```
1
                .REM    %
```

IDENTIFICATION

PRODUCT CODE: AC-F087C-MC

PRODUCT NAME: CXRPDC0 RP04,5,6 MOD

PRODUCT DATE: JUNE 1979

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

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

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

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

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

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

2. REQUIREMENTS

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

STORAGE:: RPD REQUIRES:

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

3. PASS DEFINITION

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

4. EXECUTION TIME

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

5. CONFIGURATION REQUIREMENTS

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

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

6. DEVICE/OPTION SETUP

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

7. OPERATOR OPTIONS

SR1

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

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

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

BIT 6
CLEAR(0):
DO NOT TYPE OUT BADSPOTS.
SET(1):

ALWAYS TYPE OUT ANY BADSPOT NOT ALREADY RECORDED ON BADSPOT TABLE.

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

8. NON-STANDARD PRINTOUTS

A. MOST PRINTOUTS HAVE THE STANDARD FORMATS.

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

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

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

$

```
                .SBTTL  MODULE HEADER BLOCK                                     ;DRB001
000000'                 IOMODX  <RPDC  >,176700,254,5,0,0,1500,104,RBUF,256.,,256.
000000'                 MODULE  150000,RPDC  ,176700,254,5,0,0,1500,104,RBUF,256.,,256.
                        .TITLE  RPDC DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6      23-MAY-78
                                .LIST   BIN
                        ;*********************************************************************
000000'                 BEGIN:
000000' 050122  041504      040 MODNAM: .ASCII  /RPDC / ;MODULE NAME.
000005'    000                  XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 176700                  ADDR:   176700+0                ;1ST DEVICE ADDR.
000010' 000254                  VECTOR: 254+0                   ;1ST DEVICE VECTOR.
000012'    240                  BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013'    000                  BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014' 000001                  DVID1:  0+1                     ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN                    ;SWITCH REGISTER 4
                                ;*********************************************************************
000026' 150000                  STAT:   150000                  ;STATUS WORD.
000030' 002044'                 INIT:   START                   ;MODULE START ADDR.
000032' 000252'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                       ;PASS COUNTER.
000036' 001500                  ICONT:  1500                    ;# OF ITERATIONS PER PASS=1500
000040' 000000                  ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042' 000000                  SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                  HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                  SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                  HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                  SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054' 000000                  RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:                         ;RESERVED FOR MONITOR USE
000056' 000000                  RES1:   0                       ;RESERVED FOR MONITOR USE
000060' 000000                  RES2:   0                       ;RESERVED FOR MONITOR USE
000062' 000000                  SVR0:   OPEN                    ;LOC TO SAVE R0.
000064' 000000                  SVR1:   OPEN                    ;LOC TO SAVE R1.
000066' 000000                  SVR2:   OPEN                    ;LOC TO SAVE R2.
000070' 000000                  SVR3:   OPEN                    ;LOC TO SAVE R3.
000072' 000000                  SVR4:   OPEN                    ;LOC TO SAVE R4.
000074' 000000                  SVR5:   OPEN                    ;LOC TO SAVE R5.
000076' 000000                  SVR6:   OPEN                    ;LOC TO SAVE R6.
000100' 000000                  CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102'                         SBADR:                          ;ADDR OF GOOD DATA, OR
000102' 000000                  ACSR:   OPEN                    ;CONTENTS OF CSR.
000104'                         WASADR:                         ;ADDR OF BAD DATA, OR
000104' 000000                  ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106'                         ERRTYP:                         ;TYPE OF ERROR
000106' 000000                  ASB:    OPEN                    ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN                    ;ACTUAL DATA.
000112' 002170'                 RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122' 000104                  IDNUM:  104                     ;MODULE IDENTIFICATION NUMBER=104
000124' 000716'                 RBUFVA: RBUF                    ;READ BUFFER VIRTUAL ADDRESS
```



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

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



```
000420' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000422' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000424' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000426' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000430' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000432' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000434' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000436' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000440' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000442' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000444' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000446' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000450' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000452' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000454' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000456' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000460' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000462' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000464' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000466' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000470' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000472' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000474' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000476' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000500' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000502' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000504' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000506' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000510' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000512' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000514' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000516' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000520' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000522' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000524' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000526' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000530' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000532' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000534' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000536' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000540' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000542' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000544' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000546' 177777                                          177777  ;CYLINDER ADDRESS                       ;DRB001
000550' 177777                                          177777  ;TRACK/SECTOR ADDRESS                   ;DRB001
000552'                                 ENDBBK:                                                         ;DRB001
                                                                                                        ;DRB001
                                        DLATE   = BIT2                                                  ;DRB001
                                        BPORT   = BIT4                                                  ;DRB001
                                        NORAND  = BIT5                                                  ;DRB001
                                        BDSPT   =BIT6                                                   ;DRB001
                                        RH70    = BIT7                                                  ;DRB001
                                                                                        ;**-16          ;DRB001
000552' 000000  000000  000000          S1:     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
000560' 000000  000000  000000
000566' 000000  000000  000000
```

```
000574' 000000  000000  000000
000602' 000000  000000  000000
000610' 000000  000000  000000
000616' 000000  000000  000000
000624' 000000

                        ;DO NOT CHANGE THE ORDER OF THE NEXT 4 LOCATIONS
                        ;NEEDED FOR MAP22 ROUTINE
000626' 000000          PA18:   0
000630' 000000          XMEM:   0
000632' 000000          PA22:   0
000634' 000000          EA22:   0
                                                                ;**-5           ;DRB001
000636' 000252'         NXTBBK: BADSPT          ; NEXT OPEN BADSPOT IN TABLE    ;DRB001
                                                                                ;DRB001
000640' 000000          CYLLIM: 0               ; UPPER DISK LIMIT (DEPENDS ON DISK TYPE)       ;DRB001
000642' 000000          CYLNDR: 0               ;ABSOLUTE CYLINDER ADDRESS OF CURRENT OP.;DRB001
000644' 000000          CYL:    0               ;RELATIVE DISK ADDRESS          ;DRB001
000646'                 DSKADR:                 ;FOR REFERENCING TRACK/SECTOR AS WORD           ;DRB001
000646'    000          SECTOR: .BYTE   0       ;SECTOR ADDRESS OF CURRENT OPERATION            ;DRB001
000647'    000          TRACK:  .BYTE   0       ;TRACK ADDRESS OF CURRENT OPERATION             ;DRB001
                                                                                ;DRB001
000650' 000000          BADCYL: 0               ;FROM DRIVE LOCATION REGISTERS, ON FINDING BADSPOT;DRB001
000652' 000000          BADSEC: 0               ;FROM RPDA ON BADSPOT OCCURANCE ;DRB001
000654' 000000          BADTRK: 0               ;FROM RPDA+1 ON BADSPOT         ;DRB001
                                                                                ;DRB001
000656' 000000          LASCYL: 0               ;LAST CYLINDER WHICH CAN BE WRITTEN             ;DRB001
000660' 000000          LASTRK: 0       ;LAST TRACK WHICH CAN BE WRITTEN        ;DRB001
000662' 000000          LASSEC: 0       ;LAST SECTOR WHICH CAN BE WRITTEN       ;DRB001
                                                                                ;DRB001
000664' 000000          SIZTRK: 0       ;NUMBER OF TRACKS IN WBUFSZ             ;DRB001
000666' 000000          SIZSEC: 0       ;REMAINDER OF WBUFSZ, IN SECTORS        ;DRB001
000670' 000000          FLAG:   0
000672' 000000          DVICE:  0
000674' 000000          DRIVE:  0
000676' 000000          UNITNO: 0
                                                                ;**-1           ;DRB001
000700' 000000          DLTCNT: 0
000702' 000000          FUNC:   0
000704' 000000          TIMER:  0
000706' 000000          ZERO:   0
000710' 000000          FERADR: 0
000712' 000000          CLK:    0
000714' 000000          TRY:    0
000716' 000400          RBUF:   .BLKW   256.

001716'                 TABLE:
001716' 000552'                 S
001720' 000554'                 S+2
001722' 000556'                 S+4
001724' 000560'                 S+6
001726' 000562'                 S+10
001730' 000564'                 S+12
001732' 000566'                 S+14
001734' 000570'                 S+16
001736' 000572'                 S+20
001740' 000574'                 S+22
```



```
001742' 000576'                 S+24
001744' 000600'                 S+26
001746' 000602'                 S+30
001750' 000604'                 S+32
001752' 000606'                 S+34
001754' 000610'                 S+36
001756' 000612'                 S+40
001760' 000614'                 S+42
001762' 000616'                 S+44
001764' 000620'                 S+46
                                                        ;**-1           ;DRB001
001766' 177777                  177777
001770' 000000          RHCS1:  0
001772' 000000          RHWC:   0
001774' 000000          RHBA:   0
001776' 000000          RPDA:   0
002000' 000000          RHCS2:  0
002002' 000000          RPDS:   0
002004' 000000          RPER1:  0
002006' 000000          RPAS:   0
002010' 000000          RPLA:   0
002012' 000000          RHDB:   0
002014' 000000          RPMR:   0
002016' 000000          RPDT:   0
002020' 000000          RPSN:   0
002022' 000000          RPOF:   0
002024' 000000          RPDC:   0
002026' 000000          RPCC:   0
002030' 000000          RPER2:  0
002032' 000000          RPER3:  0
002034' 000000          RPEC1:  0
002036' 000000          RPEC2:  0
002040' 000000          RHBAE:  0
002042' 000000          RHCS3:  0
                                                        ;**-4           ;DRB001
```

```
                                        .SBTTL  MODULE CODE                                                     ;DRB001
002044' 012767  000400  176042  START:  MOV     #256.,WDTO      ;256 WORDS TO MEM/ITERATION
002052' 012767  000400  176036          MOV     #256.,WDFR      ;256 WORDS FROM MEM/ITERATION
002060' 012767  000004  176032          MOV     #4,INTR         ;4 INTERRUPTS/ITERATION
002066' 016767  175722  176576          MOV     DVID1,DVICE     ;GET DRIVES TO TEST
002074' 123727  000041  000011          CMPB    @#41,#11        ;IF RP IS LOAD MEDIUM THEN
002102' 001021                          BNE     3$              ;BEGIN
002104' 113700  000040                  MOVB    @#40,R0         ;   GET LOAD-DEVICE NUMBER
002110' 012701  000001                  MOV     #1,R1           ;   INITIALIZE DEVICE MASK
002114' 105700                  1$:     TSTB    R0              ;   WHILE NOT POINTING AT LOAD-DEVICE DO
002116' 001403                          BEQ     2$              ;   BEGIN
002120' 006301                          ASL     R1              ;       SHIFT MASK TO NEXT DEVICE
002122' 105300                          DECB    R0              ;       KEEP TRACK OF SHIFTING
002124' 000773                          BR      1$              ;   END
002126' 130167  176540          2$:     BITB    R1,DVICE        ;   IF LOAD DEVICE IS SELECTED THEN
002132' 001405                          BEQ     3$              ;   BEGIN
002134' 113767  000040  176534          MOVB    @#40,UNITNO     ;       MOVE LOAD-DEVICE NUMBER TO DRIVE
002142' 004767  002252                  JSR     PC,DROP         ;       DROP THE LOAD-DEVICE
                                                                ;   END
002146'                         3$:                             ;END
002146' 005067  176472                  CLR     CYL             ;START AT CYLINDER 0                            ;DRB001
002152' 012767  000377  176466          MOV     #377,DSKADR     ;START AT TRACK 0, SECTOR 0                     ;DRB001
                                                                ;(* INCREMENT BEFORE TEST *)                    ;DRB001
002160' 004767  003652                  JSR     PC,SETUP
002164' 004767  003530                  JSR     PC,REZET        ;CLEAR THE RH
002170'                         RESTRT:
                                        GETPA$,BEGIN, RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
002176' 012767  177777  176472  LOOP1:  MOV     #-1,UNITNO      ;PRE-SET UNIT NUMBER
002204' 104414  000000'                 GWBUF$, BEGIN           ;GET WRITE BUFFER INFORMATION
                                                                        ;**-1                                   ;DRB001
002210'                         LOOP2:
002210' 004767  000744          3$:     JSR     PC,PICKDR       ;GO PICK A DRIVE
002214' 103407                          BCS     1$              ;RETURNS HERE IF ALL DRIVES DONE                ;DRB001
002216' 004767  001574                  JSR     PC,WRTLIM       ;GET DISK LIMITS FOR THIS TYPE OF DISK          ;DRB001
002222' 005067  176466                  CLR     TRY             ;ELSE CLR RE TRY COUNT
002226' 004767  000022                  JSR     PC,CYCLE        ;GO DO A CYCLE ON THIS DRIVE
002232' 000766                          BR      LOOP2           ;DO IT TO NEXT DRIVE

002234' 005767  176432          1$:     TST     DVICE           ;ANYBODY LEFT TO CHECK?
002240' 001002                          BNE     2$              ;BR IF YES
002242' 104410  000000'                 END$,BEGIN              ;
002246'                         2$:
002246' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION,
                                                                ;MONITOR SHALL TEST END OF PASS
002252' 000751                          BR      LOOP1           ;BR BACK IF NO

002254'                         CYCLE:
                                                                        ;**-4                                   ;DRB001
002254' 004767  000772                  JSR     PC,PICKBK       ;SELECT A SECTOR TO TEST                        ;DRB001
002260' 004567  000026                  JSR     R5,WRITE        ;GO WRITE A BLOCK
002264' 004567  000116                  JSR     R5,WRITCK       ;GO DO WRITE CHECK
002270' 004567  000206                  JSR     R5,READ         ;GO READ A BLOCK
002274' 104412  000000' 000126'         CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
002302' 002304'                         .+2                     ; IF ERROR, CONTINUE
```



```
002304' 004767  000266          JSR     PC,RELESE       ;RELEASE THE DRIVE                              ;DRB001
002310' 000207                  RTS     PC              ;END CYCLE                                      ;DRB001
```

