300                          ASL     R0              ;
003320' 010067  175162                  MOV     R0,XMEM         ; SAVE THE SHIFTED BITS
003324' 000167  000300                  JMP     GOGO            ; CONTINUE
003330' 012767  000151  175136  WRITCK: MOV     #151,FUNC       ; LOAD WRITE-CHECK FUNCTION
003336' 016777  175224  176236          MOV     WCNT1,@RMWC     ; LOAD WORD COUNT
003344' 016777  174564  176232          MOV     WBUFPA,@RMBA    ; LOAD BUFFER ADDRESS
003352' 016777  175176  176226          MOV     DSKADR,@RMDA    ; LOAD DISK ADDRESS
003360' 004767  001554                  JSR     PC,RM03CK       ; CHECK FOR RM03 DRIVE
003364' 016777  175072  176242          MOV     CYL,@RMDC       ; LOAD CYLINDER ADDRESS
003372' 016767  175066  175062          MOV     CYLSAV,CYL      ; RESTORE CALCULATED CYLINDER ADDRESS
                                        ;       LINEUP  WBUFEA
003400' 016700  174532                  MOV     WBUFEA,R0       ; GET EXTENDED MEMORY BITS
003404' 006300                          ASL     R0              ; SHIFT 4 PLACES TO THE LEFT
003406' 006300                          ASL     R0              ; TO LINE UP WITH RHCS1
003410' 006300                          ASL     R0              ;
003412' 006300                          ASL     R0              ;
003414' 010067  175066                  MOV     R0,XMEM         ; SAVE THE SHIFTED BITS
003420' 000167  000204                  JMP     GOGO            ; CONTINUE
003424' 004767  001010  UPDATT: JSR     PC,BLOCK
003430' 000205                          RTS     R5
                                        ;
003432' 012767  000171  175034  READ:   MOV     #171,FUNC       ; LOAD READ FUNCTION
003440' 016777  175124  176134          MOV     WCNT2,@RMWC     ; LOAD WORD COUNT
003446' 016777  174454  176130          MOV     RBUFPA,@RMBA    ; LOAD BUFFER ADDRESS
003454' 016777  175074  176124          MOV     DSKADR,@RMDA    ; LOAD DISK ADDRESS
003462' 004767  001452                  JSR     PC,RM03CK       ; CHECK FOR RM03 DRIVE
003466' 016777  174770  176140          MOV     CYL,@RMDC       ; LOAD CYLINDER ADDRESS
003474' 016767  174764  174760          MOV     CYLSAV,CYL      ; RESTORE CALCULATED CYLINDER ADDRESS
                                        ;       LINEUP  RBUFEA
003502' 016700  174422                  MOV     RBUFEA,R0       ; GET EXTENDED MEMORY BITS
003506' 006300                          ASL     R0              ; SHIFT 4 PLACES TO THE LEFT
003510' 006300                          ASL     R0              ; TO LINE UP WITH RHCS1
003512' 006300                          ASL     R0              ;
003514' 006300                          ASL     R0              ;
003516' 010067  174764                  MOV     R0,XMEM         ; SAVE THE SHIFTED BITS
003522' 000442                          BR      GOGO            ; CONTINUE
```



```
003524' 016777  175042  176056  CLEAR:  MOV     UNITNO,@RMCS2   ; LOAD UNIT ADDRESS
003532' 012777  000011  176040          MOV     #11,@RMCS1      ; ISSUE A DRIVE CLEAR
003540' 000240                          NOP                     ;WAIT
003542' 000240                          NOP                     ;FOR DRIVE CLEAR TO FINISH
003544' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
003550' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
003554' 012777  000021  176016          MOV     #21,@RMCS1      ;ISSUE A PACK ACK
003562' 032777  000200  176010  1$:     BIT     #BIT7,@RMCS1            ; FUNCTION DONE ?
003570' 001005                          BNE     2$              ; YES, CONTINUE
003572' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
003576' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
003602' 000767                          BR      1$              ; NO, WAIT TILL DONE
003604' 012777  177777  176004  2$:     MOV     #-1,@RMAS       ;CLEAR AS BIT
003612' 012777  040000  175760          MOV     #BIT14,@RMCS1   ; CLEAR ANY CONTROLLER ERRORS
003620' 016777  174752  176004          MOV     MOD1,@RMOF      ; SET BIT FOR 11 FORMAT
003626' 000205                          RTS     R5              ; RETURN
                                        ;
003630' 016777  174736  175752  GOGO:   MOV     UNITNO,@RMCS2   ; LOAD UNIT SELECT
003636' 012777  003764' 174144          MOV     #NTRUPT,@VECTOR ; SET INTERUPT ENTRY POINTER
003644' 032767  001000  174204          BIT     #ADDR22,RES1    ;11/70?
003652' 001434                          BEQ     1$              ;NO
003654' 017767  175724  174622          MOV     @RMBA,PA18      ;GET 18 BIT ADDR
003662' 006267  174620                  ASR     XMEM            ;SHIFT EA BITS TO POSITION 4,5
003666' 006267  174614                  ASR     XMEM
003672' 006267  174610                  ASR     XMEM
003676' 006267  174604                  ASR     XMEM
003702' 104416  000000  000504'         MAP22$, BEGIN,PA18              ; GET 22-BIT ADDR FROM 18-BIT ADDR
003710' 016777  174574  175666          MOV     PA22,@RMBA      ;LOAD BA REG
003716' 016777  174570  175724          MOV     EA22,@RMBAE     ;LOAD BAE REG
003724' 042767  000034  174560          BIC     #34,EA22        ;CLEAR UNWANTED BITS
003732' 000367  174554                  SWAB    EA22            ;LOAD INTO BITS 8,9
003736' 016767  174550  174542          MOV     EA22,XMEM       ;LOAD XMEM TO SET INTO FUNCTION CODE
003744' 056767  174536  174522  1$:     BIS     XMEM,FUNC       ;LOAD EXTENDED MEMORY BITS
003752' 016777  174516  175620          MOV     FUNC,@RMCS1     ;EXECUTE THE FUNCTION
003760' 104400  000000'                 EXIT$,BEGIN             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
                                        ;
003764'                         NTRUPT:
                                        ;-----------------------------------------------------------------
003764' 000004  000000' 003772'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-----------------------------------------------------------------
                                        ;
003772' 004567  000116  1$:     JSR     R5,ERRORS       ; GO CHECK FOR ERRORS
003776' 000205                  RTS     R5              ; ERRORS DETECTED, RETURN
004000' 005725                  TST     (R5)+           ; NO ERRORS, SKIP RETRY
004002' 005725                  TST     (R5)+
004004' 000205                  RTS     R5              ; RETURN OK
```

