```
                .REM _

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

                  PRODUCT CODE:       AC-E785D-MC

                  PRODUCT NAME:       CXRKBD0 RK611/RK06,RK07 MOD

                  PRODUCT DATE:       SEPTEMBER 1978

                  MAINTAINER:         DEC/X11 SUPPORT GROUP

                  THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
                  WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT
                  BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT
                  CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT
                  MAY APPEAR IN THIS MANUAL.

                  THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE
                  PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER
                  SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS
                  COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY
                  OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

                  DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR
                  THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS
                  NOT SUPPLIED BY DIGITAL.

                COPYRIGHT (C) 1976,1978 DIGITAL EQUIPMENT CORPORATION
                _
```

```
                        .REM _

1.      ABSTRACT
        --------

        RKB IS AN IOMODX THAT EXERCISES RK06/RK07 DRIVES ON A
        RK611 CONTROLLER. IT EXERCISES THE DRIVES BY DOING WRITES,
        WRITE-CHECKS, READS, AND IN-CORE COMPARISONS.
        IN DUAL PORT MODETHESE ARE DONE BACK AND FORTH BETWEEN TWO PORTS OF THE DRIVE.
        ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE TTY.

2.      REQUIREMENTS
        ------------

        HARDWARE:  1 TO 8 RK06/RK07 DISK DRIVES WITH TWO RK611 CONTROLLERS

        SOFTWARE:   MUST USE THE FOLLOWING OR LATER REVISIONS OF MONITOR
                  STANDARD MONITOR:     QABL
                  11/70 MONITOR:        QAGC
                  SHORT MONITOR:        QAED

        STORAGE:: RKB REQUIRES:
                        1. DECIMAL WORDS: 2073
                        2. OCTAL WORDS: 04031
                        3. OCTAL BYTES: 10062


3.      PASS DEFINITION
        ---------------

        SINGLE PORT:

        ONE PASS OF THE RKB MODULE CONSISTS OF 765 CYCLES OF THE BASIC TEST
        SEQUENCE (WRITE, WRITE-CHECK, READ, DATA-CHECK). THE TEST SEQUENCE
        WRITES 1024 WORDS, WRITE-CHECKS SAME, READS THE FIRST 256 WORDS,
        AND DATA-CHECKS SAME.

        DUAL PORT:

        ONE PASS OF THE RKB MODULE CONSISTS OF 300 CYCLES OF THE BASIC TEST
        SEQUENCE.  A-PORT DOES A WRITE, WRITE-CHECK,READ,DATA COMPARE
        ON THE CURRENT SECTOR (CALLED BLK1 THROUGHOUT PROGRAM) AND THEN
        WRITES BLK 0 WITH BIT 4 SET IN THE FLAG WORD (THE FIRST WORD OF
        BLOCK ZERO), AND THE SECOND WORD WITH THE CURRENT BLK.
        A-PORT THEN SITS IN A LOOP, PERIODICALLY READING BLK 0 TO
        SEE IF B-PORT HAS MODIFIED THE FLAG.
        B-PORT SITS IN A LOOP WAITING FOR THE FLAG IN BLK ZERO TO
        BE WRITTEN WITH BIT4 SET,  WHEN IT IS B-PORT PICKS THE CURRENT
        BLK FROM THIS DATA READ FROM BLK 0 AND GOES TO THAT SECTOR
        AND READS THE DATA THAT A-PORT JUST WROTE.  THEN B-PORT WRITES
        THE DATA BACK AND WRITE CHECKS IT.  NEXT IT RE-WRITES BLK 0
        WITH BIT4 CLEARED AND WITH BIT2 SET, INDICATING TO A-PORT THAT HE IS DONE.
        A-PORT AFTER READING BLK 0 AND SEEING THE FLAGS REVERSED, RE-READS
        THE DATA WHICH IT HAD WRITTEN AND AGAIN DOES AN IN-CORE COMPARE.
        THIS VERIFIES THE ABILITY OF A-PORT TO  RE-READ DATA WHICH
        IT HAD ORIGINALLY WROTE BUT THAT HAD BEEN READ AND RE-WRITTEN
```

BY B-PORT. B-PORT DOES NOT DO ANY IN CORE COMPARISONS.

FAULTY SINCE A-PORT WAS JUST ABLE TO SUCCESSFULLY USE THIS BLK.

16 LOCATIONS STARTING AT BADSPT ARE REVERVED FOR ENTRY OF 16 BADSPOTS BY USER
ON ERROR TYPEOUTS WHICH DUMP THE RK REGRISTERS, THE LAST ITEM
TYPED IN THE TABLE IS THE CURRENT BLK NUMBER. ENTER THIS INTO
THE BAD BLK TABLE TO AVOID ERRORS FROM KNOWN MEDIA BAD SPOTS.
THIS TABLE ONLY MAKES SENSE ON THE A-PORT SIDE SINCE B-PORT
ALWAYS GETS ITS BLK ADDRESSES FROM A-PORT. IF YOU MODIFY
THE WRITE BUFFER SIZE, YOU MUST ADD SECTORS TO THE BAD BLK
TABLE TO AVOID THE ERRORS. FOR EXAMPLE, IF BLK 3474 IS BAD
AND YOU DOUBLE THE WRITE TRANSFER SIZE TO 1000 OCTAL BYTES
YOU MUST ADD BLK 3473 TO THE TABLE SO THE PROGRAM DOES NO START
A TRANSFER THAT WILL EXTEND ON INTO THE KNOWN BAD BLK.

4. EXECUTION TIME

ONE PASS OF RKB RUNNING ALONE ON A PDP-11/40 TAKES APPROXIMATELY 1 MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 177440, VECTOR: 210, BR1: 5, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP

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

7. MODULE OPERATION

TEST SEQUENCE:

A. SETUP DEVICE REGISTER ADDRESSES AND MODULE VARIABLES
B. RESET ALL DRIVES ON-LINE AND DROP ALL THAT ARE NOT
C. GET A DRIVE ADDRESS
D. IF THE NUMBER OF CYCLES IS COMPLETE FOR THIS DRIVE GO TO C
    ELSE, GET A DISK ADDRESS AND A FRESH BLOCK OF DATA
E. DO A WRITE -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
F. DO A WRITE-CHECK -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
G. DO A READ -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
H. DO A DATA-CHECK -- IF ERRORS, REPORT AND CONTINUE
I. IF END OF PASS, REPORT AND GO TO C, ELSE GO TO D

8. OPERATION OPTIONS

```
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
```

SR1 BIT 9 SET(1):
IF THE RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, A HARD ERROR IS ASSUMED AND THE DRIVE IS DROPPED

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

SR1 BIT 2 SET(1):
ON ENCOUNTERING A BAD SECTOR ALWAYS PRINT A MESSAGE

SR1 BIT 2 CLEAR(0):
ON ENCOUNTERING A BAD SECTOR ONLY PRINT A MESSAGE IF SECTOR IS NOT IN THE BAD SECTOR FILE

SR1 BIT 4 SET(1):
WILL EXERCISE PORT B IN DUAL PORT MODE OF OPERATION

SR1 BIT 4 CLEAR(0):
DUAL PORT WILL EXERCISE PORT A. IN SINGLE PORT THIS BIT MUST BE 0 IN ORDER TO RUN.

SR1 BIT 6 SET(0):
WRITE/READ DATA STARTING AT A RANDOM SECTOR

SR1 BIT 6 CLEAR(1):
WRITE/READ DATA STARTING AT EVERY THIRD SECTOR

SR1 BIT 7 SET(1):
DUAL PORT MODE SELECTED

SR1 BIT 7 CLEAR(0):
SINGLE PORT MODE SELECTED.

9. NON-STANDARD PRINTOUTS

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

B. ERROR MESSAGES DUMP THE CONTENTS OF 13 RK611 REGISTERS IN THE FOLLOWING ORDER:

```
RCS1    RKWC    RKBA    RKDA    RKCS2   RKDS    RKER    RKASO
RKDC    RKECPS  RKECPT  BLK1
```