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

002312' 012767  000161  176362  WRITE:  MOV     #161,FUNC       ; LOAD WRITE FUNCTION
002320' 012767  002312' 176362          MOV     #WRITE,FERADR   ;SAVE WHERE WE WERE
002326' 016746  175610                  MOV     WBUFSZ,-(SP)    ;GET WRITE SIZE
002332' 005416                          NEG     (SP)            ;NEGATE IT
002334' 012677  177432                  MOV     (SP)+,@RHWC     ; LOAD WORD COUNT
002340' 016777  175570  177426          MOV     WBUFPA,@RHBA    ; LOAD BUFFER ADDRESS
002346' 016777  176274  177422          MOV     DSKADR,@RPDA    ; LOAD DISK ADDRESS
002354' 016777  176262  177442          MOV     CYLNDR,@RPDC    ; LOAD CYLINDER ADDRESS         ;DRB001
                                ;       LINEUP  WBUFEA          ; LINE UP EA BITS FOR RHCS1
002402' 000167  000302                  JMP     GO              ; CONTINUE
002406' 012767  000151  176266  WRITCK: MOV     #151,FUNC       ; LOAD WRITE-CHECK FUNCTION
002414' 012767  002406' 176266          MOV     #WRITCK,FERADR  ;SAVE WHERE WE WERE
002422' 016746  175514                  MOV     WBUFSZ,-(SP)    ;GET WRITE SIZE
002426' 005416                          NEG     (SP)            ;NEGATE IT
002430' 012677  177336                  MOV     (SP)+,@RHWC     ; LOAD WORD COUNT
002434' 016777  175474  177332          MOV     WBUFPA,@RHBA    ; LOAD BUFFER ADDRESS
002442' 016777  176200  177326          MOV     DSKADR,@RPDA    ; LOAD DISK ADDRESS
002450' 016777  176166  177346          MOV     CYLNDR,@RPDC    ; LOAD CYLINDER ADDRESS         ;DRB001
                                ;       LINEUP  WBUFEA          ; LINE UP EA BITS FOR RHCS1
002476' 000167  000206                  JMP     GO              ; CONTINUE
002502' 012767  000171  176172  READ:   MOV     #171,FUNC       ; LOAD READ FUNCTION
002510' 012767  002502' 176172          MOV     #READ,FERADR    ;SAVE WHERE WE WERE
002516' 016746  175410                  MOV     RBUFSZ,-(SP)    ;GET READ SIZE
002522' 005416                          NEG     (SP)            ;NEGATE IT
002524' 012677  177242                  MOV     (SP)+,@RHWC     ; LOAD WORD COUNT
002530' 016777  175372  177236          MOV     RBUFPA,@RHBA    ; LOAD BUFFER ADDRESS
002536' 016777  176104  177232          MOV     DSKADR,@RPDA    ; LOAD DISK ADDRESS
002544' 016777  176072  177252          MOV     CYLNDR,@RPDC    ; LOAD CYLINDER ADDRESS         ;DRB001
                                ;       LINEUP  RBUFEA          ; LINE UP EA BITS FOR RHCS1
002572' 000167  000112                  JMP     GO              ; CONTINUE
002576' 016777  176074  177174  RELESE: MOV     UNITNO,@RHCS2   ;SET UP UNIT TO RELEASE         ;DRB001
002604' 012777  000013  177156          MOV     #13,@RHCS1      ;EXECUTE RELEASE COMMAND        ;DRB001
002612' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,...
002616' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002622' 000207                          RTS     PC              ;IT'S RELEASED                  ;DRB001
                                                                        ;**-33          ;DRB001

002624' 016777  176046  177146  CLEAR:  MOV     UNITNO,@RHCS2   ; LOAD UNIT ADDRESS
002632' 012777  000011  177130          MOV     #11,@RHCS1      ; ISSUE A DRIVE CLEAR
002640' 000240                          NOP                     ;WAIT
002642' 000240                          NOP                     ;FOR DRIVE CLEAR TO FINISH
002644' 012777  000023  177116          MOV     #23,@RHCS1      ;ISSUE A PACK ACK
```



```
002652' 105777  177112          1$:     TSTB    @RHCS1
002656' 100401                          BMI     2$
002660' 000774                          BR      1$              ; NO, WAIT TILL DONE
002662' 017746  177120          2$:     MOV     @RPAS,-(SP)     ;CLEAR AS BIT
002666' 012677  177114                  MOV     (SP)+,@RPAS
002672' 012777  040000  177070          MOV     #BIT14,@RHCS1   ; CLEAR ANY CONTROLLER ERRORS
002700' 012777  010000  177114          MOV     #BIT12,@RPOF    ; SET BIT FOR 11 FORMAT
002706' 000205                          RTS     R5              ; RETURN

002710' 016777  175762  177062  GO:     MOV     UNITNO,@RHCS2   ; LOAD UNIT SELECT
002716' 032767  001000  175132          BIT     #ADDR22,RES1    ;22 BIT SUPPORT?                ;DRB001
002724' 001434                          BEQ     1$              ;NO                             ;DRB001
002726' 017767  177042  175672          MOV     @RHBA,PA18      ;GET 18 BIT ADDR
002734' 006267  175670                  ASR     XMEM            ;SHIFT EA BITS TO POSITION 4,5
002740' 006267  175664                  ASR     XMEM
002744' 006267  175660                  ASR     XMEM
002750' 006267  175654                  ASR     XMEM
002754' 104416  000000' 000626'         MAP22$, BEGIN,PA18              ; GET 22-BIT ADDR FROM 18-BIT ADDR
002762' 016777  175644  177004          MOV     PA22,@RHBA      ;LOAD BA REG
002770' 016777  175640  177042          MOV     EA22,@RHBAE     ;LOAD BAE REG
002776' 042767  000034  175630          BIC     #34,EA22        ;CLEAR UNWANTED BITS
003004' 000367  175624                  SWAB    EA22            ;LOAD INTO BITS 8,9
003010' 016767  175620  175612          MOV     EA22,XMEM       ;LOAD XMEM TO SET INTO FUNCTION CODE
003016' 056767  175606  175656  1$:     BIS     XMEM,FUNC       ; LOAD EXTENDED MEMORY BITS
003024' 016777  175652  176736          MOV     FUNC,@RHCS1     ; EXECUTE THE FUNCTION
003032' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.
003036' 010046                  NTRUPT: MOV     R0,-(SP)        ;SAVE R0
003040' 016700  175632                  MOV     UNITNO,R0
003044' 116000  003242'                 MOVB    BITTAB(R0),R0
003050' 017746  176732                  MOV     @RPAS,-(SP)
003054' 040016                          BIC     R0,(SP)
003056' 012677  176724                  MOV     (R6)+,@RPAS
003062' 012600                          MOV     (SP)+,R0
                                        ;-----------------------------------------------------------------
003064' 000004  000000' 003072'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-----------------------------------------------------------------

003072' 004567  001360          1$:     JSR     R5,ERRORS       ; GO CHECK FOR ERRORS
003076' 103401                          BCS     2$              ;ERRORS DETECTED                ;DRB001
003100' 000205                          RTS     R5              ;OTHERWISE,  RETURN OK

003102' 012605                  2$:     MOV     (SP)+,R5        ;RESTORE R5, SINCE WE WON'T RETURN      ;DRB001
003104' 005767  175560                  TST     FLAG            ;DON'T RETRY ON HARD ERROR              ;DRB001
003110' 100410                          BMI     4$                                              ;DRB001
003112' 005267  175576                  INC     TRY             ;COUNT AN ERROR
003116' 026727  175572  000003          CMP     TRY,#3          ;TOO MANY FOR THIS CYCLE?               ;DRB001
003124' 002002                          BGE     4$              ;BR IF SO
003126' 000177  175556                  JMP     @FERADR         ;RE-EXECUTE THE DRIVER ROUTINE          ;DRB001
003132' 005067  175532          4$:     CLR     FLAG            ;CLEAR HARD ERROR BIT                   ;DRB001
003136' 104403  000000' 006722'         MSGN$,BEGIN,EXCED       ;ASCII MESSAGE CALL WITH COMMON HEADER
                                                                        ;**-7           ;DRB001
003144' 004767  000102          3$:     JSR     PC,PICKBK       ;TRY A DIFFERENT BLOCK
003150' 005367  175522                  DEC     UNITNO          ;WANT TO RE-DO SAME DRIVE WERE ON
003154' 000167  177030                  JMP     LOOP2           ;GO DO IT
                                                                        ;**-2           ;DRB001
```