```
 716  004006' 016700  174446          ROOM:   MOV     BLK1,R0         ; SAVE THE CURRENT BLOCK NUMBER
 717  004012' 016703  174124                  MOV     WBUFS7,R3       ; GET THE TRANSFER SIZE
 718  004016' 132767  000002  003054          BITB    #BIT1,FLAG      ; PLENTY OF ROOM LEFT ?
 719  004024' 001406                          BEQ     4$              ; YES, CONTINUE
 720  004026' 142767  000001  003044          BICB    #BIT0,FLAG      ; CLEAR 32K INDICATOR
 721  004034' 016701  174464                  MOV     MHICY,R1        ; LOAD MAX. NUMBER OF BLOCKS
 722  004040' 001402                          BEQ     5$              ;EXHAUST OF ALL BLOCKS
 723  004042' 005725                  4$:     TST     (R5)+           ; YES, MUST BE A REAL ERROR
 724  004044' 000205                          RTS     R5              ; RETURN, ERROR
 725  004046' 152767  000004  003024  5$:     BISB    #BIT2,FLAG      ; SET OVERFLOW FLAG
 726  004054' 000205                          RTS     R5              ; RETURN OK
 727                                  ;       -----------------------
 728
 729  004056' 012701  000001          DROP:   MOV     #1,R1           ; INITIALIZE DROP PICKER
 730  004062' 016700  174504                  MOV     UNITNO,R0       ; GET THE DRIVE NUMBER
 731  004066' 001403                          BEQ     2$              ; IF DRIVE 0 GO DROP IT
 732  004070' 006301                  1$:     ASL     R1              ; POINT TO NEXT DRIVE
 733  004072' 005300                          DEC     R0              ; IS THIS THE ONE ?
 734  004074' 001375                          BNE     1$              ; NO, LOOK AGAIN
 735  004076' 040167  174454          2$:     BIC     R1,DVICE        ; DROP THE DRIVE
 736                                          ;*************************************************************
 737                                                                  ;CONVERT UNITNO TO ASCII AND
 738                                                                  ;STORE AT ADR1
 739  004102' 104420  000000' 000572'         OTOA$,BEGIN,UNITNO,ADR1
 740  004110' 007054'
 741                                          ;*************************************************************
 742  004112' 000207                          RTS     PC              ; RETURN
 743                                  ;       -----------------------
```



```
 744  004114' 005067  174360          ERRORS: CLR     BADSEC          ;CLEAR THE BAD SECTOR FLAG
 745  004120' 005777  175454                  TST     @RMCS1          ; ATTENTION OR ERROR ?
 746  004124' 100402                          BMI     22$             ;YES
 747  004126' 000167  000302                  JMP     RESYNC          ;NO GO ON TO NEXT FUNTION
 748  004132' 032777  001000  175454  22$:    BIT     #BIT9,@RMER1    ; ADDRESS OVERFLOW ?
 749  004140' 001403                          BEQ     1$              ; NO, CONTINUE
 750  004142' 004567  177640                  JSR     R5,ROOM         ; YES, IS IT A REAL ERROR ?
 751  004146' 000532                          BR      RESYNC          ; NO, CONTINUE
 752  004150' 032777  002000  175434  1$:     BIT     #BIT10,@RMDS    ;DID LBT SET?
 753  004156' 001126                          BNE     RESYNC          ;YES
 754  004160' 004767  001430                  JSR     PC,ERSUB1       ; LOAD ERROR INFORMATION
 755  004164' 005777  175420                  TST     @RMCS2          ;IS THIS A DATA LATE ERROR?
 756  004170' 100012                          BPL     11$             ;NO
 757  004172' 005267  174256                  INC     DLTCNT          ;ADD 1 TO DATA LATE COUNTER
 758  004176' 032767  000004  173612          BIT     #BIT2,SR1       ;TYPE ERROR AND COUNT IT?
 759  004204' 001102                          BNE     8$              ;NO
 760  004206' 104403  000000' 007010'         MSGN$,BEGIN,DLTERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
 761  004214' 000450                          BR      5$              ;CONT
 762  004216' 032777  020000  175354  11$:    BIT     #BIT13,@RMCS1   ; MASSBUS CONTROL PARITY ERROR ?
 763  004224' 001035                          BNE     3$              ; YES
 764  004226' 032777  000400  175354          BIT     #BIT8,@RMCS2    ; MASSBUS DATA PARITY ERROR ?
 765  004234' 001035                          BNE     4$              ; YES
 766  004236' 032777  040000  175334          BIT     #BIT14,@RMCS1   ; TRANSFER ERROR ?
 767  004244' 001015                          BNE     2$              ; YES
 768  004246' 032777  040000  175336          BIT     #BIT14,@RMDS    ; ANY DRIVE ERRORS ?
 769  004254' 001030                          BNE     5$              ; YES
 770  004256' 005777  175334                  TST     @RMAS           ; ANY ATTENTIONS ACTIVE ?
 771  004262' 001025                          BNE     5$              ; YES, CONTINUE
 772  004264' 005067  173616                  CLR     ERRTYP          ;UNKNOWN ERROR
 773                                          ;*************************************************************
 774  004270' 104405  000000' 001600'         HRDER$,BEGIN,TABLE      ; SPECIAL CONDITION SET BUT NO REASON FOUND
 775                                          ;*************************************************************
 776  004276' 000445                          BR      8$              ; RETURN
 777  004300'                         2$:
 778  004300' 032777  100000  175334          BIT     #BIT15,@RMER2   ;A BAD SPOT ? ******
 779  004306' 001044                          BNE     9$              ;BRANCH IF SO *******
 780  004310' 104403  000000' 006752'         MSGN$,BEGIN,TRERR       ;ASCII MESSAGE CALL WITH COMMON HEADER
 781  004316' 000407                          BR      5$              ; GO DUMP REGISTERS
 782  004320'                         3$:
 783  004320' 104403  000000' 006756'         MSGN$,BEGIN,MCPERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
 784  004326' 000403                          BR      5$              ; GO DUMP REGISTERS
 785  004330'                         4$:
 786  004330' 104403  000000' 006762'         MSGN$,BEGIN,MDPERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
 787  004336' 005777  175254          5$:     TST     @RMAS           ; ANY ATTENTIONS ACTIVE ?
 788  004342' 001402                          BEQ     6$              ; NO, CONTINUE
 789  004344' 004767  001060                  JSR     PC,WHO          ; YES, FIND OUT WHICH DRIVE IT IS
 790  004350' 016700  175246          6$:     MOV     RMDB,R0         ; SAVE ADDRESS OF DATA BUFFER
 791  004354' 032777  000200  175226          BIT     #BIT7,@RMCS2    ; CAN DATA BUFFER BE READ ?
 792  004362' 001003                          BNE     7$              ; YES, CONTINUE
 793  004364' 012767  000552' 175230          MOV     #ZERO,RMDB      ; NO, LOAD ADDRESS OF ZERO
 794  004372' 012767  000001  173506  7$:     MOV     #1,ERRTYP       ;DATA ERROR
 795                                          ;*************************************************************
 796  004400' 104406  000000' 001600'         SOFER$,BEGIN,TABLE      ; DUMP RH11 AND RM REGISTERS
 797                                          ;*************************************************************
 798  004406' 010067  175210                  MOV     R0,RMDB         ; RESTORE DATA BUFFER ADDRESS
 799  004412' 004567  177106          8$:     JSR     R5,CLEAR        ;GO CLEAR OUT ERRORS
```

```
R00  004416' 000205                        RTS     R5              ; ERRORS DETECTED, RETURN
R01  004420' 004567  177100        9$:     JSR     R5,CLEAR        ;CLEAR THE BAD SPOT ERROR
R02  004424' 012767  177777  174046        MOV     #-1,BADSEC      ;SET THE BAD SPOT ERROR FLAG
R03  004432' 000205                        RTS     R5              ;EXIT
```