ERRORS GENERATED IN RKER

```
ILF                     ;ILLEGEAL FUNCTION
SKI                     ;SEEK INCOMPLETE
NXF                     ;NON-EXECTUABLE FUNCTION
DRP                     ;DRIVE PARITY ERROR
FMT                     ;FORMAT
DTY                     ;DRIVE TYPE ERROR
```

```
                        ECH                     ;ERROR CORRECTION HARD
                        BSE                     ;BAD SECTOR ERROR
                        HVR                     ;HEADER VRC ERROR
                        CDE                     ;CYLINDER OVERFLOW ERROR
                        IDA                     ;INVALID DISK ADDRESS ERROR
                        WLE                     ;WRITE LOCK
                        DTE                     ;DRIVE TIMMING ERROR
                        OPI                     ;OPERATION INCOMPLETE
                        UNS                     ;DRIVE UNSAFE
                        DCK                     ;DATA CHECK

                        ERRORS GENERATED IN RKDS

                        ACL                     ;DRIVE AC POWER LOW
                        DCL                     ;DRIVE DC POWER LOW
                        DOT                     ;DRIVE OFF TRACK
                        WRL                     ;WRITE LOCK

                        ERRORS GENERATED IN RKCS2

                        UFE                     ;UNIT FIELD ERROR
                        MDS                     ;MULTIPLE DRIVE SELECT
                        PGE                     ;PROGRAMMING ERROR
                        NEM                     ;NON EXISTANT MEMORY
                        NED                     ;NON EXISTANT DRIVE
                        UPE                     ;UNIBUS PARITY ERROR
                        WCE                     ;WRITE CHECK ERROR
                        DLT                     ;DATA LATE

                        ERRORS GENERATED IN RKCS1

                        CDT                     ;CONTROLLER DRIVE TYPE
                        CTO                     ;CONTROLLER TIMEOUT
                        CFM                     ;CONTROLLER FORMAT
                        SPA                     ;SERIAL  PARITY ERROR
                        CER                     ;CONTROLLER ERROR

                CAUTION: THE RK06 SUBSYSTEM WILL GENERATE TWO (2) INTERRUPTS
                        WHEN ISSUED AN EXPLICIT SEEK COMMAND.  WHEN NO HEAD
                        MOTION IS REQUIRED ONLY ONE INTERRUPT REACHES THE PROCESSOR.
                        HOWEVER; ON SYSTEMS THAT HAVE MANY DEVICES BEING
                        EXERSIZED THE SECOND INTERRUPT MAY OCCUR WHILE THE MODULE
                        IS SERVICING THE FIRST (PRIORITY ZERO) WHICH CAN
                        LEAD TO CONFUSSION.
                -
```



```
                        .NLIST  MC,TTM,CND,MD
                        .LIST   ME



                        .SBTTL  CALL MONITOR FOR SETUP

                        .TITLE  RKBD DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6      23-MAY-78
                        .LIST   BIN
                ;*****************************************************************************
000000'         BEGIN:
000000' 045522  042102  040   MODNAM: .ASCII  /RKBD / ;MODULE NAME.
000005'     000               XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 177440                ADDR:   177440+0                ;1ST DEVICE ADDR.
000010' 000210                VECTOR: 210+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' 001612'               INIT:   START                   ;MODULE START ADDR.
000032' 000252'               SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                PASCNT: 0                       ;PASS COUNTER.
000036' 000765                ICONT:  765                     ;# OF ITERATIONS PER PASS=765
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'                       WASADP:                         ;ADDR OF BAD DATA, OR
000104' 000000                ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106'                       ERRTYP:                         ;TYPE OF ERROR
000106' 000000                ASR:    OPEN                    ;EXPECTED DATA.
000110' 000000                AWAS:   OPEN                    ;ACTUAL DATA.
000112' 001660'               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' 000124  IDNUM:  124                     ;MODULE IDENTIFICATION NUMBER=124
000124' 000366' 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' 002000  WBUFRQ: 1024.                   ;WRITE BUFFER SIZE REQUESTED
000142' 000000  WBUFSZ: OPEN                    ;WRITE BUFFER SIZE AVAILABLE
000144' 000000  CDERCT: OPEN                    ;CDATA/DATCK ERROR COUNT
000146' 000000  CDWDCT: OPEN                    ;CDATA/DATCK WORD COUNT
000150' 000000  FREE:   OPEN                    ;RESERVED FOR FUTURE USE
000252'         MODSP:
                ;******************************************************************************
```



```
                        .SBTTL  CONSTANTS AND STORAGE

000252' 000000  BADSPT: 0                       ;BAD SPOT TABLE, PUT IN THE OCTAL
000254' 000000          0                       ;BLOCK NUMBER, AS SHOWN IN ERROR TYPEOUTS
000256' 000000          0                       ;OF KNOWN BAD BLOCKS.
000260' 000000          0                       ;THERSE'S ROOM FOR 16.
000262' 000000          0                       ;BLK 0 IS NEVER CONSIDERED BAD
000264' 000000          0                       ;IF YOURS IS, YOU CANNOT RUN THIS PROGRAM
000266' 000000          0                       ;GO FIND ANOTHER PACK AND TRY AGAIN
000270' 000000          0
000272' 000000          0
000274' 000000          0
000276' 000000          0
000300' 000000          0
000302' 000000          0
000304' 000000          0
000306' 000000          0
000310' 000000          0
000312' 000000          0
000314' 000000  FLAG:   0                       ;FLAG WHICH IS PASSED BACK BETWEN PORTS
000316' 000000  AVFLAG: 0                       ;DRIVE AVAILABLE TIMER
000320' 000000  TCYL:   0                       ;HOLDING LOCATION FOR CYLINDER ADDR
000322' 000000  TSEC:   0                       ;"      "       "   SECTOR      "
000324' 000000  TIMER:  0                       ;TIMES OUT A PORT
000326' 000000  CYLADR: 0                               ;DESIRED CYLINDER
000330' 000000  SECADR: 0                               ;DESIRED TRACK
000332' 001000  WDCNT:  0
000334' 000000  CYLCNT: 512.                            ;REPETITION COUNT
000336' 000000  BUFADR: 0
000340' 000000  CNT:    0                               ;END OF PASS COUNTER
000342' 000000  FUNC:   0                               ;DESIRED FUNCTION
000344' 000000  XMEM:   0                               ;EXTENDED MEMORY BITS
000346' 000000  DSKADR: 0                               ;CURRENT SELECTED DRIVE NUMBER
000350' 000000  DVICE:  0                               ;AVAILABLE DRIVES
000352' 000000  DRIVE:  0                               ;CURRENT SELECTED DRIVE
000354' 000000  DRYVE:  0                               ;DRIVE COUNTER
000356' 000765  BLK1:   0                               ;SECTOR/BLOCK COUNTER
000360' 000300  ICNT:   .WORD   765             ;ITERATION COUNT
000362' 000300  DPICNT: .WORD   300             ;DUAL PORT ITERATION COUNT
000364' 000000  WCNT1:  0                               ;WORD COUNT FOR WRITES
000366' 000400  WCNT2:  0                               ;WORD COUNT FOR READS
000366' 000000  RBUF:   .BLKW   256.            ;INPUT BUFFER
001366' 000000  CLK:    0                               ;TIMEOUT CLOCK
001370' 000000  CLK1:   0                               ;TIMEOUT CLOCK
001372' 000000  SFTERR: 0                               ;SOFT ERROR FLAG
001374' 000000  INTFLG: 0                               ;INTERRUPT EXPECTED FLAG
001376' 000000  RKCS1:  0                               ;CONTROLLER STATUS REGISTER
001400' 000000  RKWC:   0                               ;WORD COUNT (2'S COMPLEMENT)
001402' 000000  RKBA:   0                               ;BUS ADDR (LOW 16 BITS)
001404' 000000  RKDA:   0                               ;TRACK AND SECTOR ADDR
001406' 000000  RKCS2:  0                               ;CONTR AND STATUS REG 2
001410' 000000  RKDS:   0                               ;DRIVE STATUS REG
001412' 000000  RKER:   0                               ;ERROR REG
001414' 000000  RKASOF: 0                               ;ATTENTION REG
001416' 000000  RKDC:   0                               ;CYLINDER ADDR
001420' 000000  RKECPS: 0                               ;ECC BAD BIT LOCATION
001422' 000000  RKECPT: 0                               ;ECC PATTREN
```

```
001424' 000354'                 BLK:    BLK1
001426' 177777                  177777

                053057          ERHARD=BIT0!BIT1!BIT2!BIT3!BIT5!BIT9!BIT10!BIT12!BIT14
                036000          S1HARD=BIT10!BIT11!BIT12!BIT13
                137400          S2HARD=BIT8!BIT9!BIT10!BIT11!BIT12!BIT13!BIT15
                004070          DSHARD=BIT3!BIT4!BIT5!BIT11

001430' 000016                  ERBUF:  .BLKW   16
001464' 000000                  CNTR1:  0
001466' 000000                  CNTR2:  0

001470' 007404'                 DRP:    DRVE
001472' 001504'                         NUMB
001474' 007414'                         DRPED
001476' 000000                  DRPMSG: 0
001500' 177777                          -1
001502' 000000                  HSRD:   0
001504' 000000                  NUMB:   .WORD   0
001506' 000000                  RETRY:  0
```



```
001510' 001542'                 TABLE:  SVCS1
001512' 001544'                         SVWC
001514' 001546'                         SVBA
001516' 001550'                         SVDA
001520' 001552'                         SVCS2
001522' 001554'                         SVDS
001524' 001556'                         SVER
001526' 001560'                         SVASOF
001530' 001562'                         SVDC
001532' 001564'                         SVECPS
001534' 001566'                         SVECPT
001536' 000354'                         BLK1
001540' 177777                          177777                  ;END OF TABLE INDICATOR

001542' 000000                  SVCS1:  0                       ;SAVE OF CONTROLLER STATUS REGISTER
001544' 000000                  SVWC:   0                       ;SAVE OF WORD COUNT (2'S COMPLEMENT)
001546' 000000                  SVBA:   0                       ;SAVE OF BUS ADDR (LOW 16 BITS)
001550' 000000                  SVDA:   0                       ;SAVE OF TRACK AND SECTOR ADDR
001552' 000000                  SVCS2:  0                       ;SAVE OF CONTR AND STATUS REG 2
001554' 000000                  SVDS:   0                       ;SAVE OF DRIVE STATUS REG
001556' 000000                  SVER:   0                       ;SAVE OF ERROR REG
001560' 000000                  SVASOF: 0                       ;SAVE OF ATTENTION REG
001562' 000000                  SVDC:   0                       ;SAVE OF CYLINDER ADDR
001564' 000000                  SVECPS: 0                       ;SAVE OF ECC BAD BIT LOCATION
001566' 000000                  SVECPT: 0                       ;SAVE OF ECC PATTREN
                                ;
                                ; HOUSEKEEPING COMMANDS
                                ;IF RK07 DRIVE IS SELECTED,THE BIT10 OF EACH COMMAND MUST BE SET
                                ;IF RK06, MUST BE RESET
                                ;

001570' 000007                  UNLOAD: .WORD   7               ;UNLOAD COMMAND
001572' 000001                  SELDRV: .WORD   1               ;SELECT A DRIVE
001574' 000011                  SPINDL: .WORD   11              ;STARTS THE SPINDLE
001576' 000005                  CLRDRV: .WORD   5               ;CLEAR THE DRIVE
001600' 100100                  DISINT: .WORD   #BIT15!BIT6     ;CLEAR CCER ERROR WITH INTERRUPT DISABLE