```
                                        ;       ------------------------

                                                                                ;**-17                  ;DRB001
003160'                                 PICKDR:
003160' 005267  175512                  1$:     INC     UNITNO          ;POINT TO NEXT DRIVE
003164' 026727  175506  000010                  CMP     UNITNO,#8.              ;DONE LOOKING?
003172' 001002                                  BNE     2$              ;BR IF NO, ELSE
003174' 000261                                  SEC                     ;SET CARRY FOR NO-MORE-DRIVES   ;DRB001
003176' 000420                                  BR      5$              ;EXIT ROUTINE                   ;DRB001
003200' 016700  175472                  2$:     MOV     UNITNO,R0       ;USE AS AN INDEX
003204' 136067  003242' 175460                  BITB    BITTAB(R0),DVICE        ;TEST THIS DEVICE?
003212' 001001                                  BNE     3$              ;BR IF YES, ELSE
003214' 000761                                  BR      1$
003216'                                 3$:
                                                                                ;**-1                   ;DRB001
003216' 004567  002322                          JSR     R5,READY        ;SEE IF DRIVE IS READY
003222' 103402                                  BCS     4$              ;BR IF IT WAS READY             ;DRB001
003224' 004767  002120                          JSR     PC,NOTRDY       ;ELSE GO CLEAR IT AND CHECK AGAIN
003230' 004767  000702                  4$:     JSR     PC,GETDVT       ;IS THIS LEGAL DRIVE TYPE?      ;DRB001
003234' 103751                                  BCS     1$              ;ILLEGAL TYPE: DROPPED,  TRY NEXT ;DRB001
003236' 000241                                  CLC                     ;RETURN, NEXT DRIVE AVAILABLE   ;DRB001
003240' 000207                          5$:     RTS     PC              ;GO HOME                        ;DRB001
003242' 001001                          BITTAB: 1001
003244' 004004                                  4004
003246' 020020                                  20020
003250' 100100                                  100100


003252'                                 PICKRK:                                 ;**-11                  ;DRB001
                                                ;(* FUNCTION: SELECT NEXT DISK ADDRESS TO TEST,       *)  ;DRB001
                                                ;(* AVOID WRITING OVER KNOWN BAD SPOTS OR DISK'S BAD  *)  ;DRB001
                                                ;(* SPOT FILE (IF OPERATOR TELLS US IT'S THERE).  CYL *)  ;DRB001
                                                ;(* IS "RELATIVE ADDRESS", MAX RANGE 0-410 (RP04/5) OR*)  ;DRB001
                                                ;(* 0-407 (RP06);  THIS IS MOVED TO CYLNDR AND INCRE- *)  ;DRB001
                                                ;(* MENTED TO SECOND HALF IF THIS COPY OF RPD IS IN   *)  ;DRB001
                                                ;(* PORT B MODE.                                     *)  ;DRB001
                                                                                                        ;DRB001
003252' 032767  000040  174536                  BIT     #NORAND,SR1     ;IF SEQUENTIAL SEEKS THEN       ;DRB001
003260' 001403                                  BEQ     1$              ;BEGIN                          ;DRB001
003262' 105267  175360                          INCB    SECTOR          ; SECTOR := SECTOR + 1          ;DRB001
003266' 000432                                  BR      2$              ;END                            ;DRB001
003270'                                 1$:                             ;ELSE (* RANDOM SEEKS *)        ;DRB001
                                                                        ;BEGIN (* FUDGE 3 UN-RELATED    ;DRB001
                                                                        ;  (* NUMBERS FROM 1 RAND CALL  *)  ;DRB001
003270' 104417  000000'                 RAND$,BEGIN
003274' 016700  174554                  MOV     RANNUM,R0       ; R0 := RANNUM                  ;DRB001
003300' 042700  176000                  BIC     #176000,R0      ; R0 := R0 MOD 1024.            ;DRB001
003304' 010067  175334                  MOV     R0,CYL          ; CYL := R0                     ;DRB001
003310' 016700  174540                  MOV     RANNUM,R0       ; "RECHARGE" R0                 ;DRB001
003314' 000300                          SWAB    R0              ; R0<0:8> := R0<8:8>            ;DRB001
003316' 042700  177740                  BIC     #177740,R0      ; R0 := R0 MOD 32.              ;DRB001
003322' 110067  175320                  MOVB    R0,SECTOR       ; SECTOR := R0                  ;DRB001
003326' 016700  174522                  MOV     RANNUM,R0       ; AND AGAIN                     ;DRB001
003332' 000241                          CLC                     ; CARRY := 0                    ;DRB001
003334' 006100                          ROL     R0              ; CARRY := R0<15:1>             ;DRB001
```



```
003336' 006100                          ROL     R0                                              ;DRB001
003340' 006100                          ROL     R0              ;                               ;DRB001
003342' 006100                          ROL     R0              ;                               ;DRB001
003344' 042700  177740                  BIC     #177740,R0      ; R0 := (R0<0:13>*8.) + R0<13:3>  ;DRB001
                                                                ; (* NOTE: THE FORMULAE USED ARE *)  ;DRB001
                                                                ; (* ABSOLUTELY IRRELEVANT       *)  ;DRB001
003350' 110067  175273                  MOVB    R0,TRACK        ; TRACK := R0                   ;DRB001
003354'                                 2$:                     ;END                            ;DRB001
003354' 126727  175266  000025          CMPB    SECTOR,#21.     ;IF SECTOR GT 21. THEN          ;DRB001
003362' 003412                          BLE     3$              ;BEGIN                          ;DRB001
003364' 105267  175257                  INCB    TRACK           ; TRACK := TRACK + 1            ;DRB001
003370' 016700  175252                  MOV     SECTOR,R0       ; R0 := SECTOR                  ;DRB001
003374' 012701  000026                  MOV     #22.,R1         ; R1 := # OF SECTORS IN TRACK   ;DRB001
003400' 004767  000762                  JSR     PC,MODLUS       ; R0 := R0 MOD R1               ;DRB001
003404' 110067  175236                  MOVB    R0,SECTOR       ; SECTOR := R0                  ;DRB001
003410'                                 3$:                     ;END                            ;DRB001
                                        ;(* NOW, CUT VALUES DOWN TO LEGAL RANGES FOR CURRENT DRIVE TYPE *) ;DRB001
003410' 126727  175233  000022          CMPB    TRACK,#18.      ;IF TRACK GT 18. THEN           ;DRB001
003416' 003412                          BLE     4$              ;BEGIN                          ;DRB001
003420' 005267  175220                  INC     CYL             ; CYL := CYL + 1                ;DRB001
003424' 116700  175217                  MOVB    TRACK,R0        ; R0 := TRACK                   ;DRB001
003430' 012701  000023                  MOV     #19.,R1         ; R1 := 19.                     ;DRB001
003434' 004767  000726                  JSR     PC,MODLUS       ; R0 := R0 MOD R1               ;DRB001
003440' 110067  175203                  MOVB    R0,TRACK        ; TRACK := R0                   ;DRB001
003444'                                 4$:                     ;END                            ;DRB001
003444' 026767  175174  175166          CMP     CYL,CYLLIM      ;IF CYL GT # OF CYLINDERS IN RANGE THEN ;DRB001
003452' 002421                          BLT     5$              ;BEGIN                          ;DRB001
003454' 032767  000040  174334          BIT     #NORAND,SR1     ; IF SEQUENTIAL SEEKS THEN      ;DRB001
003462' 001405                          BEQ     6$              ; BEGIN                         ;DRB001
003464' 005067  175154                  CLR     CYL             ;  RESET CYLINDER               ;DRB001
003470' 005067  175152                  CLR     DSKADR          ;  TRACK := 0, SECTOR := 0      ;DRB001
003474' 000410                          BR      7$              ; END                           ;DRB001
003476'                                 6$:                     ; ELSE                          ;DRB001
                                                                ; BEGIN                         ;DRB001
003476' 016700  175142                  MOV     CYL,R0          ;  R0 := CYL                    ;DRB001
003502' 016701  175132                  MOV     CYLLIM,R1       ;  R1 := CYLLIM                 ;DRB001
003506' 004767  000654                  JSR     PC,MODLUS       ;  R0 := R0 MOD R1              ;DRB001
003512' 010067  175126                  MOV     R0,CYL          ;  CYL := R0                    ;DRB001
003516'                                 7$:                     ; END                           ;DRB001
003516'                                 5$:                     ;END                            ;DRB001
003516' 016767  175122  175116          MOV     CYL,CYLNDR      ;CYLNDR := CYL                  ;DRB001
003524' 032767  000020  174264          BIT     #BPORT,SR1      ;IF BPORT AND DUAL PORT THEN
003532' 001407                          BEQ     8$              ;BEGIN                          ;DRB001
003534' 032767  000001  175126          BIT     #BIT0,FLAG      ;                               ;DRB001
003542' 001403                          BEQ     8$              ;                               ;DRB001
003544' 066767  175070  175070          ADD     CYLLIM,CYLNDR   ; INCREMENT TO SECOND HALF OF DISK  ;DRB001
003552'                                 8$:                     ;END                            ;DRB001
003552' 026767  175064  175076          CMP     CYLNDR,LASCYL   ;IF CYLNDR GT LASCYL THEN       ;DRB001
003560' 003431                          BLE     9$              ;BEGIN                          ;DRB001
003562' 016767  175070  175052          MOV     LASCYL,CYLNDR   ; CYLNDR := LASCYL (* TRUNCATE *)  ;DRB001
003570' 032767  000040  174220          BIT     #NORAND,SR1     ; IF SEQUENTIAL THEN            ;DRB001
003576' 001404                          BEQ     10$             ; BEGIN                         ;DRB001
003600' 005067  175040                  CLR     CYL             ;  RESET CYLINDER               ;DRB001
003604' 005067  175036                  CLR     DSKADR          ;  TRACK := 0, SECTOR := 0      ;DRB001
003610'                                 10$:                    ; END                           ;DRB001
003610' 126767  175033  175042          CMPB    TRACK,LASTRK    ; IF TRACK GT LASTRKK THEN      ;DRB001
```