```
R04  004434' 005725                RESYNC: TST     (R5)+           ;SKIP RETRY
R05  004436' 000205                        RTS     R5              ; RETURN OK
R06
R07
R08                                ;THIS ROUTINE DETERMINES IF THERE IS ENOUGH ROOM ON THE DISK TO
R09                                ;DO ANOTHER TRANSFER. IF NOT, PROGRAM GOES TO RESYNC TO
R10                                ;ALLOW BOTH PROCESSORS TO RESYNC AND TO RESTART.
R11
R12  004440' 005267  174014        BLOCK:  INC     BLK1            ; STEP TO NEXT BLOCK
R13  004444' 132767  000002  002426        BITB    #BIT1,FLAG      ; BLOCK # IN CYLINDER 410. OR HIGHER ?
R14  004452' 001012                        BNE     1$              ; YES, GO ADJUST BLOCK #
R15  004454' 026767  174042  173776        CMP     MLOWCY,BLK1     ; BLOCK # IN CYLINDER 410. OR LOWER ?
R16  004462' 101017                        BHI     2$              ; YES, RETURN
R17  004464' 005067  173770                CLR     BLK1            ; NO, RESET BLOCK #
R18  004470' 152767  000002  002402        BISB    #BIT1,FLAG      ; SET HIGH RANGE FLAG
R19  004476' 000411                        BR      2$              ; RETURN
R20
R21  004500' 026767  174020  173752  1$:   CMP     MHICY,BLK1      ; BLOCK # IN WITHIN RANGE ?
R22  004506' 101005                        BHI     2$              ; YES, RETURN
R23  004510' 005067  173744                CLR     BLK1            ; NO, RESET BLOCK #
R24  004514' 142767  000002  002355        BICB    #BIT1,FLAG      ; SET FLAG TO LOWER RANGE FOR NEW PACK CYCLE
R25  004522' 016700  173732          2$:   MOV     BLK1,R0         ;TRANSFER PAREMETER FOR CONVRT
R26  004526' 005767  174042                TST     MIXDV           ;IS AN RM03/RM02 ?
R27  004532' 001432                        BEQ     CONVRT          ;BRANCH IF SO
R28  004534' 016777  174032  175046        MOV     UNITNO,@RMCS2   ;LOAD THE DRIVE ADDRESS
R29  004542' 005067  173714                CLR     CYL             ;RESET CYLINDER ADDRESS
R30  004546' 005067  173712                CLR     CYLSAV          ;RESET THE SAVED CYLINDER ADDRESS
R31  004552' 005067  173776                CLR     DSKADR          ;RESET THE TRACK AND SECTOR ADDRESS
R32  004556' 022777  000620  175050        CMP     #400.,@RMDC     ;OVER CYL 400 ?
R33  004564' 101414                        BLOS    3$              ;BRANCH IF SO
R34  004566' 017767  175042  173666        MOV     @RMDC,CYL       ;
R35                                ;       MOV     @RMDC,CYLSAV    ;
R36  004574' 062767  000001  173660        ADD     #1,CYL          ;THE NUMBER CAN MODIFY TO INCREMENT
R37                                                                ;THE CYLINDER ADDRESS
R38  004602' 017767  175000  173744        MOV     @RMDA,DSKADR    ;
R39  004610' 017767  174772  173736        MOV     @RMDA,DSKADR    ;
R40  004616' 000207                  3$:   RTS     PC
```

```
 841  004620' 005067  173636          CONVRT: CLR     CYL             ; CLEAR DISK ADDRESSES
 842  004624' 005067  173636                  CLR     SEC             ;
 843  004630' 105067  002245                  CLRB    TRK             ;
 844  004634' 016701  173656                  MOV     MBLKTR,R1       ; LOAD REG. 1 WITH BLK PER TRACK
 845                                                                  ; REG 0 HAS BLK1 (BLOCK #) IN IT
 846  004640' 016702  173650                  MOV     MBLKRV,R2       ; LOAD REG. 2 WITH BLK PER REVOLUTION
 847  004644' 132767  000002  002226          BITB    #BIT1,FLAG      ; BLOCK # ON CYLINDER 410. OR LOWER ?
 848  004652' 001403                          BEQ     1$              ; YES, CONTINUE
 849  004654' 012767  000632  173600          MOV     #410.,CYL       ; NO, LOAD HIGH BASE VALUE FOR CYLINDER ADR.
 850  004662' 020001                  1$:     CMP     R0,R1           ; CORRECT CYLINDER FOUND ?
 851  004664' 103413                          BLO     2$              ; YES, CONTINUE
 852  004666' 005267  173570                  INC     CYL             ; NO, STEP TO NEXT CYLINDER
 853  004672' 160100                          SUB     R1,R0           ; SUBTRACT 1 CYLINDER FROM BLOCK #
 854  004674' 000772                          BR      1$              ; CONTINUE UNTIL CORRECT CYLINDER IS FOUND
 855  004676' 022767  001440  173556          CMP     #800.,CYL       ;NOT OVER CYL 800 FOR PROTECT BAD SE.FILE
 856  004704' 101003                          BHI     2$              ;
 857  004706' 012767  000632  173546          MOV     #410.,CYL       ;RESET TO 410
 858  004714' 020002                  2$:     CMP     R0,R2           ; CORRECT TRACK FOUND ?
 859  004716' 002404                          BLT     3$              ; YES, CONTINUE
 860  004720' 105267  002155                  INCB    TRK             ; NO, STEP TO NEXT TRACK
 861  004724' 160200                          SUB     R2,R0           ; SUBTRACT 1 TRACK FROM BLOCK #
 862  004726' 000772                          BR      2$              ; CONTINUE UNTIL CORRECT TRACK IS FOUND
 863  004730' 005700                  3$:     TST     R0              ; CORRECT SECTOR FOUND ?
 864  004732' 001405                          BEQ     4$              ; YES, CONTINUE
 865  004734' 062767  000004  173524          ADD     #4,SEC          ; NO, STEP TO NEXT 1024 WORDS
 866  004742' 005300                          DEC     R0              ; DECREASE BLOCK # BY 1
 867  004744' 000771                          BR      3$              ; CONTINUE UNTIL CORRECT SECTOR IS FOUND
 868  004746' 122767  000005  002125  4$:     CMPB    #5,TRK          ; LAST TRACK ?
 869  004754' 003007                          BGT     5$              ; NO, CONTINUE
 870  004756' 026767  173536  173502          CMP     MODE,SEC        ; YES, LAST SECTOR ?
 871  004764' 003003                          BGT     5$              ; NO, CONTINUE
 872  004766' 162767  000004  173472          SUB     #4,SEC          ; YES, ADJUST SECTOR SO NO OVERFLOW
 873  004774' 016767  173466  173552  5$:     MOV     SEC,DSKADR      ; LOAD SECTOR INTO DISK ADDRESS
 874  005002' 116767  002073  173545          MOVB    TRK,DSKADR+1    ; LOAD TRACK INTO DISK ADDRESS
 875                                                                  ;CKECK THE THE SELECT ADDRESS IS  IN THE BAD SPOT ?
 876  005010' 005000                          CLR     R0              ;INDEX VALUE TO THE BAD SPOT TABLE
 877  005012' 026760  173444  000254' BADLOP: CMP     CYL,BADSPT(R0)  ;CYLINDER ADDRESS IN THE SPOT TABLE?
 878  005020' 001403                          BEQ     1$              ;YES
 879  005022' 062700  000004                  ADD     #4,R0           ;CHECK THE NEXT SPOT
 880  005026' 000434                          BR      5$              ;COMMON BRANCH POINT
 881  005030' 062700  000002          1$:     ADD     #2,R0           ;CHECK THE TRAK AND SEC
 882  005034' 126760  002041  000255' 2$:     CMPB    TRK,BADSPT+1(R0)  ;ON THE BAD SPOT TRACK ?
 883  005042' 001403                          BEQ     2$              ;YES
 884  005044' 062700  000002                  ADD     #2,R0           ;CHECK THE NEXT SPOT
 885  005050' 000423                          BR      5$              ;COMMON BRANCH POINT
 886  005052' 126760  173410  000254' 2$:     CMPB    SEC,BADSPT(R0)  ;CHECK THE SECTOR ADDRESS IS IN THIS BLOCK RANGE ?
 887  005060' 101015                          BHI     4$              ;NO, NOT IN THIS BLOCK
 888  005062' 062767  000003  173376          ADD     #3,SEC          ;WITHIN THE BLOCK ?
 889  005070' 126760  173372  000254'          CMPB    SEC,BADSPT(R0)  ;NOT IN THE BLOCK IF LESS
 890  005076' 103403                          BLO     3$              ;
 891  005100' 012700  177777                  MOV     #-1,R0          ;ERROR FLAG
 892  005104' 000405                          BR      5$              ;COMMON BRANCH POINT
 893  005106' 162767  000003  173352  3$:     SUB     #3,SEC          ;RESTORE THE SECTOR VALUE
 894  005114' 062700  000002          4$:     ADD     #2,R0           ;CHECK NEXT BAD SPOT
 895  005120' 005700                  5$:     TST     R0              ;MINUS SIGN SET, IF ON A BAD SPOT
 896  005122' 100002                          BPL     .+6             ;SKIP CURRENT SELECTED BLOCK IF BAD SPOT
```