001602' 100000                  CLRCTR: .WORD   #BIT15          ;CLEAR CONTROLLER ERROR
001604' 000013                  RECALX: .WORD   13              ;RECALIBRATE
001606' 000003                  PACKAC: .WORD   3               ;PACK ACKNOWLEDGE
001610' 000000                  RKFLG:  .WORD   0               ;SOFTWARE RK06,RK07 FLAG
                                                                ;BIT10=1 IF RK07,RESET IF RK06
```

```
                                .SPTTL  MAIN LOOP
001612' 012767 006000 176276 START:  MOV     #3072.,WDFR
001620' 012767 003000 176266         MOV     #1536.,WDTO
001626' 012767 000010 176264         MOV     #8.,INTR
001634' 012767 000001 176512         MOV     #1,BLK1         ;START AT BLOCK(SECTOR) 1
001642' 012767 000001 176442         MOV     #1,FLAG         ;SET 1ST TIME FLAG BIT
001650' 004767 000572                JSR     PC,SETUP
001654' 004767 001040                JSR     PC,CHKDRV       ;CHECK ALL DRIVES
001660'                         RESTRT:
001660' 012706 000252'               MOV     #MODSP,SP       ;RESTORE STACK
001664' 104415 000000' 000124'       GETPAS,BEGIN,,RBUFVA    ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
001672' 016767 176234 176464         MOV     RBUFSZ,WCNT2    ;SIZE OF READ BUFFER
001700' 005467 176460                NEG     WCNT2
001704' 005067 176426                CLR     CNT             ;RESET END-OF-PASS COUNTER
001710' 005567 177460                CLR     INTFLG          ;CLEAR EXPECTED INTERRUPT FLAG
001714' 012767 177777 176430 LOOP1:  MOV     #-1,DRVVE       ;INITIALIZE DRIVE COUNT
001722' 012767 177777 176414         MOV     #-1,DSKADR      ;INITIALIZE DRIVE ADDRESS
001730' 004767 001412                JSR     PC,PICKRK       ;GO PICK A BLK1
001734' 104414 000000'               GWBUFS,BEGIN            ;GET WRITE BUFFER INFORMATION
001740' 016767 176176 176414         MOV     WRUFSZ,WCNT1    ;SAVE WRITE BUFFER SIZE
001746' 005467 176410                NEG     WCNT1
001752'                         LOOP2:
001752' 004767 001260                JSP     PC,NXTDRV       ;GO PICK A DRIVE
001756' 000403                       BR      LOOP4           ;RETURNS HERE IF ALL DRIVES DONE
001760' 004767 000042           LOOP3:  JSR     PC,CYCLE        ;GO DO A CYCLE ON THIS DRIVE
001764' 000772                       BP      LOOP2           ;DO IT TO NEXT DRIVE
                                
001766' 042767 000001 176316 LOOP4:  BIC     #1,FLAG         ;CLEAR FIRST TIME FLAG
001774' 005767 176346                TST     DVICE           ;ANYBODY LEFT TO CHECK?
002000' 001002                       BNE     2$              ;BR IF YES
002002' 000167 000006                JMP     ENTEST          ;ELSE DROP MODULE
002006'                         2$:
002006' 104413 000000'               ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
                                                             ;MONITOR SHALL TEST END OF PASS
002012' 000740                       BR      LOOP1           ;BR BACK IF NO

002014'                         ENTEST:
002014' 104403 000000' 001476'       MSGNS,BEGIN,DRPMSG      ;ASCII MESSAGE CALL WITH COMMON HEADER
002022' 104410 000000'               ENDS,BEGIN              ;
```



```
002026' 032767 000020 175762 CYCLE:  BIT     #20,SR1         ;B-PORT?
002034' 001076                       BNE     CYCLEB          ;BR IF SO, ELSE DO A-PORT
002036' 052767 000004 176246         BIS     #4,FLAG         ;SET THE A-PORT FLAG
002044' 004767 004654                JSR     PC,BLKADR       ;CONVERT BLK TO DISK ADDR
002050' 004767 001374                JSP     PC,WRITE        ;GO WRITE A BLOCK
002054' 004767 001426                JSR     PC,WRITCK       ;GO DO WRITE CHECK
002060' 004767 001436                JSR     PC,READ         ;GO READ A BLOCK
002064' 104412 000000' 000126'       CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
002072' 002074'                      .+2                     ; IF ERROR, CONTINUE
002074' 032767 000200 175714         BIT     #BIT7,SR1       ;SINGLE PORT?
002102' 001452                       BEQ     5$              ;YES-GET OUT

                                ;NOW UPDATE BLOCK 0 FOR B-PORT'S INFORMATION

002104' 004767 004730                JSP     PC,CLPRB        ;CLEAR THE READ BUFFER
002110' 016767 176176 176250         MOV     FLAG,RBUF       ;PUT INFO IN FIRST WORD
002116' 016767 176232 176244         MOV     BLK1,RBUF+2     ;PUT CURRENT ADDR IN NXT
002124' 004767 001524                JSP     PC,WRT00        ;GO UPDATE BLOCK 0
002130' 012767 050000 176164         MOV     #50000,TIMER    ;INIT WAIT LOOP
002136' 004767 004716           1$:  JSR     PC,RELEASE      ;GIVE THE DRIVE TO B
002142' 005367 176154           2$:  DEC     TIMER
002146' 001004                       BNE     3$
002150' 004767 004434                JSR     PC,DROP
002154' 007765'                      PORTHG
002156' 000207                       RTS     PC
002160' 012700 000002           3$:  MOV     #2,R0
002164'                         4$:
002164' 104407 000000'               BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002170' 104407 000000'               BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002174' 005300                       DEC     R0
002176' 001372                       BNE     4$
002200' 004767 001516                JSR     PC,RD00         ;GO READ BLOCK 0
002204' 032767 000002 176154         BIT     #2,RBUF         ;HAS B UPDATED BLK 0 YET?
002212' 001751                       BEQ     1$              ;BR BACK IF NOT
002214' 004767 001302                JSR     PC,READ         ;GO READ THE DATA B WROTE
002220' 104412 000000' 000126'       CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
002226' 002230'                      .+2                     ; IF ERROR, CONTINUE
002230' 000207                  5$:  RTS     PC
```

```
                                ;
                                ;
002232' 012767  060000  176062  CYCLER: MOV     #60000,TIMER
002240' 004767  001456          1$:     JSR     PC,RDOO         ;GO SEE IF A IS DONE YET
002244' 032767  000001  176040          BIT     #1,FLAG         ;IS THIS THE FIRST TIME SINCE START?
002252' 001410                          BEQ     2$              ;BR IF NO, SKIP THIS CHECK
002254' 032767  000004  176104          BIT     #4,RBUF         ;HAS A WRITTEN THIS BLOCK?
002262' 001410                          BEQ     3$              ;BR IF NO
002264' 032767  000001  176074          BIT     #1,RBUF         ;HAS A SET THE FIRST TIME FLAG?
002272' 001404                          BEQ     3$              ;BR IF NOT, MUST WAIT
002274' 032767  000002  176064  2$:     BIT     #2,RBUF         ;HAS A WRITTEN THIS BLOCK SINCE B DID?
002302' 001421                          BEQ     6$              ;BR IF SO, ELSE
002304' 004767  004550          3$:     JSR     PC,RELEASE      ;GIVE PORT TO A
002310' 012700  000002                  MOV     #2,R0
002314'                         4$:
002314' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002320' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002324' 005300                          DEC     R0
002326' 001372                          BNE     4$
002330' 005367  175766                  DEC     TIMER
002334' 001341                          BNE     1$
002336' 004767  004246                  JSR     PC,DROP
002342' 007765'                         PORTHG
002344' 000207                          RTS     PC
002346' 016767  176016  176000  6$:     MOV     RBUF+2,BLK1     ;GET THE CURRENT BLK1
002354' 004767  004344                  JSR     PC,BLKADR       ;GENERATE DISK ADDR FROM IT
002360' 004767  001136                  JSR     PC,READ         ;GO READ WHAT A WROTE
002364' 004767  001170                  JSR     PC,WRITEB       ;GO WRITE IT BACK OUT
002370' 004767  001222                  JSR     PC,WRITCB       ;GO WRITE CHECK IT
002374' 004767  004440                  JSR     PC,CLPRB        ;GO CLEAR BUFFER
002400' 052767  000002  175704          BIS     #2,FLAG         ;SET BIT SAYING B'S DONE
002406' 016767  175700  175752          MOV     FLAG,RBUF       ;PUT INFO INTO RBUF
002414' 004767  001234                  JSR     PC,WRT00        ;GO WRITE IT FOR A TO SEE
002420' 004767  004334                  JSR     PC,RELEASE      ;GIVE DRIVE TO A
002424' 012700  000002                  MOV     #2,R0
002430'                         7$:
002430' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002434' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002440' 005300                          DEC     R0
002442' 001372                          BNE     7$
002444' 000207                          RTS     PC
```



```
                                        .SBTTL  INITIALIZATION
                                ;
                                ;
002446' 016767  175342  175672  SETUP:  MOV     DVID1,DVICE     ;GET DRIVE INDICATOR
002454' 122737  000013  000041          CMPB    #13,@#41        ;IF RK IS LOAD MEDIUM THEN
002462' 001024                          BNE     3$              ;BEGIN
002464' 113700  000040                  MOVB    @#40,R0         ;   GET LOAD-DEVICE NUMBER
002470' 012701  000001                  MOV     #1,R1           ;   INITIALIZE DEVICE MASK
002474' 105700                  10$:    TSTB    R0              ;   WHILE NOT POINTING AT LOAD DEVICE DO
002476' 001403                          BEQ     20$             ;   BEGIN
002500' 006301                          ASL     R1              ;      POINT AT NEXT DEVICE
002502' 105300                          DECB    R0              ;      COUNT SHIFTS
002504' 000773                          BR      10$             ;   END
002506' 130167  175634          20$:    BITB    R1,DVICE        ;   IF LOAD-DEVICE IS SELECTED THEN
002512' 001410                          BEQ     3$              ;   BEGIN
002514' 140167  175626                  BICB    R1,DVICE        ;      DROP LOAD-DEVICE
002520' 012767  007426'  176750          MOV     #LOAD,DRPMSG    ;      SPECIFY LOAD-DEVICE DROP
002526' 104403  000000'  001476'        MSGN$,BEGIN,DRPMSG      ;ASCII MESSAGE CALL WITH COMMON HEADER
                                ;   END
002534'                         3$:                             ;END
                                ;
                                ;
                                ;SET UP DEVICE REGISTERS - USE ADDRESS SPECIFIED BY ADDR IN HEADER
                                ;
002534' 016700  175246          1$:     MOV     ADDR,R0         ;GET DEVICE ADDRESS
002540' 010067  176632                  MOV     R0,RKCS1        ;GENERATE CONTROLLER REG ADDRESSES
002544' 005720                          TST     (R0)+
002546' 010067  176626                  MOV     R0,RKWC
002552' 005720                          TST     (R0)+
002554' 010067  176622                  MOV     R0,RKBA
002560' 005720                          TST     (R0)+
002562' 010067  176616                  MOV     R0,RKDA
002566' 005720                          TST     (R0)+
002570' 010067  176612                  MOV     R0,RKCS2
002574' 005720                          TST     (R0)+
002576' 010067  176606                  MOV     R0,RKDS
002602' 005720                          TST     (R0)+
002604' 010067  176602                  MOV     R0,RKER
002610' 005720                          TST     (R0)+
002612' 010067  176576                  MOV     R0,RKASOF
002616' 005720                          TST     (R0)+
002620' 010067  176572                  MOV     R0,RKDC
002624' 022020                          CMP     (R0)+,(R0)+     ;SKIP UNUSED REGISTER
002626' 022020                          CMP     (R0)+,(R0)+     ;SKIP RKDB & RKMR1
002630' 010067  176564                  MOV     R0,RKECPS
002634' 005720                          TST     (R0)+
002636' 010067  176560                  MOV     R0,RKECPT
002642' 016767  175500  175500          MOV     DVICE,DRIVE     ;SAVE IT IS DRIVE