```
  746  003616' 003412                  BLE     11$             ; BEGIN                                 ;DRB001
  747  003620' 116767  175034  175021  MOVB    LASTRK,TRACK    ;  TRACK := LASTRK                      ;DRB001
  748  003626' 126767  175014  175026  CMPB    SECTOR,LASSEC   ;  IF SECTOR GT LASSEC THEN             ;DRB001
  749  003634' 003403                  BLE     12$             ;  BEGIN                                ;DRB001
  750  003636' 116767  175020  175002  MOVB    LASSEC,SECTOR   ;   SECTOR := LASSEC                    ;DRB001
  751  003644'                     12$:                        ;  END                                  ;DRB001
  752  003644'                     11$:                        ; END                                   ;DRB001
  753  003644'                     9$:                         ;END                                    ;DRB001
  754                                                                                                  ;DRB001
  755                                                                                                  ;DRB001
  756  003644' 012700  000252'        MOV     #BADSPT,R0      ;R0 := ADDRESS OF TABLE                 ;DRB001
  757  003650' 016701  174762          MOV     NXTBBK,R1       ;R1 := END OF TABLE                     ;DRB001
  758  003654' 116702  174766          MOVB    SECTOR,R2       ;R2 := SECTOR                           ;DRB001
  759  003660' 116703  174763          MOVB    TRACK,R3        ;R3 := TRACK                            ;DRB001
  760  003664' 016704  174752          MOV     CYLNDR,R4       ;R4 := CYLNDR                           ;DRB001
  761                                  ;(* CALCULATE UPPER LIMIT OF THIS WRITE *)                      ;DRB001
  762  003670' 066702  174772          ADD     SIZSEC,R2       ;R2 := R2 + # OF SECTORS IN WRITE        ;DRB001
  763  003674' 020227  000025          CMP     R2,#21.         ;IF R2 GT 21. THEN                      ;DRB001
  764  003700' 003403                  BLE     13$             ;BEGIN                                  ;DRB001
  765  003702' 005203                  INC     R3              ; R3 := R3 + 1 (INCR. UPPER TRACK)      ;DRB001
  766  003704' 162702  000026          SUB     #22.,R2         ; TRUNCATE R2                           ;DRB001
  767  003710'                     13$:                        ;END                                    ;DRB001
  768  003710' 066703  174750          ADD     SIZTRK,R3       ;R3 := R3 + TRKSIZ                      ;DRB001
  769  003714' 020327  000022          CMP     R3,#18.         ;IF R3 GT 18. THEN                      ;DRB001
  770  003720' 003403                  BLE     14$             ;BEGIN                                  ;DRB001
  771  003722' 005204                  INC     R4              ; R4 := R4 + 1 (INCR. UPPER CYLINDER)   ;DRB001
  772  003724' 162703  000023          SUB     #19.,R3         ; TRUNCATE R3                           ;DRB001
  773  003730'                     14$:                        ;END                                    ;DRB001
  774  003730' 020001                  CMP     R0,R1           ;WHILE R0 LT R1 DO                      ;DRB001
  775  003732' 002030                  BGE     15$             ;BEGIN                                  ;DRB001
  776  003734' 026710  174702          CMP     CYLNDR,(R0)     ; IF CYLNDR LE (R0) AND R4 GE (R0)      ;DRB001
  777  003740' 003022                  BGT     17$             ;  AND TRACK LE 3(R0) AND R3 GE 3(R0)   ;DRB001
  778  003742' 020410                  CMP     R4,(R0)         ;  AND SECTOR LE 2(R0) AND R2 GE 2(R0) THEN;DRB001
  779  003744' 002420                  BLT     17$             ;                                       ;DRB001
  780  003746' 126760  174675  000003  CMPB    TRACK,3(R0)     ;                                       ;DRB001
  781  003754' 003014                  BGT     17$             ;                                       ;DRB001
  782  003756' 120360  000003          CMPB    R3,3(R0)        ;                                       ;DRB001
  783  003762' 002411                  BLT     17$             ;                                       ;DRB001
  784  003764' 126760  174656  000002  CMPB    SECTOR,2(R0)    ;                                       ;DRB001
  785  003772' 003005                  BGT     17$             ;                                       ;DRB001
  786  003774' 120260  000002          CMPB    R2,2(R0)        ;                                       ;DRB001
  787  004000' 002402                  BLT     17$             ; BEGIN                                 ;DRB001
  788  004002' 000167  177244          JMP     PICKBK          ;  TRY ANOTHER SECTOR                   ;DRB001
  789  004006'                     17$:                        ; END                                   ;DRB001
  790  004006' 062700  000004          ADD     #4,R0           ; INCREMENT TO NEXT BAD SPOT            ;DRB001
  791  004012' 000746                  BR      15$             ;END                                    ;DRB001
  792  004014'                     16$:                        ;                                       ;DRB001
  793  004014' 000207                  RTS     PC              ;CPU GO HOME!!!!                        ;DRB001
  794                                                                                                  ;DRB001
  795                                                                                                  ;DRB001
  796  004016'                     WRTLIM:                                                             ;DRB001
  797                                  ;(* WE ALLOW USER TO CHANGE WBUFSZ "AT WILL"--THEREFORE *)    ;DRB001
  798                                  ;(* WE DO NOT AUTOMATICALLY KNOW THE TRANSFER SIZE,     *)    ;DRB001
  799                                  ;(* COMPUTATION IS MADE EASIER, FURTHERMORE, IF WE HAVE *)    ;DRB001
  800                                  ;(* THE SIZE BROKEN INTO TRACKS AND SECTORS (SINCE      *)    ;DRB001
  801                                  ;(* WBUFSZ IS ONE WORD, THE TRANSFER CAN'T BE GREATER   *)    ;DRB001
```



```
  802                                  ;(* THAN 12 TRACKS).  THIS ROUTINE ALSO CALCULATES THE  *)    ;DRB001
  803                                  ;(* UPPER LIMITS ON DISK WRITES SO AS NOT TO OVERFLOW    *)    ;DRB001
  804                                  ;(* HIGH CYLINDER.                                       *)    ;DRB001
  805                                                                                                  ;DRB001
  806  004016' 016700  174120          MOV     WBUFSZ,R0       ;R0 := ALLOCATED WRITE BUFFER           ;DRB001
  807  004022' 000300                  SWAB    R0              ;R0 := R0 / 256.                        ;DRB001
  808  004024' 042700  177400          BIC     #177400,R0      ; (* R0 IS WBUFSZ IN SECTORS *)          ;DRB001
  809  004030' 005067  174630          CLR     SIZTRK          ;SIZTRK := 0                            ;DRB001
  810  004034' 020027  000026  18$:    CMP     R0,#22.         ;WHILE R0 GE 22. DO                     ;DRB001
  811  004040' 002405                  BLT     20$             ;BEGIN                                  ;DRB001
  812  004042' 105267  174616          INCB    SIZTRK          ; SIZTRK := SIZTRK + 1                  ;DRB001
  813  004046' 162700  000026          SUB     #22.,R0         ; DECREMENT SECTORS BY 1 TRACK          ;DRB001
  814  004052' 000770                  BR      18$             ;END                                    ;DRB001
  815  004054' 010067  174606  20$:    MOV     R0,SIZSEC       ;SIZSEC := REMAINDER                    ;DRB001
  816  004060' 012701  000022          MOV     #18.,R1         ;R1 := HIGH TRACK                       ;DRB001
  817  004064' 012702  000025          MOV     #21.,R2         ;R2 := HIGH SECTOR                      ;DRB001
  818  004070' 160002                  SUB     R0,R2           ;R2 := R2 - SIZSEC                      ;DRB001
  819  004072' 005702                  TST     R2              ;IF R2 LT 0 THEN                        ;DRB001
  820  004074' 002003                  BGE     30$             ;BEGIN                                  ;DRB001
  821  004076' 062702  000026          ADD     #22.,R2         ; INCREMENT BACK UP                     ;DRB001
  822  004102' 005301                  DEC     R1              ; CAN'T FIT THAT LAST TRACK...          ;DRB001
  823  004104'                     30$:                        ;END                                    ;DRB001
  824  004104' 166701  174554          SUB     SIZTRK,R1       ;R1 := R1 - SIZTRK                      ;DRB001
  825  004110' 005701                  TST     R1              ;IF R1 LT 0 THEN                        ;DRB001
  826  004112' 002004                  BGE     40$             ;BEGIN                                  ;DRB001
  827  004114' 062701  000023          ADD     #19.,R1         ; INCREMENT R1 BACK UP                  ;DRB001
  828  004120' 005367  174532          DEC     LASCYL          ; LASCYL := LASCYL - 1                  ;DRB001
  829  004124'                     40$:                        ;END                                    ;DRB001
  830  004124' 010167  174530          MOV     R1,LASTRK       ;LASTRK := HIGHEST TRACK WE CAN WRITE   ;DRB001
  831  004130' 010267  174526          MOV     R2,LASSEC       ;LASSEC := HIGHEST SECTOR WE CAN WRITE  ;DRB001
  832  004134' 000207                  RTS     PC              ;END OF ROUTINE                         ;DRB001
  833  004136'                     GETDVT:                                                             ;DRB001
  834                                  ;(* FIND WHICH TYPE OF DRIVE THIS IS.  IF RP04/5, SET CYLLIM *);DRB001
  835                                  ;(* TO 206.  IF RP06, SET CYLLIM TO 408.  DROP THE DRIVE IF  *);DRB001
  836                                  ;(* IT IS NOT A RP04/5/6                                     *);DRB001
  837                                                                                                  ;DRB001
  838  004136' 016777  174534  175634  MOV     UNITNO,@RHCS2   ;SELECT THE DRIVE                       ;DRB001
  839  004144' 027727  175646  024022  CMP     @RPDT,#24022    ;IF IT IS AN RP06 DUAL PORT THEN        ;DRB001
  840  004152' 001012                  BNE     1$              ;BEGIN                                  ;DRB001
  841  004154' 052767  000001  174506  BIS     #BIT0,FLAG      ; DUAL PORT                             ;DRB001
  842  004162' 012767  000630  174450  MOV     #408.,CYLLIM    ; CYLLIM := 408.                        ;DRB001
  843  004170' 012767  001456  174460  MOV     #814.,LASCYL    ; LASCYL := DISK HIGH LIM               ;DRB001
  844  004176' 000471                  BR      2$              ;END                                    ;DRB001
  845  004200'                     1$:                         ;ELSE                                   ;DRB001
  846  004200' 027727  175612  024021  CMP     @RPDT,#24021    ;IF IT IS RP04/5 DUAL PORT THEN         ;DRB001
  847  004206' 001404                  BEQ     3$              ;                                       ;DRB001
  848  004210' 027727  175602  024020  CMP     @RPDT,#24020    ;                                       ;DRB001
  849  004216' 001012                  BNE     4$              ;BEGIN                                  ;DRB001
  850  004220'                     3$:                         ;                                       ;DRB001
  851  004220' 052767  000001  174442  BIS     #BIT0,FLAG      ; DUAL PORT                             ;DRB001
  852  004226' 012767  000316  174404  MOV     #206.,CYLLIM    ; CYLLIM := 206.                        ;DRB001
  853  004234' 012767  000632  174414  MOV     #410.,LASCYL    ; LASCYL := DISK HIGH LIM               ;DRB001
  854  004242' 000447                  BR      5$              ;END                                    ;DRB001
  855  004244'                     4$:                         ;ELSE                                   ;DRB001
  856  004244' 027727  175546  020022  CMP     @RPDT,#20022    ;IF RP06 SINGLE PORT THEN               ;DRB001
  857  004252' 001012                  BNE     6$              ;BEGIN                                  ;DRB001
```

