```
.REM    %
```

IDENTIFICATION

PRODUCT CODE: AC-S077A-MC

PRODUCT NAME: CXRMCA0 RM05/3/2 MODULE

DATE CREATED: APRIL 1980

MAINTAINER: CX DIAGNOSTIC GROUP

AUTHOR: MIKE LEAVITT

1. ABSTRACT

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

2. REQUIREMENTS

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

TEST MEDIA:

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

LOAD DEVICE:

ANY DEVICE THAT IS SUPPORTED BY THE XXDP SOFTWARE PACKAGE.

3. PASS DEFINITION

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

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

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

4. EXECUTION TIME

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

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

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

11. BAD SECTOR TABLE

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

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

%

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



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

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

000416                          RBUF:   .BLKW   256.

                                ;RH/RM SAVED REGISTER POINTERS

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

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



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

001550  000406'                 XFERAD: FERADR
001552  177777                          177777

                                ;REGISTER ADDRESS INDEX VALUE

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

                                ;DISK ADDRESS EQUATES FOR LIMITS TABLES

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

                                ;TABLE OF ADDRESS LIMITS FOR RM05 AND RM03/02

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

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

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



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

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

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



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

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

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



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

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



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

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



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

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



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

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



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

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



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

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



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

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



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

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

. ABS.  000000      000
        007214      001
ERRORS DETECTED:  0

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



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

SEQ 0032

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

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

SEQ 0033

SEQ 0034

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

SEQ 0035

SEQ 0036

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

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

SEQ 0037

SEQ 0038

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


SPOINT      3-5#        4-7
```

SEQ 0039

SEQ 0040

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

ZERO    3-32#
```

SEQ 0041

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