```
 897  005124' 000167  177310                  JMP     BLOCK           ;SELECT OTHER BLOCK
 898  005130' 022700  000200                  CMP     #128.,R0        ;END OF TABLE
 899  005134' 101326                          BHI     BADLOP          ;NO, THEN BRANCH BACK
 900  005136' 000207                          RTS     PC              ; RETURN
 901                                  ;       ------------------------
 902
 903  005140' 016767  173316  173316  RM03CK: MOV     CYL,CYLSAV      ;SAVE THE CALCULATED ADDRESS
 904  005146' 005067  173422                  CLR     MIXDV           ;RESET MIX DRIVE FLAG
 905  005152' 022777  024025  174446          CMP     #24025,@RMDT    ;DUAL PROT RM02 ?
 906  005160' 001417                          BEQ     1$              ;BRANCH IF SO
 907  005162' 022777  020025  174436          CMP     #20025,@RMDT    ;SINGLE PORT RM02 ?
 908  005170' 001413                          BEQ     1$              ;BRANCH IF SO
 909  005172' 022777  024024  174426          CMP     #24024,@RMDT    ;DUAL PORT RM03 ?
 910  005200' 001407                          BEQ     1$              ;YES
 911  005202' 022777  020024  174416          CMP     #20024,@RMDT    ;NO...SINGLE PORT RM03 ?
 912  005210' 001403                          BEQ     1$              ;YES
 913                                  ;       JSR     PC,DROP         ;DROP THE DRIVE ;IF NOT A RM03
 914                                  ;       MSGN$,DRPI,BEGIN        ;MESSAGE - NOT A RM03
 915                                  ;       TST     (SP)+           ;CLEAR STACK FOR WHOMEVER CALLING RM03CK
 916                                  ;       MOV     (SP)+,R5        ;CLEAR STACK FOR WHOMEVER CALLING "WRITE",
 917                                  ;                               ;"WRITCK" AND "READ" ROUTINES
 918                                  ;       JMP     NEXT            ;TO NEXT DRIVE
 919  005212' 012767  177777  173354          MOV     #-1,MIXDV       ;SET NON-RM03,RM02 DRIVE FLAG
 920  005220' 000207                  1$:     RTS     PC
 921                                  ;       ------------------------
 922
```

```
 923  005222' 005267  173344          FDUNIT: INC     UNITNO          ; COUNT A DRIVE
 924  005226' 142767  000010  001644          BICB    #BIT3,FLAG      ; CLEAR END OF DRIVES FLAG
 925  005234' 022767  000010  173330          CMP     #8.,UNITNO      ; ALL DRIVES CHECKED ?
 926  005242' 003404                          BLE     1$              ; YES, GO FLAG END OF DRIVES
 927  005244' 006267  173310                  ASR     DRIVE           ; NO, IS NEXT DRIVE CHOSEN ?
 928  005250' 103364                          BCC     FDUNIT          ; NO, GO TRY ANOTHER DRIVE
 929  005252' 000411                          BR      2$              ; RETURN
 930  005254' 152767  000010  001616  1$:     BISB    #BIT3,FLAG      ; SET END OF DRIVES FLAG
 931  005262' 012767  177777  173302          MOV     #-1,UNITNO      ; RESET DRIVE COUNTER
 932  005270' 016767  173262  173262          MOV     DVICE,DRIVE     ; RESTORE CHOSEN DRIVES
 933  005276' 000207                  2$:     RTS     PC              ; RETURN
 934                                          ;       ------------------------
 935
 936
 937                                          ;SEIZE THE DRIVE BY READING CS1 REG
 938
 939  005300' 012767  077777  173150  NOTRDY: MOV     #77777,CLK      ; SET THE TIMER
 940  005306' 005777  174266          4$:     TST     @RMCS1          ;GRAB THE DRIVE
 941  005312' 032777  004000  174260          BIT     #BIT11,@RMCS1   ;DO I HAVE THE DRIVE DVA?
 942  005320' 001013                          BNE     2$              ;YES
 943  005322' 104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
 944  005326' 104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
 945  005332' 005367  173120                  DEC     CLK             ;COUNT # OF TRIES
 946  005336' 001363                          BNE     4$              ;NOT DONE YET
 947  005340' 104403  000000' 007014'         MSGN$,BEGIN,NOT ;ASCII MESSAGE CALL WITH COMMON HEADER
 948  005346' 000405                          BR      5$              ;COULD NOT GET DRIVE
 949  005350' 004567  176150          2$:     JSR     R5,CLEAR        ; RESET THE CONTROLLER AND DRIVE
 950  005354' 004567  000122                  JSR     R5,READY        ; IS DRIVE READY ?
 951  005360' 000422                          BR      1$              ; YES, CONTINUE
 952  005362' 004767  000226          5$:     JSR     PC,ERSUB1       ; LOAD ERROR INFORMATION
 953  005366' 012767  000006  172512          MOV     #6,ERRTYP       ;DRIVE NOT AVAILABLE
 954                                          ;*****************************************************************
 955  005374' 104405  000000' 001600'         HRDER$,BEGIN,TABLE      ; COULD NOT GET DRIVE ... TIME-OUT
 956                                          ;*****************************************************************
 957  005402' 012777  000013  174170          MOV     #13,@RMCS1      ;RELEASE DRIVE
 958  005410' 004767  176442                  JSR     PC,DROP         ; NO, DROP THE DRIVE
 959  005414' 104403  000000' 007024'         MSGN$,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
 960  005422' 000167  175414                  JMP     FINI            ;COULD NOT GET DRIVE DROP MODULE
 961  005426' 000207                  1$:     RTS     PC              ;RETURN
 962                                          ;       ------------------------
```