```
 858  004254' 042767  000001  174406          BIC     #BIT0,FLAG      ; SINGLE PORT                           ;DRB001
 859  004262' 012767  001457  174350          MOV     #815.,CYLLIM    ; CAN USE ENTIRE DISK                   ;DRB001
 860  004270' 012767  001456  174360          MOV     #814.,LASCYL    ; LASCYL := DISK HIGH LIMIT             ;DRB001
 861  004276' 000431                          BR      2$              ;END                                    ;DRB001
 862  004300'                                 6$:                     ;ELSE                                   ;DRB001
 863  004300' 027727  175512  020021          CMP     @RPDT,#20021    ;IF RP04/5 SINGLE PORT THEN              ;DRB001
 864  004306' 001404                          BEQ     8$              ;                                       ;DRB001
 865  004310' 027727  175502  020020          CMP     @RPDT,#20020    ;                                       ;DRB001
 866  004316' 001012                          BNE     9$              ;                                       ;DRB001
 867  004320'                                 8$:                     ;BEGIN                                  ;DRB001
 868  004320' 042767  000001  174342          BIC     #BIT0,FLAG      ; SINGLE PORT                           ;DRB001
 869  004326' 012767  000633  174304          MOV     #411.,CYLLIM    ; USE ENTIRE DISK                       ;DRB001
 870  004334' 012767  000632  174314          MOV     #410.,LASCYL    ; LASCYL := DISK HIGH LIMIT             ;DRB001
 871  004342' 000407                          BR      2$              ;END                                    ;DRB001
 872  004344'                                 9$:                     ;ELSE                                   ;DRB001
 873                                                                  ;BEGIN                                  ;DRB001
 874  004344' 004767  000050                  JSR     PC,DROP         ; DROP THE DRIVE                        ;DRB001
 875  004350' 104403  000000' 007012'         MSGN$,BEGIN,BADTYP      ;ASCII MESSAGE CALL WITH COMMON HEADER
 876  004356' 000261                          SEC                     ; ERROR RETURN                          ;DRB001
 877  004360' 000401                          BR      7$              ;                                       ;DRB001
 878  004362'                                 5$:                     ;END                                    ;DRB001
 879  004362'                                 2$:                     ;                                       ;DRB001
 880  004362' 000241                          CLC                     ;NO ERROR                               ;DRB001
 881  004364'                                 7$:                     ;                                       ;DRB001
 882  004364' 000207                          RTS     PC              ;RETURN                                 ;DRB001
 883  004366'                                 MODLUS:                                                         ;DRB001
 884                                          ;(* THIS LITTLE ROUTINE COMPUTES R0 MODULUS R1       *)          ;DRB001
 885                                          ;(*  I.E., THE REMAINDER OF R0/R1, AND RETURNS THE  *)          ;DRB001
 886                                          ;(* ANSWER IN R0.                                    *)          ;DRB001
 887  004366' 020001                          1$:     CMP     R0,R1           ;WHILE R0 GE R1 DO              ;DRB001
 888  004370' 002402                          BLT     2$              ;BEGIN                                  ;DRB001
 889  004372' 160100                          SUB     R1,R0           ; R0 := R0-R1                           ;DRB001
 890  004374' 000774                          BR      1$              ;END                                    ;DRB001
 891  004376'                                 2$:                     ;                                       ;DRB001
 892  004376' 000207                          RTS     PC              ;THAT'S ALL                             ;DRB001
 893
 894  004400' 012700  000716'         CLRRB:  MOV     #RBUF,R0        ;CLEAR RBUF BUFFER
 895  004404' 016701  173522                  MOV     RBUFSZ,R1       ;GET ITS ADDR AND SIZE
 896  004410' 005020                  CLRCOM: CLR     (R0)+           ;CLEAR ANOTHER
 897  004412' 005301                          DEC     R1              ;COUNT ANOTHER
 898  004414' 001375                          BNE     CLRCOM          ;BR BACK TILL DONE
 899  004416' 000207                          RTS     PC
 900
 901  004420' 012701  000001          DROP:   MOV     #1,R1           ; INITIALIZE DROP PICKER
 902  004424' 016700  174246                  MOV     UNITNO,R0       ; GET THE DRIVE NUMBER
 903  004430' 001403                          BEQ     2$              ; IF DRIVE 0 GO DROP IT
 904  004432' 006301                  1$:     ASL     R1              ; POINT TO NEXT DRIVE
 905  004434' 005300                          DEC     R0              ; IS THIS THE ONE ?
 906  004436' 001375                          BNE     1$              ; NO, LOOK AGAIN
 907  004440' 040167  174226          2$:     BIC     R1,DVICE        ; DROP THE DRIVE
 908                                          ;*************************************************************
 909                                                                  ;CONVERT UNITNO TO ASCII AND
 910                                                                  ;STORE AT ADR1
 911  004444' 104420  000000' 000676'         OTOA$,BEGIN,UNITNO,ADR1
 912  004452' 007024'
 913                                          ;*************************************************************
```



```
 914  004454' 000207                          RTS     PC              ; RETURN
 915                                  ;       ------------------------
 916
 917  004456'                                 ERRORS:                                                         ;DRB001
 918  004456' 005777  175306                  TST     @RHCS1          ;IF NO ERROR THEN                       ;DRB001
 919  004462' 100403                          BMI     1$              ;BEGIN                                  ;DRB001
 920  004464' 000241                          CLC                     ; SET NO ERROR                          ;DRB001
 921  004466' 000167  000654                  JMP     2$              ; RETURN NO ERROR                       ;DRB001
 922                                                                  ;END                                    ;DRB001
 923  004472'                                 1$:                     ;ELSE                                   ;DRB001
 924                                                                  ;BEGIN                                  ;DRB001
 925  004472' 005000                          CLR     R0              ; R0 := 0                               ;DRB001
 926  004474' 016701  175270                  MOV     RHCS1,R1        ; BEGIN OF REGISTER POINTERS            ;DRB001
 927  004500'                                 3$:                     ; REPEAT                                ;DRB001
 928                                                                  ; BEGIN                                 ;DRB001
 929  004500' 012160  000552'                 MOV     (R1)+,S(R0)     ;  MOV REGISTER TO TEMPORARY            ;DRB001
 930  004504' 005720                          TST     (R0)+           ;  ADVANCE POINTER                      ;DRB001
 931  004506' 020027  000046                  CMP     R0,#46          ; END                                   ;DRB001
 932  004512' 002772                          BLT     3$              ; UNTIL R0 GE 46                        ;DRB001
 933  004514' 004767  001162                  JSR     PC,ERSUB1       ; GET SOME TRIVIAL INFO                 ;DRB001
 934  004520' 005767  174036                  TST     S+10            ; IF DATA LATE ERROR THEN               ;DRB001
 935  004524' 100015                          BPL     4$              ; BEGIN                                 ;DRB001
 936  004526' 005267  174146                  INC     DLTCNT          ;  COUNT IT UNCONDITIONALLY             ;DRB001
 937  004532' 032767  000004  173256          BIT     #DLATE,SR1      ;  IF OPERATOR WANTS MESSAGE THEN       ;DRB001
 938  004540' 001005                          BNE     5$              ;  BEGIN                                ;DRB001
 939  004542' 104403  000000' 006726'         MSGN$,REGIN,DLTERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
 940  004550' 000167  000440                  JMP     6$              ;   HARD ERROR PROCESS                  ;DRB001
 941                                                                  ;  END                                   ;DRB001
 942  004554'                                 5$:                     ;  ELSE                                 ;DRB001
 943  004554' 000167  000502                  JMP     7$              ;   NO MESSAGE "HARD ERROR" RETURN      ;DRB001
 944                                                                  ; END                                   ;DRB001
 945  004560' 032767  000100  174000  4$:     BIT     #BIT6,S+14      ; IF ER1<6> OR ( ER1<4> AND NOT         ;DRB001
 946  004566' 001020                          BNE     8$              ;  ER1<15> AND NOT ER1<8> ) THEN        ;DRB001
 947  004570' 032767  000020  173770          BIT     #BIT4,S+14      ;                                       ;DRB001
 948  004576' 001471                          BEQ     9$              ;                                       ;DRB001
 949  004600' 032767  100000  173760          BIT     #BIT15,S+14     ;                                       ;DRB001
 950  004606' 001402                          BEQ     30$             ;                                       ;DRB001
 951  004610' 000167  000400                  JMP     6$              ; (* HARD ERROR *)                      ;DRB001
 952  004614' 032767  000400  173744  30$:    BIT     #BIT8,S+14      ;                                       ;DRB001
 953  004622' 001402                          BEQ     8$              ;                                       ;DRB001
 954  004624' 000167  000442                  JMP     10$             ; (* SOFT ERROR *)                      ;DRB001
 955  004630'                                 8$:                     ; BEGIN (* BADSPOT *)                   ;DRB001
 956  004630' 032767  000100  173160          BIT     #BDSPT,SR1      ;  IF OPERATOR WANTS TYPEOUT THEN       ;DRB001
 957  004636' 001434                          BEQ     12$             ;                                       ;DRB001
 958  004640'                                 11$:                    ;  BEGIN                                ;DRB001
 959  004640' 116767  173714  174004          MOVB    S+6,BADSEC      ;   SET CURRENT SECTOR                  ;DRB001
 960  004646' 116767  173707  174000          MOVB    S+7,BADTRK      ;   SET CURRENT TRACK                   ;DRB001
 961  004654' 016767  173730  173766          MOV     S+36,BADCYL     ;   SET CURRENT CYLINDER                ;DRB001
 962                                          ;*************************************************************
 963                                                                  ;CONVERT UNITNO TO ASCII AND
 964                                                                  ;STORE AT ADR1
 965  004662' 104420  000000' 000676'         OTOA$,BEGIN,UNITNO,ADR1
 966  004670' 007024'
 967                                          ;*************************************************************
 968                                          ;*************************************************************
 969                                                                  ;CONVERT BADCYL TO ASCII AND
```

```
                                                ;STORE AT CYLNO
004672' 104420  000000' 000650'         OTOA$,BEGIN,BADCYL,CYLNO
004700' 006614'
                                        ;*************************************************************
                                        ;*************************************************************
                                                ;CONVERT BADTRK TO ASCII AND
                                                ;STORE AT TRKNO
004702' 104420  000000' 000654'         OTOA$,BEGIN,BADTRK,TRKNO
004710' 006632'
                                        ;*************************************************************
                                        ;*************************************************************
                                                ;CONVERT BADSEC TO ASCII AND
                                                ;STORE AT SECNO
004712' 104420  000000' 000652'         OTOA$,BEGIN,BADSEC,SECNO
004720' 006650'
                                        ;*************************************************************
004722' 104403  003000' 006736'         MSGN$,BEGIN,BADMES      ;ASCII MESSAGE CALL WITH COMMON HEADER
004730'                         12$:                            ;  END                          ;DRB001
004730' 026727  173702  000552'         CMP     NXTBBK,#ENDBBK  ;  IF MORE SPACE IN TABLE THEN  ;DRB001
004736' 002010                          BGE     13$             ;  BEGIN                        ;DRB001
004740' 016700  173672                  MOV     NXTBBK,R0       ;   R0 POINTS TO NEXT FREE SLOT IN TABLE;DRB001
004744' 016720  173700                  MOV     BADCYL,(R0)+    ;   STUFF CYLINDER NUMBER       ;DRB001
004750' 016720  173664                  MOV     S+6,(R0)+       ;   STUFF REST OF DISK ADDRESS  ;DRB001
004754' 010067  173656                  MOV     R0,NXTBBK       ;   UPDATE POINTER              ;DRB001
004760'                         13$:                            ;  END                          ;DRB001
004760' 000540                          BR      7$              ;  NO MESSAGE "HARD ERROR" SO NO RE-TRY ;DRB001
004762'                         9$:                             ; END (* BAD SPOT *)            ;DRB001
004762' 032767  020000  173562          BIT     #BIT13,S        ; IF MASSBUSS CTRL PARITY ERROR THEN ;DRB001
004770' 001404                          BEQ     16$             ; BEGIN                         ;DRB001
004772' 104403  000000' 006700'         MSGN$,BEGIN,MCPERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
005000' 000505                          BR      6$              ;  HARD ERROR                   ;DRB001
005002'                         16$:                            ; END                           ;DRB001
005002' 032767  000400  173552          BIT     #BIT8,S+10      ; IF MASSBUSS DATA PARITY ERROR THEN ;DRB001
005010' 001404                          BEQ     17$             ; BEGIN                         ;DRB001
005012' 104403  000000' 006710'         MSGN$,BEGIN,MDPERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
005020' 000475                          BR      6$              ;  HARD ERROR                   ;DRB001
005022'                         17$:                            ; END                           ;DRB001
005022' 032767  040000  173532          BIT     #BIT14,S+10     ; IF WRITE-CHECK ERROR THEN      ;DRB001
005030' 001404                          BEQ     18$             ; BEGIN                         ;DRB001
005032' 104403  000000' 006672'         MSGN$,BEGIN,TRERR       ;ASCII MESSAGE CALL WITH COMMON HEADER
005040' 000514                          BR      10$             ;  SOFT ERROR                   ;DRB001
005042'                         18$:                            ; END                           ;DRB001
005042' 032767  040000  173502          BIT     #BIT14,S        ; IF ANY OTHER TRANSFER ERROR THEN ;DRB001
005050' 001404                          BEQ     19$             ; BEGIN                         ;DRB001
005052' 104403  000000' 006672'         MSGN$,BEGIN,TRERR       ;ASCII MESSAGE CALL WITH COMMON HEADER
005060' 000455                          BR      6$              ;  HARD ERROR                   ;DRB001
005062'                         19$:                            ; END                           ;DRB001
005062' 032767  024000  173474          BIT     #BIT13!BIT11,S+12 ; IF OTHER HARD ERROR THEN    ;DRB001
005070' 001034                          BNE     20$             ;                               ;DRB001
005072' 032767  000200  173466          BIT     #BIT7,S+14      ;                               ;DRB001
005100' 001405                          BEQ     21$             ;                               ;DRB001
005102' 032767  000400  173456          BIT     #BIT8,S+14      ;                               ;DRB001
005110' 001070                          BNE     10$             ; (* SOFT-=SPECIAL CASE *)      ;DRB001
005112' 000423                          BR      20$             ;                               ;DRB001
005114' 032767  057057  173444  21$:    BIT     #57057,S+14     ;                               ;DRB001
005122' 001017                          BNE     20$             ;                               ;DRB001
```