002650' 012767  177777  175474          MOV     #-1,DPYVE       ;INITIALIZE DRIVE COUNT
002656' 012767  177777  175460          MOV     #-1,DSKADR      ;INITIALIZE SHIFTED DRIVE NUMBER
002664' 016700  175120                  MOV     VECTOR,R0       ;GET VECTOR ADDRESS
002670' 012720  004062'                 MOV     #INTSPV,(P0)+   ;GET VECTOR ADDRESS
002674' 116710  175112                  MOVB    BR1,(R0)        ;SET PRIORITY
002700' 032767  000200  175110          BIT     #BIT7,SR1       ;DUAL PORT?
002706' 001403                          BEQ     2$              ;NO- BRANCH
002710' 016767  175444  175440          MOV     DPICNT,ICNT     ;YES, LOAD DUAL PORT ITERATION COUNT
```

```
   620  002716' 000207                  2$:     RTS     PC
```



```
   621                                     ;CHECK SELECTED DRIVES TO SEE IF READY
   622                                     ;IF NOT READY AFTER DELAY, DROP THE DRIVE
   623  002720' 012777  000040  176460     CHKDRV: MOV     #BIT5,@RKCS2    ;SUB-SYSTEM CLEAR
   624  002726' 004767  002572             JSR     PC,CNTRDY       ;WAIT FOR CONTROLLER READY
   625  002732' 005777  176440             TST     @RKCS1          ;ERROR?
   626  002736' 100013                     BPL     1$              ;NO-CONTINUE
   627  002740' 104403  000000' 007725'    MSGNS,BEGIN,UNCLR       ;ASCII MESSAGE CALL WITH COMMON HEADER
   628  002746' 012767  000047  175132     MOV     #47,ERRTYP      ;WONT CLEAR
   629                                     ;*****************************************************************
   630  002754' 104405  000000' 001376'    ERRORS,BEGIN,RKCS1      ;
   631                                     ;*****************************************************************
   632  002762' 000167  177026             JMP     ENTEST          ;GO DROP MODULE
   633  002766' 004767  003472     1$:     JSR     PC,DRVADR       ;GET A DRIVE ADDRESS
   634  002772' 000520                     BR      9$              ;DID ALL DRIVES
   635  002774' 004767  002572             JSR     PC,READY        ;CHECK OUT RK06 OR RK07
   636  003000' 000240                     NOP                     ;DUMMY RETURN ADDRESS
   637                                                             ;THE ABOVE LINE ADDED 5-OCT-77
   638  003002' 004767  003246             JSR     PC,RDY1         ;DRIVE AVAILABLE
   639  003006' 000167  177754             JMP     1$
   640  003012' 016777  175326  176366  2$:MOV     DSKADR,@RKCS2   ;NO, LOAD NEXT DRIVE ADDRESS
   641                                     ;
   642  003020' 016777  176562  176350     MOV     #3,@RKCS1       ;DRIVE SELECT / PACK ACKNOWLEDGE
   643  003026' 004767  002472             MOV     PACKAC,@RKCS1   ;PACK ACKNOWLEDGE
   644  003032' 005777  176340             JSR     PC,CNTRDY
   645  003036' 100472                     TST     @RKCS1          ;ERROR?
   646  003040' 032777  100000  176342     BMI     8$              ;YES-DROP DRIVE
   647  003046' 001466                     BIT     #BIT15,@RKDS    ;STATUS VALID
   648  003050' 004767  003200             BEQ     8$
   649  003054' 000167  177706             JSR     PC,RDY1
   650  003060' 016777  175260  176320 10$:JMP     1$
   651                                     MOV     DSKADR,@RKCS2
   652  003066' 016777  176502  176302     ;
   653  003074' 005067  176266             MOV     #11,@RKCS1      ;START SPINDLE
   654  003100' 004767  002420             MOV     SPINDL,@RKCS1   ;START THE SPINDLE
   655  003104' 005777  176266             CLR     CLK             ;CLEAR TIMER
   656  003110' 100427                     JSP     PC,CNTRDY       ;WAIT FOR CONTROLLER READY
   657  003112' 032777  100000  176270     TST     @RKCS1          ;ERROR?
   658  003120' 001010                     BMI     5$              ;YES-DROP DRIVE
   659  003122' 104407  000000'            BIT     #BIT15,@RKDS    ;ATTN SET-OPERATION COMPLETE?
   660  003126' 104407  000000'            BNE     20$             ;YES-CONTINUE
   661  003132' 005367  176230             BREAKS,BEGIN            ;TEMPORARY RETURN TO MONITOR....
   662  003136' 100765                     BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
   663  003140' 000421                     DEC     CLK             ;TIME LET?
   664  003142' 004767  003106             BMI     3$              ;YES-TRY AGAIN
   665  003146' 000167  177614             BR      6$              ;NO
   666  003152'                    20$:     JSR     PC,RDY1
   667                                     JMP     1$
   668  003152' 016777  176420  176216 4$: ;
   669  003160' 004767  002172             MOV     #5,@RKCS1       ;CLEAR ATTN BITS
   670  003164' 000407                     MOV     CLRDRV,@RKCS1   ;DRIVE CLEAR
   671  003166' 000412                     JSR     PC,DRVRDY       ;DRIVE READY
   672  003170' 012767  007747' 176300     BR      6$              ;NO-ERROR
   673  003176' 104403  000000' 001476' 5$:MOV     #DRVERR,ORPMSG  ;SET UP ERROR MESSAGE
   674                                     MSGNS,BEGIN,ORPMSG      ;ASCII MESSAGE CALL WITH COMMON HEADER
   675  003204' 004767  003400     6$:     JSR     PC,DROP         ;DROP DRIVE
   676  003210' 010031'                    SPIND
```

```
677  003212' 000665                    7$:     BR      1$
678  003214'
679  003214' 004767  003056                    JSR     PC,RECAL        ;TRY TO RECALIBRATE THE DRIVE
680  003220' 000240                            NOP
681  003222' 000661                            BR      1$              ;DO NEXT DRIVE
682  003224' 004767  003360            8$:     JSR     PC,DROP
683  003230' 007502'                           UNAVAL
684  003232' 000655                            BR      1$
685  003234' 000207                    9$:     RTS     PC              ;RETURN
```



```
686
687
688
689
690  003236' 004767  003222            NXTDRV: JSR     PC,DRVADR       ;GET A DRIVE ADDRESS
691  003242' 000440                            BR      5$              ;BAD RETURN, NO DRIVE
692  003244' 004767  003004                    JSR     PC,RDY1
693  003250' 000167  177762                    JMP     NXTDRV
694  003254' 016777  175064  176124    1$:     MOV     DSKADR,@RKCS2   ;LOAD DISK ADDRESS
695                                     ;       MOV     #BIT0,@RKCS1    ;SELECT DRIVE
696  003262' 016777  176304  176106             MOV     SELDRV,@RKCS1   ;SELECT DRIVE
697  003270' 004767  002230                    JSR     PC,CNTRDY       ;WAIT FOR CONTROLLER READY
698  003274' 005777  176076                    TST     @RKCS1          ;ERROR?
699  003300' 100413                            BMI     3$              ;YES-GO DROP DRIVE
700  003302' 004767  002050                    JSR     PC,DRVRDY       ;WAIT FOR DRIVE TO FINISH
701  003306' 000410                            BR      3$
702  003310' 032777  004000  176072    2$:     BIT     #BIT11,@RKDS    ;WRITE PROTECTED ?
703  003316' 001410                            BEQ     4$
704  003320' 004767  003264                    JSR     PC,DROP         ;YES, DROP THE DRIVE
705  003324' 007464'                           PROT
706  003326' 000743                            BR      NXTDRV          ;GO ON TO NEXT DRIVE
707  003330' 004767  003254            3$:     JSR     PC,DROP         ;GO DROP DRIVE
708  003334' 007747'                           DRVERR
709  003336' 000737                            BR      NXTDRV          ;GET NEXT DRIVE
710
711  003340' 062716  000002            4$:     ADD     #2,(SP)         ;GOOD RETURN
712  003344' 000207                    5$:     RTS     PC
713
```

```
                                                ;DO NEXT BLOCK(SECTOR)
                                                ;DO RANDOM SEEKS?
                                                ;NO-CONTINUE
                                                ;GET RANDOM NUMBER
                                                ;BLOCK 0 SELECTED?
                                                ;YES-GET ANOTHER BLOCK
                                                ;OUT OF BOUNDS
                                                ;NO-CONTINUE
                                                ;GO BACK TO BLOCK 1
                                                ;TRY AGAIN
                                                ;GET BAD SPOT TABLE
                                                ;LOOK FOR 16 ENTRIES
                                                ;IS THIS A BAD BLK1?
                                                ;IF YES, GO PICK A NEW ONE
                                                ;COUNT A TABLE LOOK-UP
                                                ;BR BACK IF MORE TO GO
```



```
                                                ;LOAD WRITE FUNCTION
                                                ;LOAD WORD COUNT
                                                ;LOAD BUFFER ADDRESS
                                                ;LOAD EXTENDED MEMORY BITS
                                                ;LOAD WRITE-CHECK FUNCTION
                                                ;LOAD READ FUNCTION
                                                ;LOAD WORD COUNT
                                                ;LOAD BUFFER ADDRESS
                                                ;LOAD EXTENDED MEMORY BITS
                                                ;LOAD WRITE FUNCTION
                                                ;LOAD WORD COUNT
                                                ;LOAD BUFFER ADDRESS
                                                ;LOAD EXTENDED MEMORY BITS
                                                ;LOAD WRITE CHECK FUNCTION
                                                ;LOAD WORD COUNT
                                                ;LOAD BUFFER ADDRESS
                                                ;LOAD EXTENDED MEMORY BITS
                                                ;LOAD WRITE FUNCTION
                                                ;LOAD WORD COUNT
                                                ;LOAD BUFFER ADDRESS
                                                ;LOAD EXTENDED MEMORY BITS
                                                ;DO CYL 0
                                                ;SECTOR 0
                                                ;LOAD READ COMMAND
```

```
                                        .SBTTL  DEVICE DRIVER & INTERRUPT SERVICE

003732' 016767  174366  174356  EXCUTE: MOV     CYLADR,TCYL     ;GET CYL ADDR
003740' 016767  174362  174352          MOV     SECADR,TSEC     ;GET SECTOR ADDRESS
003746' 012767  000003  175532  EXCUT1: MOV     #3,RETRY
003754' 005067  175414          EXCUT2: CLR     INTFLG          ;CLEAR EXPECTED INTERRUPT FLAG
003760' 004767  002270                  JSR     PC,RDY1
003764' 000167  175670                  JMP     RESTRT
003770' 016777  174340  175404  1$:     MOV     BUFADR,@RKBA
003776' 016777  174326  175374          MOV     WDCNT,@RKWC
004004' 016777  174306  175404          MOV     TCYL,@RKDC      ;LOAD CYLINDER ADDR
004012' 016777  174302  175364          MOV     TSEC,@RKDA      ;LOAD THE ADDR OF SECTOR
004020' 016700  174316                  MOV     XMEM,R0         ;EXTENDED MEM BITS
004024' 006300                          ASL     R0
004026' 006300                          ASL     R0
004030' 006300                          ASL     R0
004032' 006300                          ASL     R0
004034' 056700  174300                  BIS     FUNC,R0         ;ADD FUNCTION CODE
004040' 056700  175544                  BIS     RKFLG,R0        ;SET/RESET THE BIT10
004044' 010077  175326                  MOV     R0,@RKCS1       ;EXECUTE THE FUNCTION
004050' 012767  000001  175316          MOV     #1,INTFLG       ;TURN ON EXPECTED INTERUPT FLAG
004056' 104400  000000'                 EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```