```
 963  005430' 017701  174162          WHO:    MOV     @RMAS,R1        ; GET THE ATTENTION SUMMARY
 964  005434' 017704  174150                  MOV     @RMCS2,R4       ; SAVE THE STATUS REGISTER
 965  005440' 012702  000001                  MOV     #BIT0,R2        ; SET POINTER TO DRIVE 0
 966  005444' 005003                          CLR     R3              ; ZERO THE DRIVE COUNTER
 967  005446' 030201                  1$:     BIT     R2,R1           ; FIND IT ?
 968  005450' 001006                          BNE     2$              ; YES, CONTINUE
 969  005452' 005203                          INC     R3              ; NO, INCREMENT THE DRIVE COUNTER
 970  005454' 006302                          ASL     R2              ; SET POINTER TO NEXT DRIVE
 971  005456' 032702  000400                  BIT     #BIT8,R2        ; ALL DONE ?
 972  005462' 001771                          BEQ     1$              ; NO, GO AGAIN
 973  005464' 000207                          RTS     PC              ; SOMEBODY LIED -- NO ATTENTIONS SET
 974  005466' 042704  000007          2$:     BIC     #7,R4           ; CLEAR OUT OLD UNIT NUMBER
 975  005472' 050304                          BIS     R3,R4           ; LOAD THE NEW UNIT NUMBER
 976  005474' 010477  174110                  MOV     R4,@RMCS2       ; RESTORE THE STATUS REGISTER
 977  005500' 000207                          RTS     PC              ; RETURN
 978                                          ;       ------------------------
 979
 980  005502' 016777  173064  174100  READY:  MOV     UNITNO,@RMCS2   ; LOAD UNIT ADDRESS
 981  005510' 017700  174076                  MOV     @RMDS,R0        ; SAVE STATUS IN R0
 982  005514' 105700                          TSTB    R0              ; DRIVE READY ?
 983  005516' 100022                          BPL     1$              ; NO
 984  005520' 032700  000100                  BIT     #BIT6,R0        ; VOLUME VALID ?
 985  005524' 001417                          BEQ     1$              ; NO
 986  005526' 032700  000400                  BIT     #BIT8,R0        ; DRIVE PRESENT ?
 987  005532' 001414                          BEQ     1$              ; NO
 988  005534' 032700  004000                  BIT     #BIT11,R0       ; WRITE LOCKED ?
 989  005540' 001011                          BNE     1$              ; YES
 990  005542' 032700  010000                  BIT     #BIT12,R0       ; MEDIUM ON LINE ?
 991  005546' 001406                          BEQ     1$              ; NO
 992  005550' 032700  040000                  BIT     #BIT14,R0       ; ANY ERRORS ?
 993  005554' 001003                          BNE     1$              ; YES
 994  005556' 005700                          TST     R0              ; ATTENTION SET ?
 995  005560' 100401                          BMI     1$              ; YES
 996  005562' 000205                          RTS     R5              ; RETURN READY
 997  005564' 005725                  1$:     TST     (R5)+           ; SKIP INSTRUCTION FOLLOWING CALL
 998  005566' 000205                          RTS     R5              ; RETURN AS NOT READY
 999                                          ;       ------------------------
1000
1001
1002  005570' 014167  172312          ERSUB2: MOV     -(R1),ASB       ; LOAD THE DATA
1003  005574' 010167  172302                  MOV     R1,SBADR        ; LOAD ADDRESS OF DATA WRITTEN
1004  005600' 014267  172304                  MOV     -(R2),WASAS     ; LOAD THE DATA
1005  005604' 010267  172274                  MOV     R2,WASADR       ; LOAD ADDRESS OF DATA READ
1006  005610' 005721                          TST     (R1)+           ; RESET REG. 1
1007  005612' 005722                          TST     (R2)+           ; RESET REG. 2
1008
1009  005614' 016767  173760  172256  ERSUB1: MOV     RMCS1,CSRA      ; LOAD ADR OF CURRENT CSR
1010  005622' 017767  173752  172252          MOV     @RMCS1,ACSR     ; LOAD CONTENTS OF CURRENT CSR
1011  005630' 000207                          RTS     PC              ; RETURN
```

```
1012
1013
1014
1015
1016
1017  005632' 012777  000040  173750  REZET:  MOV     #BIT5,@RMCS2    ; ISSUE AN RH11 INIT
1018  005640' 012777  177777  173750          MOV     #-1,@RMAS       ;CLEAR ALL ATA BITS
1019  005646' 012767  077777  172602          MOV     #77777,CLK      ; SET THE TIMER
1020  005654' 032777  000200  173716  1$:     BIT     #BIT7,@RMCS1    ; CONTROLLER READY ?
1021  005662' 001017                          BNE     2$              ; YES, CONTINUE
1022  005664' 104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
1023  005670' 104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
1024  005674' 005367  172556                  DEC     CLK             ; WAIT SOME MORE ?
1025  005700' 001365                          BNE     1$              ; YES
1026  005702' 005067  172650                  CLR     DVICE           ; NO, SET TO DROP THE MODULE
1027  005706' 012767  000004  172172          MOV     #4,ERRTYP       ;CONTROLLER NOT READY
1028                                          ;*****************************************************
1029  005714' 104405  000000' 001600'         ERDER$,BEGIN,TABLE      ; CONTROLLER NOT READY
1030                                          ;*****************************************************
1031  005722' 000207                  2$:     RTS     PC              ; RETURN
1032                                  ;       -------------------------
```