```
005124' 005767  173462                  TST     S+40            ;                               ;DRB001
005130' 001014                          BNE     20$             ;                               ;DRB001
005132' 032767  000156  173454          BIT     #156,S+42       ;                               ;DRB001
005140' 001010                          BNE     20$             ;                               ;DRB001
005142' 032767  000200  172646          BIT     #RH70,SR1       ;                               ;DRB001
005150' 001405                          BEQ     22$             ;                               ;DRB001
005152' 032777  140000  174662          BIT     #140000,@PHCS3  ;                               ;DRB001
005160' 001401                          BEQ     22$             ;                               ;DRB001
005162'                         20$:                            ; BEGIN                         ;DRB001
005162' 000414                          BR      6$              ;  HARD ERROR                   ;DRB001
005164'                         22$:                            ; END                           ;DRB001
005164' 032767  120400  173374          BIT     #120400,S+14    ; IF OTHER SOFT ERRORS THEN      ;DRB001
005172' 001004                          BNE     23$             ;                               ;DRB001
005174' 032767  140000  173412          BIT     #140000,S+42    ;                               ;DRB001
005202' 001401                          BEQ     24$             ;                               ;DRB001
005204'                         23$:                            ; BEGIN                         ;DRB001
005204' 000432                          BR      10$             ;  SOFT ERROR                   ;DRB001
005206'                         24$:                            ; END                           ;DRB001
                                        ;(* IF WE FALL THRU TO HERE, WE HAVE AN UNKNOWN ERROR *) ;DRB001
005206' 104403  000000' 006754'         MSGN$,BEGIN,UNKNWN      ;ASCII MESSAGE CALL WITH COMMON HEADER
005214'                         6$:     ; (* PROCESS HARD ERROR *)                             ;DRB001
005214' 016700  174572                  MOV     PHDB,R0         ; SAVE RHDB ADDRESS             ;DRB001
005220' 105767  173336                  TSTB    S+10            ; IF OR NOT SET THEN            ;DRB001
005224' 100407                          BMI     25$             ; BEGIN                         ;DRB001
005226' 012767  000706' 174556          MOV     #ZERO,RHDB      ;  FAKE ADDRESS TO PREVENT DLT ON ACCESS;DRB001
005234' 012767  000001  172644          MOV     #1,ERRTYP       ;  DATA PROBLEM                 ;DRB001
005242' 000402                          BR      26$             ; END                           ;DRB001
005244'                         25$:                            ; ELSE                          ;DRB001
005244' 005067  172636                  CLR     ERRTYP          ;  SOME OTHER ERROR             ;DRB001
005250'                         26$:                            ;                               ;DRB001
                                        ;*************************************************************
005250' 104405  000000' 001716'         HRDER$,BEGIN,TABLE      ;
                                        ;*************************************************************
005256' 010067  174530                  MOV     R0,RHDB         ; RESTORE DATA BUFFER ADDRESS   ;DRB001
005262' 052767  000200  173400  7$:     BIS     #BIT7,FLAG      ; HARD ERROR FLAG : NO RETRY    ;DRB001
005270' 000423                          BR      27$             ; DO COMMON ERROR EXIT          ;DRB001
                                                                                                ;DRB001
005272'                         10$:    ; (* PROCESS SOFT ERROR *)                             ;DRB001
005272' 016700  174514                  MOV     RHDB,R0         ; SAVE RHDB ADDRESS             ;DRB001
005276' 105767  173260                  TSTB    S+10            ; IF OR NOT SET THEN            ;DRB001
005302' 100407                          BMI     28$             ; BEGIN                         ;DRB001
005304' 012767  000706' 174500          MOV     #ZERO,RHDB      ;  DON'T TRY TO ACCESS REGISTER ;DRB001
005312' 012767  000001  172566          MOV     #1,ERRTYP       ;  DATA ERROR                   ;DRB001
005320' 000402                          BR      29$             ; END                           ;DRB001
005322'                         28$:                            ; ELSE                          ;DRB001
005322' 005067  172560                  CLR     ERRTYP          ;  OTHER ERROR TYPE             ;DRB001
005326'                         29$:                            ;                               ;DRB001
                                        ;*************************************************************
005326' 104406  000000' 001716'         SOFER$,BEGIN,TABLE      ;
                                        ;*************************************************************
005334' 005067  173330                  CLR     FLAG            ; NO HARD ERROR                 ;DRB001
                                                                                                ;DRB001
005340' 004767  000004          27$:    JSR     PC,NOTRDY       ; CLEAR OUT ERRORS, MAKE SURE DRIVE'S OK;DRB001
005344' 000261                          SEC                     ; SET ERROR RETURN              ;DRB001
005346' 000205                  28$:    RTS     R5              ; RETURN HOMEWARD,              ;DRB001
```

```
 1082
 1083  005350' 012767  077777  173334  NOTRDY: MOV     #77777,CLK      ; SET THE TIMER
 1084  005356'                         4$:
 1085  005356' 032777  004000  174404          BIT     #BIT11,@RHCS1   ;DO I HAVE THE DRIVE DVA?
 1086  005364' 001406                          BEQ     6$              ;NO
 1087  005366' 010046                          MOV     R0,-(SP)
 1088  005370' 012600                          MOV     (SP)+,R0        ;JUST WASTE A LITTLE TIME
 1089  005372' 032777  004000  174370          BIT     #BIT11,@RHCS1           ;STILL GOT IT?
 1090  005400' 001017                          BNE     2$
 1091  005402'                         6$:
 1092  005402' 104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,,,,
 1093  005406' 104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION,
 1094  005412' 104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,,,,
 1095  005416' 104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION,
 1096  005422' 005367  173264                  DEC     CLK             ;COUNT # OF TRIES
 1097  005426' 001353                          BNE     4$              ;NOT DONE YET
 1098  005430' 104403  000000' 006762'        MSGN$,BEGIN,NOT ;ASCII MESSAGE CALL WITH COMMON HEADER
 1099  005436' 000426                          BR      7$              ;COULD NOT GET DRIVE
 1100  005440' 004567  175160          2$:     JSR     R5,CLEAR        ;SET THE CONTROLLER AND DRIVE
 1101  005444' 004567  000074                  JSR     P5,READY        ; IS DRIVE READY ?
 1102  005450' 000434                          BR      1$              ; YES, CONTINUE
 1103  005452' 004767  000224          5$:     JSR     PC,ERSUB1       ; LOAD ERROR INFORMATION
 1104  005456' 005000                          CLR     R0              ;MOVE DRIVE REG INTO TABLE
 1105  005460' 016701  174304                  MOV     RHCS1,P1
 1106  005464' 012160  000552'          22$:    MOV     (R1)+,S(R0)
 1107  005470' 005720                          TST     (R0)+
 1108  005472' 022700  000046                  CMP     #46,R0
 1109  005476' 001372                          BNE     22$
 1110  005500' 012767  000006  172400          MOV     #6,ERRTYP       ;DRIVE NOT READY
 1111                                          ;*************************************************************
 1112  005506' 104405  000000' 001716'        HRDER$,BEGIN,TABLE      ; DRIVE  NOT READY
 1113                                          ;*************************************************************
 1114                                                                  ;**-1                   ;DRB001
 1115  005514' 004767  175056          7$:     JSR     PC,RELESE       ;RELEASE THE DRIVE      ;DRB001
 1116  005520' 004767  176674                  JSR     PC,DROP         ; NO, DROP THE DRIVE
 1117  005524' 104403  000000' 007000'        MSGN$,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
 1118  005532' 016706  172274                  MOV     SPOINT,R6
 1119  005536' 000167  174446                  JMP     LOOP2
 1120  005542' 000207                  1$:     RTS     PC              ;RETURN
 1121                                  ;       ------------------------
 1122
 1123
 1124  005544' 016777  173126  174226  READY:  MOV     UNITNO,@RHCS2   ; LOAD UNIT ADDRESS
 1125  005552' 017700  174212                  MOV     @RHCS1,R0
 1126  005556' 032700  004000                  BIT     #BIT11,R0
 1127  005562' 001433                          BEQ     1$
 1128  005564' 017700  174212                  MOV     @RPDS,R0        ; SAVE STATUS IN R0
 1129  005570' 105700                          TSTB    R0              ; DRIVE READY ?
 1130  005572' 100027                          BPL     1$              ; NO
 1131  005574' 032700  000100                  BIT     #BIT6,R0        ; VOLUME VALID ?
 1132  005600' 001424                          BEQ     1$              ; NO
 1133  005602' 032700  000400                  BIT     #BIT8,R0        ; DRIVE PRESENT ?
 1134  005606' 001421                          BEQ     1$              ; NO
 1135  005610' 032700  004000                  BIT     #BIT11,R0       ; WRITE LOCKED ?
 1136  005614' 001016                          BNE     1$              ; YES
 1137  005616' 032700  010000                  BIT     #BIT12,R0       ; MEDIUM ON LINE ?
```



```
 1138  005622' 001413                          BEQ     1$              ; NO
 1139  005624' 032700  040000                  BIT     #BIT14,R0       ; ANY ERRORS ?
 1140  005630' 001010                          BNE     1$              ; YES
 1141  005632' 005700                          TST     R0              ; ATTENTION SET ?
 1142  005634' 100406                          BMI     1$              ; YES
 1143  005636' 032777  004000  174124          BIT     #BIT11,@RHCS1   ;DVA SET?
 1144  005644' 001402                          BEQ     1$              ;BR IF NOT
 1145                                                                  ;**-3                   ;DRB001
 1146  005646' 000261                          SEC                     ; READY                 ;DRB001
 1147  005650' 000401                          BR      2$              ; RETURN                ;DRB001
 1148  005652' 000241                  1$:     CLC                     ; NOT READY             ;DRB001
 1149  005654' 000205                  2$:     RTS     R5              ;RETURN                 ;DRB001
 1150                                  ;       ------------------------
 1151
 1152
 1153  005656' 014167  172224          ERSUB2: MOV     -(R1),ASB       ; LOAD THE DATA
 1154  005662' 010167  172214                  MOV     R1,SBADR        ; LOAD ADDRESS OF DATA WRITTEN
 1155  005666' 014267  172216                  MOV     -(R2),AWAS      ; LOAD THE DATA
 1156  005672' 010267  172206                  MOV     R2,WASADR       ; LOAD ADDRESS OF DATA READ
 1157  005676' 005721                          TST     (R1)+           ; RESET REG. 1
 1158  005700' 005722                          TST     (R2)+           ; RESET REG. 2
 1159
 1160  005702' 016767  174052  172170  ERSUB1: MOV     RHCS1,CSRA      ; LOAD ADR OF CURRENT CSR
 1161  005710' 017767  174054  172164          MOV     @RHCS1,ACSR     ; LOAD CONTENTS OF CURRENT CSR
 1162  005716' 000207                          RTS     PC              ; RETURN
 1163
 1164
 1165  005720' 017700  174044          REZET:  MOV     @RHCS1,R0
 1166  005724' 032700  004000                  BIT     #BIT11,R0
 1167  005730' 001005                          BNE     3$
 1168  005732' 104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,,,,
 1169  005736' 104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION,
 1170  005742' 000766                          BR      REZET
 1171  005744' 012777  000040  174026  3$:     MOV     #BIT5,@RHCS2    ; ISSUE AN RH11 INIT
 1172  005752' 012767  077777  172732          MOV     #77777,CLK      ; SET THE TIMER
 1173  005760' 105777  174004          1$:     TSTB    @RHCS1          ; CONTROLLER READY ?
 1174  005764' 100417                          BMI     2$              ; YES, CONTINUE
 1175  005766' 104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR,,,,
 1176  005772' 104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION,
 1177  005776' 005367  172710                  DEC     CLK             ; WAIT SOME MORE ?
 1178  006002' 001366                          BNE     1$              ; YES
 1179  006004' 012767  000003  172074          MOV     #3,ERRTYP       ;CONTROLLER NOT READY
 1180                                          ;*************************************************************
 1181  006012' 104405  000000' 000000         HRDER$,BEGIN,NULL       ; CONTROLLER NOT READY
 1182                                          ;*************************************************************
 1183  006020' 104410  000000'                END$,BEGIN              ;
 1184  006024' 017746  173756          2$:     MOV     @RPAS,-(SP)
 1185  006030' 012677  173752                  MOV     (SP)+,@RPAS
 1186  006034' 000207                          RTS     PC              ; RETURN
 1187                                  ;       ------------------------
 1188
 1189
 1190  006036' 016700  171744          SETUP:  MOV     ADDR,R0         ; GET DEVICE ADDRESS
 1191  006042' 010067  173722                  MOV     R0,RHCS1        ; GENERATE REGISTER ADDRESSES
 1192  006046' 062700  000002                  ADD     #2,R0
 1193  006052' 010067  173714                  MOV     R0,RHWC
```