```
004062' 042777  100100  175306  INTSRV: BIC     #BIT15!BIT6,@RKCS1      ;DISABLE INTERRUPTS
                                        ;-------------------------------------------------------------
004070' 000004  000000' 004076'         PIRQS,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-------------------------------------------------------------

004076' 005767  175272          1$:     TST     INTFLG          ;ARE WE REALLY EXPECTING AN INTERRUPT?
004102' 001404                          BEQ     2$              ;NO-MUST BE ERROR CONDITION
004104' 032777  140000  175264          BIT     #BIT15!BIT14,@RKCS1     ;ERROR OR DRIVE INTERRUPT
004112' 001523                          BEQ     6$              ;NO-MUST BE OKAY

004114' 012700  000012          2$:     MOV     #12,R0          ;SAVE 11. REGISTERS
004120' 016701  175252                  MOV     RKCS1,R1        ;START OF SAVE REGISTERS
004124' 012702  001542'                 MOV     #SVCS1,R2       ;START OF MEMORY TO PUT THEM IN
004130' 012122                  3$:     MOV     (R1)+,(R2)+
004132' 005300                          DEC     R0
004134' 003375                          BGT     3$
004136' 016767  175234  173734          MOV     RKCS1,CSRA      ;SET UP CSR FOR ERROR CALL
004144' 016767  175372  173730          MOV     SVCS1,ACSR      ;SET UP CONTENTS OF SAME
004152' 012777  100000  175216          MOV     #BIT15,@RKCS1   ;CLEAR THE ERROR
004160' 004767  001340                  JSR     PC,CNTRDY       ;CONTROLLER READY
004164' 005777  175206                  TST     @RKCS1          ;ERROR?
004170' 100005                          BPL     4$              ;NO-CONTINUE
004172' 004767  002412                  JSR     PC,DROP         ;YES-DROP DRIVE
004176' 007725'                         UNCLR
004200' 000167  175454                  JMP     RESTRT          ;START OVER
004204' 005767  175332          4$:     TST     SVCS1           ;ERROR?
004210' 100446                          BMI     5$              ;REPORT THE ERROR
004212' 012767  007646' 175256          MOV     #FXTRA,DRPMSG
004220' 104403  000000' 001476'         MSGNS,BEGIN,DRPMSG      ;ASCII MESSAGE CALL WITH COMMON HEADER
004226' 012767  000011  173652          MOV     #11,ERRTYP      ;ILLEGAL INTERRUPT
                                        ;*************************************************************
004234' 104405  000000' 001510'         ERDERS,BEGIN,TABLE      ;
                                        ;*************************************************************
004242' 012777  000040  175136          MOV     #BIT5,@RKCS2    ;SUBSYSTEM CLEAR
004250' 004767  001250                  JSR     PC,CNTRDY       ;WAIT FOR CONTROLLER READY
004254' 005777  175116                  TST     @RKCS1          ;ERROR?
004260' 100404                          BMI     9$              ;YES GO DROP TEST
004262' 012706  000252'                 MOV     #MODSP,SP       ;INITIALIZE THE SP
004266' 000167  175466                  JMP     LOOP3           ;JMP TO LOOP3
004272' 012767  007725' 175176  8$:     MOV     #UNCLR,DRPMSG   ;YES-LOAD MESSAGE
004300' 104403  000000' 001476'         MSGNS,BEGIN,DRPMSG      ;ASCII MESSAGE CALL WITH COMMON HEADER
004306' 012767  000047  173572          MOV     #47,ERRTYP      ;WONT CLEAR
                                        ;*************************************************************
004314' 104405  000000' 001376'         ERDERS,BEGIN,RKCS1      ;
                                        ;*************************************************************
004322' 000167  175466                  JMP     ENTEST          ;GO END IT ALL