```
1033  005724' 016700  172056          SETUP:  MOV     ADDR,R0         ; GET DEVICE ADDRESS
1034  005730' 010067  173644                  MOV     R0,RMCS1        ; GENERATE REGISTER ADDRESSES
1035  005734' 005720                          TST     (R0)+
1036  005736' 010067  173640                  MOV     R0,RMWC
1037  005742' 005720                          TST     (R0)+
1038  005744' 010067  173634                  MOV     R0,RMBA
1039  005750' 005720                          TST     (R0)+
1040  005752' 010067  173630                  MOV     R0,RMDA
1041  005756' 005720                          TST     (R0)+
1042  005760' 010067  173624                  MOV     R0,RMCS2
1043  005764' 005720                          TST     (R0)+
1044  005766' 010067  173620                  MOV     R0,RMDS
1045  005772' 005720                          TST     (R0)+
1046  005774' 010067  173614                  MOV     R0,RMER1
1047  006000' 005720                          TST     (R0)+
1048  006002' 010067  173610                  MOV     R0,RMAS
1049  006006' 005720                          TST     (R0)+
1050  006010' 010067  173604                  MOV     R0,RMLA
1051  006014' 005720                          TST     (R0)+
1052  006016' 010067  173600                  MOV     R0,RMDB
1053  006022' 005720                          TST     (R0)+
1054  006024' 010067  173574                  MOV     R0,RMMR1
1055  006030' 005720                          TST     (R0)+
1056  006032' 010067  173570                  MOV     R0,RMDT
1057  006036' 005720                          TST     (R0)+
1058  006040' 010067  173564                  MOV     R0,RMSN
1059  006044' 005720                          TST     (R0)+
1060  006046' 010067  173560                  MOV     R0,RMOF
1061  006052' 005720                          TST     (R0)+
1062  006054' 010067  173554                  MOV     R0,RMDC
1063  006060' 005720                          TST     (R0)+
1064  006062' 010067  173550                  MOV     R0,RMHR
1065  006066' 005720                          TST     (R0)+
1066  006070' 010067  173544                  MOV     R0,RMMR2
1067  006074' 005720                          TST     (R0)+
1068  006076' 010067  173540                  MOV     R0,RMER2
1069  006102' 005720                          TST     (R0)+
1070  006104' 010067  173534                  MOV     R0,RMEC1
1071  006110' 005720                          TST     (R0)+
1072  006112' 010067  173530                  MOV     R0,RMEC2
1073  006116' 032767  001000  171732          BIT     #ADDR22,RES1    ;11/70 MONITOR?
1074  006124' 001417                          BEQ     1$              ;NO
1075  006126' 016700  171654                  MOV     ADDR,R0         ;LOCATE THE RMBAE
1076  006132' 062700  000050                  ADD     #50,R0          ;ASSUME 22 REGISTERS
1077  006136' 032767  100000  171652          BIT     #BIT15,SR1      ;32 REGISTER ?
1078  006144' 001402                          BEQ     3$              ;BRANCH IF NOT 32 REGISTERS
1079  006146' 062700  000024                  ADD     #24,R0          ;ADJUST THE RMBAE ADDRESS
1080  006152' 010067  173472          3$:     MOV     R0,RMBAE
1081  006156' 005720                          TST     (R0)+
1082  006160' 010067  173466                  MOV     R0,RMCS3
1083  006164' 016700  171620          1$:     MOV     VECTOR,R0       ; GET VECTOR ADDRESS
1084  006170' 012720  002574'                 MOV     #STRT,(R0)+     ; SET POINTER JUST IN CASE
1085  006174' 116710  171612                  MOVB    BRI,(R0)        ; SET PRIORITY
1086  006200' 000207                  2$:     RTS     PC              ; RETURN
```

```
1087  006202' 020040  051124  047101  MES1:   .ASCIZ  '  TRANSFER  ERROR%'
1088  006210' 043123  051105  020040
1089  006216' 051105  047522  022522
1090  006224'    000
1091  006225'    040  046440  051501  MES2:   .ASCIZ  '  MASSBUS  PARITY  ERROR%'
1092  006232' 041123  051525  020040
1093  006240' 040520  044522  054524
1094  006246' 020040  051105  047522
1095  006254' 022522     000
1096  006257'    040  046440  051501  MES3:   .ASCIZ  '  MASSBUS  DATA  PARITY  ERROR%'
1097  006264' 041123  051525  020040
1098  006272' 040504  040524  020040
1099  006300' 040520  044522  054524
1100  006306' 020040  051105  047522
1101  006314' 022522     000
1102  006317'    040  042040  044522  MES4:   .ASCIZ  '  DRIVE  '
1103  006324' 042526  020040     000
1104  006331'    040  042040  047522  MES5:   .ASCIZ  '  DROPPED%'
1105  006336' 050120  042105  000045
1106  006344' 051040  052105  054522  MES6:   .ASCIZ  ' RETRY EXCEEDED%'
1107  006352' 042440  041530  042505
1108  006360' 042504  022504     000
1109  006365'    040  020040  051127  MES7:   .ASCIZ  '   WRITE'
1110  006372' 052111  000105
1111  006376' 020040  053440  044522  MES8:   .ASCIZ  '   WRITE-CHECK'
1112  006404' 042524  041455  042510
1113  006412' 045503     000
1114  006415'    040  020040  042522  MES9:   .ASCIZ  '   READ'
1115  006422' 042101     000
1116  006425'    040  040504  040524  MES10:  .ASCIZ  ' DATA LATE ERROR%'
1117  006432' 046040  052101  020105
1118  006440' 051105  047522  022522
1119  006446'    000
1120  006447'    040  051104  053111  MES11:  .ASCIZ  ' DRIVE NOT READY%'
1121  006454' 020105  047516  020124
1122  006462' 042522  042101  022531
1123  006470'    000
1124  006471'    040  047503  046125  MES12:  .ASCIZ  ' COULD NOT GET DRIVE%'
1125  006476' 020104  047516  020124
1126  006504' 042507  020124  051104
1127  006512' 053111  022505     000
1128  006517'    137  057537  020137  MES13:  .ASCIZ  '____ NOT A RM03/RM02%'
1129  006524' 047516  020124  020101
1130  006532' 046522  031460  051057
1131  006540' 030115  022462     000
1132  006545'    045  051104  050117  MESLDP: .ASCIZ  '%DROPPED RMDP LOAD DRIVE%'
1133  006552' 042520  020104  046522
1134  006560' 050104  046040  040517
1135  006566' 020104  051104  053111
1136  006574' 022505     000
1137  006577'    045  053117  051105  XMES14: .ASCIZ  '%OVER 32 BAD SECTORS RECORDED,PACK NOT ACCEPTABLE%'
1138  006604' 031440  020062  040502
1139  006612' 020104  042523  052103
1140  006620' 051117  020123  042522
1141  006626' 047503  042122  042105
1142  006634' 050054  041501  020113
```



```
1143  006642' 047516  020124  041501
1144  006650' 042503  052120  041101
1145  006656' 042514  000045
1146  006662' 051045  052105  044522  XMES15: .ASCIZ  '%RETRIEVING THE BAD SPOT FILE FAILS-- DROP MODULE%'
1147  006670' 053105  047111  020107
1148  006676' 044124  020105  040507
1149  006704' 020104  050123  052117
1150  006712' 043040  046111  020105
1151  006720' 040506  046111  026523
1152  006726' 020055  051104  050117
1153  006734' 046440  042117  046125
1154  006742' 022505     000
1155          006746'                  .EVEN
1156  006746' 006545'                  LDRIVE: MESLDP
1157  006750' 177777                           -1
1158  006752' 006202'                  TRERR:  MES1
1159  006754' 177777                           177777
1160  006756' 006225'                  MCPERR: MES2
1161  006760' 177777                           177777
1162  006762' 006257'                  MDPERR: MES3
1163  006764' 177777                           177777
1164  006766' 006365'                  EXCED1: MES7
1165  006770' 006344'                          MES6
1166  006772' 177777                           177777
1167  006774' 006376'                  EXCED2: MES8
1168  006776' 006344'                          MES6
1169  007000' 177777                           177777
1170  007002' 006415'                  EXCED3: MES9
1171  007004' 006344'                          MES6
1172  007006' 177777                           177777
1173  007010' 006425'                  DLTERR: MES10
1174  007012' 177777                           177777
1175
1176  007014' 006447'                  NOT:    MES11
1177  007016' 177777                           177777
1178  007020' 006471'                  TOUT:   MES12
1179  007022' 177777                           177777
1180
1181  007024' 006317'                  DRP:    MES4
1182  007026' 007061'                          NUMB
1183  007030' 006331'                          MES5
1184  007032' 177777                           177777
1185  007034' 006317'                  DRP1:   MES4
1186  007036' 007061'                          NUMB
1187  007040' 006517'                          MES13
1188  007042' 177777                           177777
1189  007044' 006577'                  MES14:  XMES14
1190  007046' 177777                           177777
1191  007050' 006662'                  MES15:  XMES15
1192  007052' 177777                           177777
1193
1194  007054' 000005                   ADR1:   .BLKB   5
1195  007061'    000                   NUMB:   .BYTE   0
1196  007062'    000                           .BYTE   0
1197          007064'                          .EVEN
1198  007064'    000                   TRY1:   .BYTE   0
```