```
1194  006056' 062700  000002            ADD     #2,R0
1195  006062' 010067  173706            MOV     R0,RHBA
1196  006066' 062700  000002            ADD     #2,R0
1197  006072' 010067  173700            MOV     R0,RPDA
1198  006076' 062700  000002            ADD     #2,R0
1199  006102' 010067  173672            MOV     R0,RHCS2
1200  006106' 062700  000002            ADD     #2,R0
1201  006112' 010067  173664            MOV     R0,RPDS
1202  006116' 062700  000002            ADD     #2,R0
1203  006122' 010067  173656            MOV     R0,RPER1
1204  006126' 062700  000002            ADD     #2,R0
1205  006132' 010067  173650            MOV     R0,RPAS
1206  006136' 062700  000002            ADD     #2,R0
1207  006142' 010067  173642            MOV     R0,RPLA
1208  006146' 062700  000002            ADD     #2,R0
1209  006152' 010067  173634            MOV     R0,RHDB
1210  006156' 062700  000002            ADD     #2,R0
1211  006162' 010067  173626            MOV     R0,RPMR
1212  006166' 062700  000002            ADD     #2,R0
1213  006172' 010067  173620            MOV     R0,RPDT
1214  006176' 062700  000002            ADD     #2,R0
1215  006202' 010067  173612            MOV     R0,RPSN
1216  006206' 062700  000002            ADD     #2,R0
1217  006212' 010067  173604            MOV     R0,RPOF
1218  006216' 062700  000002            ADD     #2,R0
1219  006222' 010067  173576            MOV     R0,RPDC
1220  006226' 062700  000002            ADD     #2,R0
1221  006232' 010067  173570            MOV     R0,RPCC
1222  006236' 062700  000002            ADD     #2,R0
1223  006242' 010067  173562            MOV     R0,RPER2
1224  006246' 062700  000002            ADD     #2,R0
1225  006252' 010067  173554            MOV     R0,RPER3
1226  006256' 062700  000002            ADD     #2,R0
1227  006262' 010067  173546            MOV     R0,RPEC1
1228  006266' 062700  000002            ADD     #2,R0
1229  006272' 010067  173540            MOV     R0,RPEC2
1230                                                                ;**-7
1231  006276' 032767  001000  171552    BIT     #ADDR22,RES1    ;DO WE HAVE 22-BIT ADDR SUPPORT?  ;DRB001
1232  006304' 001415                    BEQ     1$              ;NO                               ;DRB001
1233  006306' 032767  000200  171502    BIT     #RH70,SR1       ;IS THIS AN RH70                  ;DRB001
1234  006314' 001411                    BEQ     1$              ;NO                               ;DRB001
1235  006316' 062700  000002            ADD     #2,R0
1236  006322' 010067  173512            MOV     R0,RHBAE
1237  006326' 062700  000002            ADD     #2,R0
1238  006332' 010067  173504            MOV     R0,RHCS3
1239  006336' 000403                    BR      2$                                                ;DRB001
1240  006340' 042767  001000  171510  1$: BIC     #ADDR22,RES1    ; CAN'T USE 22 BIT ADDRESSING     ;DRB001
1241  006346' 016700  171436          2$: MOV     VECTOR,R0       ; GET VECTOR ADDRESS
1242  006352' 012720  003036'            MOV     #NTRUPT,(R0)+   ; SET POINTER JUST IN CASE
1243  006356' 116710  171430            MOVB    BR1,(R0)        ; SET PRIORITY
1244  006362' 000207                    RTS     PC              ; RETURN
1245
```



```
1246
1247                                .SBTTL  MODULE MESSAGES                          ;**-40
1248  006364' 042440  051122  051117  MES1:  .ASCIZ  ' ERROR%'
1249  006372' 000045
1250  006374' 050040  051101  052111  MES2:  .ASCIZ  ' PARITY'
1251  006402' 000131
1252  006404' 042040  044522  042526  MES3:  .ASCIZ  ' DRIVE '
1253  006412' 000040
1254  006414' 046440  051501  041123  MES4:  .ASCIZ  ' MASSBUS'
1255  006422' 051525     000
1256  006425'    040  040504  040524  MES5:  .ASCIZ  ' DATA'
1257  006432'    000
1258  006433'    040  042522  051124  MES6:  .ASCIZ  ' RETRY EXCEEDED%'
1259  006440' 020131  054105  042503
1260  006446' 042105  042105  000045
1261  006454' 046040  052101  000105  MES7:  .ASCIZ  ' LATE'
1262  006462' 047516  020124  042522  MES8:  .ASCIZ  'NOT READY%'
1263  006470' 042101  022531     000
1264  006475'    040  047503  046125  MES9:  .ASCIZ  ' COULD NOT GET'
1265  006502' 020104  047516  020124
1266  006510' 042507  000124
1267  006514' 042040  047522  050120  MES10: .ASCIZ  ' DROPPED'
1268  006522' 042105     000
1269  006525'    072  047040  052117  MES11: .ASCIZ  ': NOT RP04/5/6%'
1270  006532' 051040  030120  027464
1271  006540' 027465  022466     000
1272  006545'    040  051124  047101  MES12: .ASCIZ  ' TRANSFER'
1273  006552' 043123  051105     000
1274  006557'    045     000         MES13: .ASCIZ  '%'
1275  006561'    072  041040  042101  MES15: .ASCIZ  ': BADSPOT AT (OCTAL) CYL: '
1276  006566' 050123  052117  040440
1277  006574' 020124  047450  052103
1278  006602' 046101  020051  054503
1279  006610' 035114  000040
1280  006614' 000002                    CYLNO:  .BLKB   2
1281  006616' 020040  020040  020054  MES16: .ASCIZ  '    , TRK: '
1282  006624' 051124  035113  000040
1283  006632' 000004                    TRKNO:  .BLKB   4
1284  006636' 020040  020054  042523  MES17: .ASCIZ  '  , SEC: '
1285  006644' 035103  000040
1286  006650' 000004                    SECNO:  .BLKB   4
1287  006654' 020040  000045  MES18: .ASCIZ  ' %'
1288  006660' 052440  045516  047516  MES19: .ASCIZ  ' UNKNOWN'
1289  006666' 047127     000
1290          006672'                 .EVEN
1291
1292  006672' 006545'                 TRERR:  MES12
1293  006674' 006364'                         MES1
1294  006676' 177777                         177777
1295
1296  006700' 006414'                 MCPERR: MES4
1297  006702' 006374'                         MES2
1298  006704' 006364'                         MES1
1299  006706' 177777                         177777
1300
1301  006710' 006414'                 MDPERR: MES4
```

```
  1302  006712' 006425'                MES5                                    ;DRB001
  1303  006714' 006374'                MES2                                    ;DRB001
  1304  006716' 006364'                MES1                                    ;DRB001
  1305  006720' 177777                 177777                                  ;DRB001
  1306                                                                          ;DRB001
  1307  006722' 006433'          EXCED:  MES6                                  ;DRB001
  1308  006724' 177777                 177777                                  ;DRB001
  1309                                                                          ;DRB001
  1310  006726' 006425'          DLTERR: MES5                                  ;DRB001
  1311  006730' 006454'                MES7                                    ;DRB001
  1312  006732' 006364'                MES1                                    ;DRB001
  1313  006734' 177777                 177777                                  ;DRB001
  1314                                                                          ;DRB001
  1315  006736' 006404'          BADMES: MES3                                  ;DRB001
  1316  006740' 007031'                 NUMB                                    ;DRB001
  1317  006742' 006561'                 MES15                                   ;DRB001
  1318  006744' 006616'                 MES16                                   ;DRB001
  1319  006746' 006636'                 MES17                                   ;DRB001
  1320  006750' 006654'                 MES18                                   ;DRB001
  1321  006752' 177777                 177777                                  ;DRB001
  1322                                                                          ;DRB001
  1323  006754' 006660'          UNKNWN: MES19                                 ;DRB001
  1324  006756' 006364'                 MES1                                   ;DRB001
  1325  006760' 177777                 177777                                  ;DRB001
  1326                                                                          ;DRB001
  1327  006762' 006404'          NOT:    MES3                                  ;DRB001
  1328  006764' 006462'                 MES8                                   ;DRB001
  1329  006766' 177777                 177777                                  ;DRB001
  1330                                                                          ;DRB001
  1331  006770' 006475'          TOUT:   MES9                                  ;DRB001
  1332  006772' 006404'                 MES3                                   ;DRB001
  1333  006774' 006557'                 MES13                                  ;DRB001
  1334  006776' 177777                 177777                                  ;DRB001
  1335                                                                          ;DRB001
  1336  007000' 006404'          DRP:    MES3                                  ;DRB001
  1337  007002' 007031'                 NUMB                                   ;DRB001
  1338  007004' 006514'                 MES10                                  ;DRB001
  1339  007006' 006557'                 MES13                                  ;DRB001
  1340  007010' 177777                 177777                                  ;DRB001
  1341                                                                          ;DRB001
  1342  007012' 006404'          BADTYP: MES3                                  ;DRB001
  1343  007014' 007031'                 NUMB                                   ;DRB001
  1344  007016' 006514'                 MES10                                  ;DRB001
  1345  007020' 006525'                 MES11                                  ;DRB001
  1346  007022' 177777                 177777                                  ;DRB001
  1347                                                                          ;DRB001
  1348  007024' 000005            ADR1:   .BLKB   5                           ;DRB001
  1349  007031'    040     000    NUMB:   .ASCIZ  / /                         ;DRB001
  1350          007034'                  .EVEN                                 ;DRB001
  1351                                                                          ;DRB001
  1352                                  ;-----------------------------          ;DRB001
  1353                                                                          ;DRB001
  1354          000001                    .END                                  ;DRB001
```