004326' 004767  000036          5$:     JSR     PC,ERRORS
004332' 000414                          BR      7$
004334' 005367  175146                  DEC     RETRY
004340' 002205                          BGE     EXCUT2
004342' 032767  000001  173446          BIT     #BIT0,SR1
004350' 001405                          BEQ     7$
004352' 004767  002232                  JSR     PC,DROP
004356' 001506'                         RETRY
004360' 000401                          BR      7$
```

```
  857
  858  004362' 000207                  6$:     RTS     PC
  859
  860  004364' 000167  175270          7$:     JMP     RESTRT
```



```
  861                                          .SPTTL  ERROR HANDLER
  862
  863  004370'                         ERRORS:
  864  004370' 004767  000154                  JSR     PC,ERRGEN       ;GENERATE THE ASCII MESSAGE
  865
  866  004374' 032767  000200  175154          BIT     #BIT7,SVER      ;BAD SECTOR ERROR
  867  004402' 001410                          BEQ     1$              ;SKIP THIS IF NOT
  868  004404' 004767  000362                  JSR     PC,BSRD         ;READ & COMAPRS BAD FILE SECTOR
  869  004410' 000432                          BR      2$              ;NO MATCH - REPORT AS AN ERROR
  870  004412' 032767  000004  173376          BIT     #BIT2,SR1       ;TYPE ERROR MESSAGE?
  871  004420' 001450                          BEQ     4$              ;NO JUST EXIT
  872  004422' 000425                          BR      2$              ;TYPE IT ALL
  873
  874  004424' 032767  100000  175124  1$:     BIT     #BIT15,SVER     ;DATA CHECK ERROR?
  875  004432' 001421                          BEQ     2$              ;NO
  876  004434' 032767  000100  175114          BIT     #BIT6,SVER      ;ECC HARD ERROR?
  877  004442' 001015                          BNE     2$              ;YES - GO REPORT IT
  878
  879  004444' 016700  175072                  MOV     SVCS1,R0
  880  004450' 042700  017741                  BIC     #17741,R0       ;KEEP THE FUNCTION CODE
  881  004454' 022700  000020                  CMP     #20,R0
  882  004460' 001006                          BNE     2$
  883
  884  004462' 004767  000554                  JSR     PC,ECCOR        ;APPLY THE CORRECTION STUF
  885  004466' 032767  000010  173322          BIT     #BIT3,SR1       ;TYPE A MESSAGE
  886  004474' 001422                          BEQ     4$              ;NO - JUST EXIT
  887
  888  004476' 026727  175004  000003  2$:     CMP     RETRY,#3
  889  004504' 002410                          BLT     3$
  890
  891  004506' 104403  000000' 001430'         MSGNS,BEGIN,ERBUF       ;ASCII MESSAGE CALL WITH COMMON HEADER
  892  004514' 005067  173366                  CLR     ERRTYP
  893                                          ;*****************************************************************
  894  004520' 104405  000000' 00151C'         ERDERS,BEGIN,TABLE      ;
  895                                          ;*****************************************************************
  896
  897
  898  004526' 005767  174640          3$:     TST     SETERR          ;WAS IT A SOFT ERROR?
  899  004532' 001003                          BNE     4$              ;YES
  900  004534' 004767  001536                  JSR     PC,RECAL        ;NO - HARD - RECALIBRATE THE DRIVE
  901  004540' 000402                          BR      5$
  902
  903  004542' 062716  000002          4$:     ADD     #2,(SP)
  904  004546' 000207                  5$:     RTS     PC
```

```
                                        .SBTTL  ERROR MESSAGE GENERATOR
004550' 005067  174616          ERRGEN: CLR     SFTERR          ;CLEAR SOFT ERROR FLAG
004554' 012700  007114'                 MOV     #ERTAP,R0       ;ERROR TABLE PARAMETERS
004560' 012705  001430'                 MOV     #ERBUF,R5       ;PRINT POINTER BUFFER
004564' 012715  007354'                 MOV     #HARD,(R5)      ;ASSUME THAT ITS HARD
004570' 032767  137400  174754          BIT     #S2HARD,SVCS2   ;ANY HARD ERRORS IN RKCS2?
004576' 001020                          BNE     1$              ;YES, REPORT
004600' 032767  036000  174734          BIT     #S1HARD,SVCS1   ;ANY HARD ERRORS IN RKCS1?
004606' 001014                          BNE     1$              ;YES REPORT
004610' 032767  004070  174736          BIT     #DSHARD,SVDS    ;ANY HARD ERRORS IN RKDS?
004616' 001010                          BNE     1$              ;REPORT
004620' 032767  053057  174730          BIT     #ERHARD,SVER    ;ANY HARD ERRORS IN RKER
004626' 001004                          BNE     1$
004630' 012715  007370'                 MOV     #SOFT,(R5)      ;MUST BE SOFT BY NOW
004634' 005267  174532                  INC     SFTERR          ;SET SOFT ERROR FLAG
004640' 005725                  1$:     TST     (R5)+
004642' 012767  000004  174614          MOV     #4,CNTR1        ;COUNTER FOR 4 REGISTERS
004650' 012700  007114'                 MOV     #ERTAP,R0       ;START OF ERROR TABLE PARAMETERS
004654' 012001                  2$:     MOV     (R0)+,R1        ;GET ADDRESS OF POINTER TO REGISTER
004656' 012004                          MOV     (R0)+,R4        ;ADDRESS OF ERROR MESSAGE TABLE
004660' 012003                          MOV     (R0)+,R3        ;GET MASK
004662' 011102                          MOV     (R1),R2         ;GET CONTENTS OF REGISTER
004664' 040302                          BIC     R3,R2           ;KEEP ONLY THOSE IN BIT MASK
004666' 001424                          BEQ     6$              ;GO TO NEXT REGISTER IF ZERO
004670' 005103                          COM     R3
004672' 012767  000020  174566          MOV     #16.,CNTR2

004700' 000241                  3$:     CLC                     ;CLEAR THE CARRY
004702' 006003                          ROR     R3              ;ROTATE THE MASK WORD
004704' 103402                          BCS     4$
004706' 006002                          ROR     R2              ;ROTATE THE VALUE - DUMMY ROTATE
004710' 000413                          BR      6$

004712' 000241                  4$:     CLC
004714' 006002                          ROR     R2              ;IS BIT SET?
004716' 103006                          BCC     5$              ;NO
004720' 012725  007350'                 MOV     #BLNKS,(R5)+    ;TWO SPACES
004724' 010425                          MOV     R4,(R5)+        ;PUT ADDRESS OF MESSAGE IN TABLE
004726' 020527  001454'                 CMP     R5,#ERBUF+24    ;END OF BUFFER SPACE
004732' 002010                          BGE     7$              ;YES - GO PRINT THEM

004734' 062704  000004          5$:     ADD     #4,R4           ;ADDRESS OF NEXT MESSAGE
004740' 005367  174522          6$:     DEC     CNTR2           ;DONE WITH THIS WORD?
004744' 001355                          BNE     3$

004746' 005367  174512                  DEC     CNTR1           ;DONE WITH ALL REGISTERS?
004752' 001340                          BNE     2$

004754' 012725  007404'          7$:    MOV     #DRVE,(R5)+
004760' 012725  001504'                 MOV     #NUMB,(R5)+
004764' 012715  177777                  MOV     #-1,(R5)        ;END OF MESSAGES
004770' 000207                          RTS     PC
```



```
                                        .SBTTL  BAD SECTOR CHECK

004772' 012767  000001  174502  BSRD:   MOV     #1,HSRD         ;FLAG TO INDICATE READING HARDWARE SECTORS
005000' 012700  000776                  MOV     #776,R0         ;INITIAL SECTOR COUNT