```
1199  007065'  000                TRY2:   .BYTE   0
1200  007066'  000                TRY3:   .BYTE   0
1201  007067'  000                TRY4:   .BYTE   0
1202  007070'  000                TRY5:   .BYTE   0
1203  007071'  000                TRY6:   .BYTE   0
1204  007072'  000                TRY7:   .BYTE   0
1205  007073'  000                TRY8:   .BYTE   0
1206  007074'  000                TRY9:   .BYTE   0
1207  007075'  000                TRY10:  .BYTE   0
1208  007076'  000                TRY11:  .BYTE   0
1209  007077'  000                TRY12:  .BYTE   0               ;NEED DO NOT DELETE FROM CODE
1210  007100'  000                FLAG:   .BYTE   0
1211  007101'  000                TRK:    .BYTE   0
1212  007102'  000                FIX:    .BYTE   0
1213          007104'                    .EVEN
1214                             ;       ------------------------
1215
1216          000001                     .END
```



```
ABW     002664R     539     542#
ACSR    000102R     219#   1010*
ADDR    000006R     185#   1033    1075
ADDR22= 001000      248#    689    1073
ADR1    001054R     739    1194#
ASB     000106R     223#   1002*
ASTAT   000104R     221#
AWAS    000110R     224#   1004*
BADFL   002212R     461#    505     507
BADFL2  002270R     470#    515
BADLOP  005012R     877#    899
BADSEC  000500R     326#    576     578*    744*    802*
BADSPT  000254R     248     249#    456     480     877     882     886     889
BEGIN   000000R     182#    431     471     499     516     524     529     565     569     573     585     592     599
                    605     674     675     679     680     696     704     708     739     760     774     780     783
                    786     786     943     944     947     955     959    1022    1023    1029
BIT0  = 000001      248#    602     720     965
BIT1  = 000002      248#    718     813     818     824     847
BIT10 = 002000      248#    752
BIT11 = 004000      248#    941     988
BIT12 = 010000      248#    405     990
BIT13 = 020000      248#    762
BIT14 = 040000      248#    683     766     768     992
BIT15 = 100000      248#    778    1077
BIT2  = 000004      248#    556     558     725     758
BIT3  = 000010      248#    445     502     538     924     930
BIT4  = 000020      248#
BIT5  = 000040      248#    407    1017
BIT6  = 000100      248#    984
BIT7  = 000200      248#    677     791    1020
BIT8  = 000400      248#    764     971     986
BIT9  = 001000      248#    748
BLKSAV  000562R     366#    527*
BLK1    000460R     318#    440*    527     559*    716     912*    815     817*    821     823*    825
BLOCK   004440R     652     812#    897
BREAK$= 104407      248#    674     675     679     680     943     944    1022    1023
BR1     000012R     197#   1085
BR2     000013R     198#
BTOD$ = 104421      248#
BUFIN   000600R     230     373#    479
CDATA$= 104412      248#    565
CDERCT  000144R     238#
CDWDCT  000146R     239#
CLEAR   003524R     670#    799     801     949
CLK     000456R     317#    939*    945*   1019*   1024*
CNT     000470R     322#    398     403*    520
CNT1    000472R     323#    518     568*
CONFIG  000056R     207#
CONWRT  004620R     827     841#
CSRA    000100R     217#   1009*
CT      002160R     449     451#
CYCLE   003026R     567#
CYL     000462R     319#    461*    627     628*    642     643*    660     661*    829*    834*    836*    841*    849*
                    852*    855     857*    877     903
CYLSAV  000464R     320#    628     643     661     830*    903*
DAOCL   002756R     558#
```

```
DAOST   002070R          435     438#
DATCK$= 104411           248#
DATER$= 104404           248#
DLTCNT  000454R          316#    404*    757*
DLTERR  007010R          760    1173#
DRIVE   000560R          365#    438*    927*    932*
DROP    004056R          604     729#    958
DRP     007024R          605     959    1181#
DRP1    007034R         1185#
DSKADR  000554R          363#    462*    463*    466*    512*    513*    514     625     640     658     831*    838*    839*
                         873*    874*
DVICE   000556R          364#    418*    428     430*    434     438     452     535     735*    932     1026*
DVIDI   000014R          1898    418
EA22    000512R          335#    698     699*    700*    701
ENDIT$= 104413           248#    569
END$  = 104410           248#    573
ERRORS  004114R          711     744#
ERRTYP  000106R          222#    772*    794*    953*   1027*
ERSUB1  005614R          754     952    1009#
ERSUB2  005570R         1002#
EXCED1  006766R          585    1164#
EXCED2  006774R          592    1167#
EXCED3  007002R          599    1170#
EXIT$ = 104400           248#    704
FDUNIT  005222R          444     501     532     923#    928
FERADR  000502R          327#    397
FILE    000252R          248#
FINT    003042R          436     447     455     500     517     537     572#    960
FIX     007102R         1212#
FLAG    007100R          417*    445     502     538     556     558*    718     720*    725*    813     818*    824*    847
                         924*    930*   1210#
FREE    000150R          240#
FUNC    000474R          324#    622*    637*    655*    702*    703
GETPA$= 104415           248#    471     524
GOA     002776R          557     561#    590
GOB     003006R          563#    597
GOC     002736R          554#    583
GOGO    003630R          636     651     669     687#
GWBUF$= 104414           248#    529
HRDCNT  000044R          202#
HRDERS= 104405           248#    774     955    1029
HRDPAS  000050R          204#
ICONT   000036R          199#
ICOUNT  000040R          200#
IDNUM   000122R          229#
IMODX.= 000000           241#    530
INIT    000030R          196#
INTR    000120R          228#    402*
LDRIVE  006746R          431    1156#
MAP22$= 104416           248#    696
MBLKRV  000514R          336#    411     846
MBLKTR  000516R          337#    844
MCPERR  006756R          783    1160#
MDPERR  006762R          786    1162#
MESLDP  006545R         1132#   1156
MES1    006202R         1087#   1158
```