```
ACSR     000102R          231#   1161*
ADDR     000006R          197#   1190
ADDR22=  001000           260#    588    1231    1240
ADR1     007024R          911     965    1348#
ASB      000106R          235#   1153*
ASTAT    000104R          233#
AWAS     000110R          236#   1155*
BADCYL   000650R          393#    961*    971     991
BADMES   006736R          986    1315#
BADSEC   000652R          394#    959*    983
BADSPT   000252R          261#    384     756
BADTRK   000654R          395#    960*    977
BADTYP   007012R          875    1342#
BDSPT =  000100           366#    956
BEGIN    000000R          194#    490     493     505     507     517     568     569     595     603     612     627     679
                          875     911     939     965     971     977     983     986     999    1004    1009    1014    1045
                         1057    1074    1092    1093    1094    1095    1098    1112    1117    1168    1169    1175    1176
                         1181    1183
BITTAB   003242R          606     645     657#
BIT0  =  000001           260#    733     841     851     858     868
BIT1  =  000002           260#
BIT10 =  002000           260#
BIT11 =  004000           260#   1017    1085    1089    1126    1135    1143    1166
BIT12 =  010000           260#    584    1137
BIT13 =  020000           260#    997    1017
BIT14 =  040000           260#    583    1007    1012    1139
BIT15 =  100000           260#    949
BIT2  =  000004           260#    363
BIT3  =  000010           260#
BIT4  =  000020           260#    364     947
BIT5  =  000040           260#    365    1171
BIT6  =  000100           260#    366     945    1131
BIT7  =  000200           260#    367    1019    1060
BIT8  =  000400           260#    952    1002    1021    1133
BIT9  =  001000           260#
BPORT =  000020           364#    731
BREAK$=  104407           260#    568     569    1092    1093    1094    1095    1168    1169    1175    1176
BR1      000012R          199#   1243
BR2      000013R          200#
BTOD$ =  104421           260#
CDATA$=  104412           260#    517
CDERCT   000144R          250#
CDWDCT   000146R          251#
CLEAP    002624R          573#   1100
CLK      000712R          413#   1083*   1096*   1172*   1177*
CLRCOM   004410R          896#    898
CLRRB    004400R          894#
CONFIG   000056R          219#
CSRA     000100R          229#   1160*
CYCLE    002254R          500     511#
CYL      000644R          388#    483*    682*    709*    715     719*    724     727*    730     742*
CYLLIM   000640R          386#    715     725     735     842*    852*    859*    869*
CYLNDR   000642R          387#    543     553     563     730*    735*    737     739*    760     776
CYLNO    006614R          971    1280#
DATCK$=  104411           260#
DATER$=  104404           260#
```

```
DLATE = 000004          363#    937
DLTCNT  000700R         408#    936#
DLTERR  006726R         939    1310#
DRIVE   000674R         405#
DROP    004420R         420     874     901#   1116
DRP     007000R        1117    1336#
DSKADR  000646R         389#    484#    542     552     562     720#    743#
DVICE   000672R         404#    467#    477     503     645     907#
DVID1   000014R         201#    467
EA22    000634R         382#    597     598#    599#    600
ENDBBK  000552R         361#    988
ENDIT$= 104413          260#    507
END$  = 104410          260#    505    1183
ERROPS  004456R         615     917#
ERRTYP  000106R         234#   1051#   1054#   1068#   1071#   1110#   1179#
ERSUB1  005702R         933    1103    1160#
ERSUB2  005656R        1153#
EXCED   006722R         627    1307#
EXIT$ = 104400          260#    603
FERADR  000710R         412#    537#    547#    557#    625
FLAG    000670R         403#    620     626#    733     841#    851#    858#    868#   1060#   1076#
FREE    000150R         252#
FUNC    000702R         409#    536#    546#    556#    601#    602
GETDVT  004136R         653     833#
GETPA$= 104415          260#    490
GO      002710R         545     555     565     587#
GWBUF$= 104414          260#    493
HRDCNT  000044R         214#
HRDER$= 104405          260#   1057    1112    1181
HRDPAS  000050R         216#
ICONT   000036R         211#
ICOUNT  000040R         212#
IDNUM   000122R         241#
IMODX,= 000000          253#    494
INIT    000030R         208#
INTR    000120R         240#    466#
LASCYL  000656R         397#    737     739     828#    843#    853#    860#    870#
LASSEC  000662R         399#    748     750     831#
LASTRK  000660R         398#    745     747     830#
LOOP1   002176R         492#    509
LOOP2   002210R         495#    501     631    1119
MAP22$= 104416          260#    595
MCPERR  006700R         999    1296#
MDPERR  006710R        1004    1301#
MES1    006364R        1248#   1293    1298    1304    1312    1324
MES10   006514R        1267#   1338    1344
MES11   006525R        1269#   1345
MES12   006545R        1272#   1292
MES13   006557R        1274#   1333    1339
MES15   006561R        1275#   1317
MES16   006616R        1281#   1318
MES17   006636R        1284#   1319
MES18   006654R        1287#   1320
MES19   006660R        1288#   1323
MES2    006374R        1250#   1297    1303
MES3    006404R        1252#   1315    1327    1332    1336    1342
```



```
MES4    006414R        1254#   1296    1301
MES5    006425R        1256#   1302    1310
MES6    006433R        1258#   1307
MES7    006454R        1261#   1311
MES8    006462R        1262#   1328
MES9    006475R        1264#   1331
MODLUS  004366R         703     712     726     883#
MODNAM  000000R         195#
MODSP   000252R         209     258#
MSGN$ = 104403          260#    627     875     939     986     999    1004    1009    1014    1045    1098    1117
MSGS$ = 104402          260#
MSG$  = 104401          260#
NORAND= 000040          365#    672     717     740
NOT     006762R        1098    1327#
NOTRDY  005350R         652    1078    1083#
NTRUPT  033036R         604#   1242
NULL  = 000000          260#   1181
NUMB    007031R        1316    1337    1343   1349#
NXTBBK  000636R         384#    757     988     990     993#
OPEN  = 000000          196     202     203     204     205     222     223     224     225     226     227     228     229
                        231     233     235     236     238     239     240     243     244     246     247     249     250
                        251     252     260#
OTOA$ = 104420          260#    911     965     971     977     983
PASCNT  000034R         210#
PA18    000626R         379#    590#    595
PA22    000632R         381#    596
PICKBK  003252R         513     629     663#    788
PICKDR  003160R         496     638#
PIRQ$ = 000004          260#    612
POPSP = 005726          260#
POPSP2= 022626          260#
PRTY  = 000000          260#
PRTY0 = 000000          200     260#
PRTY1 = 000040          260#
PRTY2 = 000100          260#
PRTY3 = 000140          260#
PRTY4 = 000200          260#
PRTY5 = 000240          199     260#
PRTY6 = 000300          260#
PRTY7 = 000340          260#
PS    = 177776          260#
PSW   = 177776          260#
PUSH  = 005746          260#
PUSH2 = 024646          260#
RAND$ = 104417          260#    679
RANNUM  000054R         218#    680     683     687
RBUF    000716R         242     415#    894
RBUFEA  000130R         244#    565
RBUFPA  000126R         243#    517     561
RBUFSZ  000132R         245#    558     895
RBUFVA  000124R         242#    490
READ    002502R         516     556#    557
READY   005544R         650    1101   1124#
RELESE  002576R         519     566#   1115
RESTRT  002170R         237     488#
RES1    000056R         220#    588    1231   1240#
```

```
RES2    000060R          221#
REXET   005720R          487    1165#   1170
RHBA    001774R          442#    541*    551*    561*    590     596*    1195*
RHBAE   002040R          460#    597*   1236*
RHCS1   001770R          440#    567*    574*    577*    578     583*    602*    918     926     1085    1089    1105    1125
                        1143    1160    1161    1165    1173    1191*
RHCS2   002000R          444#    566*    573*    587*    838*   1124*   1171*   1199*
RHCS3   002042R          461#   1032    1238*
RHDB    002012R          449#   1047    1050*   1059*   1064    1067*   1209*
RHWC    001772R          441#    540*    550*    560*   1193*
RH70  = 000200           367#   1030    1233
RPAS    002006R          447#    581     582*    607     609*   1184    1185*   1205*
RPCC    002026R          455#   1221*
RPDA    001776R          443#    542*    552*    562*   1197*
RPDC    002024R          454#    543*    553*    563*   1219*
RPDS    002002R          445#   1128    1201*
RPDT    002016R          451#    839     846     848     856     863     865    1213*
RPEC1   002034R          458#   1227*
RPEC2   002036R          459#   1229*
RPER1   002004R          446#   1203*
RPER2   002030R          456#   1223*
RPER3   002032R          457#   1225*
RPLA    002010R          448#   1207*
RPMR    002014R          450#   1211*
RPOF    002022R          453#    584*   1217*
RPSN    002020R          452#   1215*
RSTRT   000112R          237#
S       000552R          369#    418     419     420     421     422     423     424     425     426     427     428     429
                         430     431     432     433     434     435     436     437     929*    934     945     947     949
                         952     959     960     961     992     997    1002    1007    1012    1017    1019    1021    1024
                        1026    1028    1037    1039    1048    1065    1106*
SBADR   000102R          230#   1154*
SECNO   006650R          983    1286#
SECTOR  000646R          390#    674*    686*    698     701     704*    748     750*    758     784
SETUP   006036R          486    1190#
SIZSEC  000666R          402#    762     815*
SIZTRK  000664R          401#    768     809*    812*    824
SOFCNT  000042R          213#
SOFER$= 104406           260#   1074
SOFPAS  000046R          215#
SPOINT  000032R          209#   1118
SPSIZ = 000040             1#    253
SR1     000016R          202#    672     717     731     740     937     956    1030    1233
SR2     000020R          203#
SR3     000022R          204#
SR4     000024R          205#
START   002044R          208     464#
STAT    000026R          207#
SVR0    000062R          222#
SVR1    000064R          223#
SVR2    000066R          224#
SVR3    000070R          225#
SVR4    000072R          226#
SVR5    000074R          227#
SVR6    000076R          228#
SY8CNT  000052R          217#
```



```
TABLE   001716R          417#   1057    1074    1112
TIMER   000704R          410#
TOUT    006770R         1331#
TRACK   000647R          391#    696*    700*    707     710     713*    745     747*    759     780
TRERR   006672R         1009    1014   1292#
TRKNO   006632R          977    1283#
TRPDFD= 000022           260#
TRY     000714R          414#    499*    622*    623
UNITNO  000676R          406#    479*    492*    566     573     587     605     630*    639*    640     644     838     902
                         911     965    1124
UNKNWN  006754R         1045   1323#
VECTOR  000010R          198#   1241
WASADR  000104R          232#   1156*
WBUFEA  000136R          247#    545     555
WBUFPA  000134R          246#    541     551
WBUFRQ  000140R          248#
WBUFSZ  000142R          249#    538     548     806
WDFR    000116R          239#    465*
WDTO    000114R          238#    464*
WRITCK  002406R          515     546#    547
WRITE   002312R          514     536#    537
WRTLIM  004016R          498     796#
XFLAG   000005R          196#
XMEM    000630R          380#    545*    555*    565*    591*    592*    593*    594*    600*    601
ZERO    000706R          411#   1050    1067
.     = 007034R          415#    518    1280#   1283#   1286#   1290#   1348#   1350#

. ABS.  000000     000
        007034     001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XRPDC0,XRPDC0/SOL/CRF:SYM=DDXCOM,XRPDC0
 RUN-TIME:  7 12 1 SECONDS
 RUN-TIME RATIO: 40/20=1.9
 CORE USED:  7K  (13 PAGES)
```