005004' 062700  000002          1$:     ADD     #2,R0
005010' 020027  001012                  CMP     R0,#1012        ;DONE WITH HARDWARE FILE?
005014' 001002                          BNE     2$
005016' 005067  174460                  CLR     HSRD            ;DOING SOFTWARE FROM NOW ON

005022' 020027  001026          2$:     CMP     R0,#1026        ;END OF SOFTWARE FILE?
005026' 002104                          BGE     6$              ;END OF FILE AND NO MATCH
005030' 004767  001220                  JSR     PC,RDV1
005034' 000167  174620                  JMP     RESTRT

005040' 016777  173300  174340  10$:    MOV     DSKADR,@RKCS2   ;LOAD DISK ADDRESS
005046' 012777  177400  174324          MOV     #-400,@RKWC     ;ONE SECTOR TRANSFER
005054' 016777  173046  174320          MOV     RBUFPA,@RKBA    ;BUFFER ADDRESS
005062' 012777  000632  174326          MOV     #410,@RKDC      ;DESIRED CYLINDER
005070' 005767  174514                  TST     RKFLG           ;BRANCH IF IT IS A RK06
005074' 001403                          BEQ     .+8.            ;SKIP THE NEXT MOV INSTRUCTION
005076' 012777  001456  174312          MOV     #1456,@RKDC     ;LAST CYLINDER FOR RK07
005104' 010077  174274                  MOV     R0,@RKDA        ;DISK SECTOR/TRACK ADDRESS
005110' 016705  173014                  MOV     RBUFEA,R5       ;GET READ BUFF EA BITS
005114' 006305                          ASL     R5              ;ALIGN BITS
005116' 006305                          ASL     R5
005120' 006305                          ASL     R5
005122' 006305                          ASL     R5
005124' 052705  000021                  BIS     #21,R5          ;LOADFUNCTION
005130' 056705  174454                  BIS     RKFLG,R5        ;SET THE RK06/RK07 FLAG
005134' 010577  174236                  MOV     R5,@RKCS1       ;ISSUE THE FUNCTION
005140' 004767  000212                  JSR     PC,DRVRDY       ;WAIT FOR DRIVE READY
005144' 000435                          BR      6$
005146' 005777  174224                  TST     @RKCS1          ;ANY ERRORS?
005152' 100714                          BMI     1$              ;YES IF SET

005154' 012701  000177                  MOV     #177,R1         ;MAXINUM NUMBER OF BAD SECTOR ENTRIES
005160' 012702  000376'                 MOV     #RBUF+10,R2     ;START OF BAD SECTOR INFO
005164' 005301                  3$:     DEC     R1
005166' 001414                          BEQ     4$              ;THIS IS THE END

005170' 012203                          MOV     (R2)+,R3        ;CYLINDER INFOR
005172' 012204                          MOV     (R2)+,R4        ;SECTOR/TRACK INFO
005174' 020327  177777                  CMP     R3,#-1          ;END OF SECTOR INFO
005200' 001407                          BEQ     4$
005202' 026703  174354                  CMP     SVDC,R3         ;DESIRED CYLINDER?
005206' 001366                          BNE     3$              ;NO - KEEP LOOKING
005210' 026704  173112                  CMP     SECADR,R4       ;DESIRED SECTOR?
005214' 001363                          BNE     3$              ;NO - KEEP LOOKING
005216' 000406                          BR      5$              ;MATCH RETURN

005220' 005767  174256          4$:     TST     HSRD            ;DOING THE HARDWARE GUYS?
005224' 001403                          BEQ     5$              ;NO - ALL DONE THEN
005226' 012700  001010                  MOV     #1010,R0        ;SET UP FOR SOFTWARE FILE
005232' 000664                          BR      1$
```

```
1014  005234' 062716  000002          5$:     ADD     #2,(SP)
1015  005240' 000207                  6$:     RTS     PC
```



```
1016                                            .SBTTL  ERROR CORRECTION ROUTINE
1017
1018  005242' 016702  174300          ECCOR:  MOV     SVBA,R2         ;BEGINNING OF SECTOR ADDRESS
1019  005246' 162702  001000                  SUB     #1000,R2        ;START OF BAD DATA BUFFER
1020  005252' 016701  174306                  MOV     SVECPS,R1       ;POSITION OF ERROR
1021  005256' 005301                          DEC     R1
1022
1023  005260' 010100                          MOV     R1,R0
1024  005262' 006201                          ASR     R1
1025  005264' 006201                          ASR     R1              ;DETERMINE WORDS THAT ARE BAD
1026  005266' 006201                          ASR     R1
1027  005270' 042701  000001                  BIC     #1,R1           ;CLEAR BYTE INDICATOR
1028  005274' 060102                          ADD     R1,R2           ;ADDRESS OF ERROR
1029
1030  005276' 016703  174264                  MOV     SVECPT,R3       ;ERROR CORRECTION PATTERN
1031  005302' 005004                          CLR     R4
1032  005304' 042740  177760                  BIC     #177760,R0      ;BIT POSITION FOR START OF ECC CORRECTION
1033  005310' 001404                          BEQ     2$              ;CORRECTION STARTS ON WORD BOUNDRY
1034
1035  005312' 006303                  1$:     ASL     R3              ;SHIFT PATTERN LEFT ONE BIT
1036  005314' 006104                          ROL     R4
1037  005316' 005300                          DEC     R0
1038  005320' 001374                          BNE     1$              ;CHECK IF IN POSITION
1039
1040  005322' 011200                  2$:     MOV     (R2),R0         ;CORRECT 1ST WORD WITH EXCLUSIVE OR
1041  005324' 010301                          MOV     R3,R1
1042  005326' 040312                          BIC     R3,(R2)
1043  005330' 040001                          BIC     R0,R1
1044  005332' 050122                          BIS     R1,(R2)+
1045  005334' 020267  174206                  CMP     R2,SVBA
1046  005340' 001405                          BEQ     3$
1047
1048  005342' 011200                          MOV     (R2),R0         ;CORRECT SECOND WORD
1049  005344' 010401                          MOV     R4,R1
1050  005346' 040412                          BIC     R4,(R2)
1051  005350' 040001                          BIC     R0,R1
1052  005352' 050112                          BIS     R1,(R2)
1053
1054  005354' 000207                  3$:     RTS     PC              ;RETURN
```

```
                                        .SBTTL  COMMON SUPPORT ROUTINES
                                ;
005356' 005067  174004          DRVRDY: CLR     CLK
005362' 004767  000136                  JSR     PC,CNTRDY       ;CONTROLLER READY?
005366' 005777  174004                  TST     @RKCS1          ;ANY ERRORS?
005372' 100442                          BMI     3$              ;YES
005374' 004767  000654                  JSR     PC,RDY1
005400' 000167  000116                  JMP     6$
005404' 016777  172734  173774   1$:    MOV     DSKADR,@RKCS2
005412' 016777  174154  173756          MOV     SELDRV,@RKCS1   ;SELECT THE DRIVE
                                                                ;29-SEP-77
                                ;       MOV     #1,@RKCS1       ;SELECT DRIVE
005420' 004767  000100                  JSR     PC,CNTRDY
005424' 005777  173746                  TST     @RKCS1          ;ERROR?
005430' 100423                          BMI     3$              ;YES
005432' 032777  100000  173750   2$:    BIT     #BIT15,@RKDS    ;STATUS VALID
005440' 001417                          BEQ     3$
005442' 032777  000200  173740          BIT     #BIT7,@RKDS     ;DRIVE READY ?
005450' 001022                          BNE     5$              ;YES, RETURN
005452' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
005456' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
005462' 005367  173700                  DEC     CLK             ;NO, WAIT SOME MORE ?
005466' 100746                          BMI     1$              ;YES, WAIT
005470' 012767  007526' 174000          MOV     #NOTRED,DRPMSG
005476' 000403                          BR      4$
005500' 012767  007747' 173770   3$:    MOV     #DRVERR,DRPMSG
005506'                          4$:
005506' 104403  000000' 001476'         MSGN$,BEGIN,DRPMSG      ;ASCII MESSAGE CALL WITH COMMON HEADER
005514' 000402                          BR      6$
005516' 062716  000002           5$:    ADD     #2,(SP)         ;ADJUST STACK FOR GOOD RETURN
005522' 000207                   6$:    RTS     PC              ;RETURN
```



```
005524' 005067  173646          CNTRDY: CLR     CLK1
005530' 032777  000200  173640   1$:    BIT     #BIT7,@RKCS1    ;CONTROLLER READY ?
005536' 001014                          BNE     2$              ;YES, CONTINUE
005540' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
005544' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
005550' 005367  173614                  DEC     CLK1            ;WAIT SOME MORE ?
005554' 001365                          BNE     1$              ;YES
005556' 012767  007617' 173712          MOV     #CNTRNV,DRPMSG
005564' 000167  174224                  JMP     ENTEST          ;STOP THE RUN
005570' 000207                   2$:    RTS     PC              ;READY, RETURN
```

```
                                        ;ROUTINE DETERMINES IF DRIVE IS AVAILABLE, READY, VOLUME
                                        ;VALID, AND NOT WPITE PROTECTED