```
MES10   006425R         1116#   1173
MES11   006447R         1120#   1176
MES12   006471R         1124#   1178
MES13   006517R         1128#   1187
MES14   007044R          499    1189#
MES15   007050R          516    1191#
MES2    006225R         1091#   1160
MES3    006257R         1096#   1162
MES4    006317R         1102#   1181    1185
MES5    006331R         1104#   1183
MES6    006344R         1106#   1165    1168    1171
MES7    006365R         1109#   1164
MES8    006376R         1111#   1167
MES9    006415R         1114#   1170
MHICY   000524R          340#    721     821
MIXDV   000574R          371#    473     826     904*    919*
MLOWCV  000522R          339#    815
MODF    000520R          338#    870
MODNAM  000000R          183#
MODSP   000252R          197     246#
MOT     000576R          372#    405*    409*    464     684
MSGN$ = 104403           248#    431     499     516     585     592     599     605     760     780     783     786     947
                         959
MSGS$ = 104402           248#
MSG$  = 104401           248#
NEXT    002620R          532#    571     575     606
NEXTA   003206R          579     586     593     600     602#
NOT     007014R          947    1176#
NOTRDY  005300R          450     506     545     939#
NTRUPT  003764R          688     706#
NULL  = 000000           248#
NUMB    007061R         1182    1186    1195#
NXT1    003046R          575#
ONCE    000476R          325#    416*
OPEN  = 000000           184     190     191     192     193     210     211     212     213     214     215     216     217
                         219     221     223     224     226     227     228     231     232     234     235     237     238
                         239     240     248#
OTOA$ = 104420           248#    739
PASCNT  000034R          198#
PA18    000504R          332#    691*    696
PA22    000510R          334#    697
PIRQ$ = 000004           248#    708
POPSP = 005726           248#
POPSP2= 022626           248#
PRTY  = 000000           248#
PRTV0 = 000000           188     248#
PRTV1 = 000040           248#
PRTV2 = 000100           248#
PRTV3 = 000140           248#
PRTV4 = 000200           248#
PRTV5 = 000240           187     248#
PRTV6 = 000300           248#
PRTV7 = 000340           248#
PS    = 177776           248#
PSW   = 177776           248#
PUSH  = 005746           248#
```

```
PUSH2 = 024646          248#
RAND$ = 104417          249#
RANNUM  000054R         206#
RBUFEA  000130R         232#    663
RBUFPA  000126R         231#    565     657
RBUFSZ  000132R         233#    475     525
RBUFVA  000124R         230#    471     524
READ    003432R         477     563     655#
READY   005502R         448     504     543     950     990#
RESTRT  002534R         225     516#
RESYNC  004434R         747     751     753     804#
RES1    000056R         208#    689    1073
RES2    000060R         209#
RETRX   002502R         478     512#
RETRY1  003052R         555     576#
RETRY2  003122R         562     587#
RETRY3  003154R         564     594#
REZET   005632R         443    1017#
RMAS    001616R         382#    682*    770     787     963    1018*   1048*
RMBA    001604R         377#    624*    639*    657*    691     697*   1038*
RMBAE   001650R         395#    698*   1080*
RMCSI   001600R         375#    671*    676*    677     683*    703*    745     762     766     940     941     957*    1009
                       1010    1020    1034*
RMCS2   001610R         379#    467*    533*    670*    697*    755     764     791     828*    964     976*    980*    1017*
                       1042*
RMCS3   001652R         396#   1082*
RMDA    001606R         378#    469*    625*    640*    658*    838     839    1040*
RMDB    001622R         394#    790     793*    798*   1052*
RMDC    001634R         389#    469*    627*    642*    660*    832     834    1062*
RMDS    001612R         380#    752     768     981    1044*
RMDT    001626R         386#    905     907     909     911    1056*
RMEC1   001644R         393#   1070*
RMEC2   001646R         394#   1072*
RMER1   001614R         381#    748    1046*
RMER2   001642R         392#    778    1068*
RMHR    001636R         390#   1064*
RMLA    001620R         383#   1050*
RMMR1   001624R         385#   1054*
RMMR2   001640R         391#   1066*
RMOF    001632R         388#    684*   1060*
RMSN    001630R         387#   1058*
RMWC    001602R         376#    623*    638*    656*   1036*
RMO3CK  005140R         472     542     626     641     659     903#
ROOM    004006R         716#    750
RSTRT   000112R         225#
RSTRT1  002554R         510     519     521     523#
SBADR   000102R         218#   1003*
SEC     000466R         321#    842*    865*    870     872*    873     886     888*    889     893*
SETUP   005724R         442    1033#
SOFCNT  000042R         201#
SOFERS= 104406          248#    796
SOFPAS  000046R         203#
SPOINT  000032R         197#    439
SPSIZ = 000040            1#    241
SR1     000016R         190#    407     602     758    1077
SR2     000020R         191#
```



```
SR3     000022R         192#
SR4     000024R         193#
START   001662R         196     400#    522
STAT    000026R         195#
STRT    002574R         527#    541     560    1084
STRT1   002602R         528#
SVR0    000062R         210#
SVR1    000064R         211#
SVR2    000066R         212#
SVR3    000070R         213#
SVR4    000072R         214#
SVR5    000074R         215#
SVR6    000076R         216#
SYSCNT  000052R         205#
TABLE   001600R         374#    774     796     955    1029
TBUF    000564R         367#
TOUT    007020R        1178#
TRERR   006752R         790    1158#
TRK     007101R         843*    860*    868     874     882    1211#
TRPDFD= 000022          248#
TRY1    007064R         546*    580*    581    1198#
TRY10   007075R        1207#
TRY11   007076R         551*   1208#
TRY12   007077R        1209#
TRY2    007065R         587*    588    1199#
TRY3    007066R         547*    594*    595    1200#
TRY4    007067R        1201#
TRY5    007070R         549*   1202#
TRY6    007071R        1203#
TRY7    007072R         549*   1204#
TRY8    007073R        1205#
TRY9    007074R         550*   1206#
T16     000526R         343#    406
T18     000540R         352#    410
UNITNO  000572R         370#    441*    467     508*    533     540*    670     697     730     739     828     923*    925
                        931*    980
UPDATT  003424R         552     652#
VECTOR  000010R         186#    688*   1083
WASADR  000104R         220#   1005*
WBUFEA  000136R         235#    630     645
WBUFPA  000134R         234#    624     639
WBUFRQ  000140R         236#
WBUFSZ  000142R         237#    530     717
WCNT1   000566R         368#    530*    531*    623     638
WCNT2   000570R         369#    475*    476*    525*    526*    656
WDFR    000116R         227#    400*
WDTO    000114R         226#    401*
WHO     005430R         789     963#
WRITCK  003330R         561     637#
WRITE   003234R         554     622#
XFERAD  001654R         397#
XFERCT  001656R         398#
XFLAG   000005R         184#
XMEM    000506R         333#    635*    650*    668*    692*    693*    694*    695*    701*    702
XMES14  006577R        1137#   1189
XMES15  006662R        1146#   1191
```

```
ZERO    000552R          362#    793
.     = 007104R          373#    446     465     566     577     896    1155#   1194#   1197#   1213#

. ABS.  000000     000
        007104     001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XRMAD0,XRMAD0/SOL/CRF:SYM=DDXCOM,XRMAD0
 RUN-TIME:  2  3  .5 SECONDS
 RUN-TIME RATIO: 20/6=4.5
 CORE USED:  7K  (13 PAGES)
```