005572'                                 READY:
005572' 004767  000056                          JSR     PC,RKCHK        ;CHECK IT IS A RK06/RK07
005576' 000423                          BR      1$              ;ERROR EXIT FROM RKCHK
005600' 016777  172540  173600           MOV     DSKADR,@RKCS2   ;LOAD DRIVE NUMBER
                                        ;       MOV     #BIT0,@RKCS1    ;SELECT DRIVE
005606' 016777  173760  173562          MOV     SELDRV,@RKCS1   ;SELECT THE DRIVE
005614' 004767  177704                  JSR     PC,CNTRDY       ;WAIT FOR CONTROLLER READY
005620' 017701  173564                  MOV     @RKDS,R1        ;SAVE RKDS IN P1
005624' 032777  000001  173556          BIT     #BIT0,@RKDS     ;DRIVE AVAILABLE?
005632' 001405                          BEQ     1$              ;NO
005634' 005701                          TST     R1              ;DRIVE READY?
005636' 001403                          BEQ     1$              ;NO
005640' 032701  000001                  BIT     #BIT0,R1        ;VOLUME VALID?
005644' 001002                          BNE     2$              ;YES
005646' 062716  000002          1$:     ADD     #2,(SP)         ;SKIP INSRTUCTION FOLLOWING CALL
005652' 000207                  2$:     RTS     PC
                                        ;ROUTINE TO CHECK CURRENT SELECTED DRIVE IS A RK06 OR RK07
005654' 016777  172464  173524  RKCHK:  MOV     DSKADR,@RKCS2           ;LOAD THE DRIVE ADDRESS
005662' 012777  000001  173506          MOV     #BIT0,@RKCS1            ;ASSUME IT IS A RK06,SELECT DRIVE
005670' 004767  177630                  JSR     PC,CNTRDY               ;WAIT THE CONTROLLER READY
005674' 017701  173506                  MOV     @RKCS2,R1               ;READ THE STATUS
005700' 032701  010000                  BIT     #BIT12,R1               ;NED SET ?
005704' 001056                          BNE     7$                      ;BRANCH IF SO
005706' 017701  173476                  MOV     @RKDS,R1                ;READ THE DRIVE STATUS
005712' 032701  100000                  BIT     #BIT15,R1               ;DRIVE STATUS AVAILABLE ?
005716' 001446                          BEQ     6$                      ;BRANCH IF NOT
005720' 032701  000400                  BIT     #BIT8,R1                ;IS IT A RK06 ?
005724' 001420                          BEQ     3$                      ;BRANCH IF IT IS
005726' 017701  173460                  MOV     @RKER,R1                ;READ THE ERROR REGISTER
005732' 032701  000040                  BIT     #BIT5,R1                ;DRIVE TYPE ERROR MUST BE SET,IF RK07

005736' 001430                          BEQ     6$                      ;OTHERWISE ,ERROR
005740' 005001                  1$:     CLR     R1                      ;RK07 SET UP
005742' 012702  000011                  MOV     #9.,R2                  ;NINE COMMANDS
005746' 052761  002000  001570' 2$:     BIS     #BIT10,UNLOAD(R1)       ;SET RK07 BIT
005754' 005302                          DEC     R2                      ;
005756' 001416                          BEQ     5$                      ;BRANCH IF ALL DONE
005760' 062701  000002                  ADD     #2,P1                   ;ADJUST TABLE ADDRESS
005764' 000770                          BR      2$                      ;LOOPING BACK
005766' 005001                  3$:     CLR     R1                      ;RK06 SETUP
005770' 012702  000011                  MOV     #9.,R2                  ;NINE COMMANDS
005774' 042761  002000  001570' 4$:     BIC     #BIT10,UNLOAD(R1)       ;RESET RK07 FLAG
006002' 005302                          DEC     R2                      ;
006004' 001403                          BEQ     5$                      ;EXIT IF ALL DONE
006006' 062701  000002                  ADD     #2,P1                   ;ADJUST TABLE ADDRESS
006012' 000770                          BR      4$                      ;BRANCH BACK
006014' 062716  000002          5$:     ADD     #2,(SP)                 ;NORMAL EXIT
006020' 012777  100000  173350  6$:     MOV     #BIT15,@RKCS1           ;CLEAR THE CONTROLLER
006026' 000207                  7$:     RTS     PC                      ;EXIT
```



```
                                        ;ROUTINE WAITS FOR DRIVE READY, IF NOT READY

006030' 012767  077777  173330  NOTRDY: MOV     #77777,CLK      ;SET THE TIMER
006036' 012777  100000  173332  4$:     MOV     #BIT15,@RKCS1   ;CLEAR ERRORS FROM CU
006044' 004767  177454                  JSR     PC,CNTRDY       ; WAIT FOR CONTROLLER READY
006050' 016777  172270  173330          MOV     DSKADR,@RKCS2

                                        ;       MOV     #BIT0,@RKCS1    ;SELECT DRIVE
006056' 016777  173510  173312          MOV     SELDRV,@RKCS1   ;SELECT THE DRIVE
006064' 004767  177434                  JSR     PC,CNTRDY       ;WAIT FOR CU READY
006070' 032777  000001  173312          BIT     #BIT0,@RKDS     ;DRIVE AVAILABLE
006076' 001403                          BEQ     6$              ;NO
006100' 010146                          MOV     R1,-(SP)
006102' 012601                          MOV     (SP)+,R1
006104' 000414                          BR      2$
006106'                         6$:
006106' 104407  000000'                 BPEAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
006112' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
006116' 104407  000000'                 PREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
006122' 104407  000000'                 BPEAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
006126' 005367  173234                  DEC     CLK             ;COUNT TIMER TO ZERO
006132' 001341                          BNE     4$
006134' 000406                          BR      7$              ;DROP DRIVE
006136' 004767  000026          2$:     JSR     PC,CLEAR        ;CLEAR CU AND DRIVE
006142' 004767  177424                  JSR     PC,READY        ;IS DRIVE READY?
                                        ;       JMP     1$              ;YES THIS LINE DELETED 5-OCT-77,
006146' 000405                          BR      1$              ;ADDED 5-OCT-77
006150' 000240                          NOP                     ;ADDED 5-OCT-77
006152' 004767  000432          7$:     JSR     PC,DROP
006156' 007526'                         NOTRED
006160' 000462                          BR      3$
006162' 062716  000002          1$:     ADD     #2,(SP)
006166' 000207                  3$:     RTS     PC
```

```
                                        ;ROUTINE ISSUES A DRIVE CLEAR TO SELECTED DRIVE
                                        ;AND THEN ISSUES A DRIVE SELECT AND PACK ACKNOWLEDGE.
                                        ;WILL ALSO CLEAR ANY CONTROLLER ERRORS BEFORE RETURNING.

006170' 016777  172150  173210  CLEAR:  MOV     DSKADR,@RKCS2   ;GET DRIVE #
                                ;       MOV     #5,@RKCS1       ;ISSUE DRIVE CLEAR
006176' 016777  173374  173172          MOV     CLRDRV,@RKCS1           ;CLEAR THE DRIVE
006204' 004767  177314                  JSR     PC,CNTRDY
006210' 000240                          NOP
006212' 000240                          NOP
006214' 000240                          NOP
006216' 016777  172122  173162          MOV     DSKADR,@RKCS2
                                ;       MOV     #3,@RKCS1       ;ISSUE SELECT DRIVE AND PACK ACK.
006224' 016777  173356  173144          MOV     PACKAC,@RKCS1   ;ISSUE PACK ACKNOWLEDGE
006232' 004767  177266          1$:     JSR     PC,CNTRDY       ;WAIT FOR CU READY
006236' 012777  100000  173132          MOV     #BIT15,@RKCS1   ;CLEAR CU ERRORS
006244' 000240                          NOP
006246' 000240                          NOP
006250' 000240                          NOP
006252' 000207                          RTS     PC

                                        ;ROUTINE CALLS READY AND NOTRDY SUBROUTINES

006254' 004767  177312          RDY1:   JSR     PC,READY        ;CHECK FOR DRIVE AVAILABLE
006260' 000403                          BR      1$              ;GOOD RETURN
006262' 004767  177542                  JSR     PC,NOTRDY       ;WAIT FOR DRIVE AVAILABLE
006266' 000207                          RTS     PC              ;DRIVE NOT AVAILABLE
006270' 062716  000004          1$:     ADD     #4,(SP)
006274' 000207                          RTS     PC
```



```
006276' 004767  177752          RECAL:  JSR     PC,RDY1
006302' 000167  000154                  JMP     5$
006306'                         10$:
                                ;       MOV     #5,@RKCS1       ;CLR ATTN BIT
006306' 016777  173264  173062          MOV     CLRDRV,@RKCS1   ;CLEAR THE ATT BIT
006314' 004767  177204                  JSR     PC,CNTRDY       ;CONTROLER READY
006320' 005777  173052                  TST     @RKCS1          ;ANY ERRORS?
006324' 100447                          BMI     3$              ;YES-GO REPORT
006326' 004767  177722                  JSR     PC,RDY1
006332' 000167  000124                  JMP     5$
006336'                         20$:
                                ;       MOV     #13,@RKCS1      ;RECALIBRATE THE DRIVE
006336' 016777  173242  173032          MOV     RECALX,@RKCS1   ;RECALIBRATE THE DRIVE
006344' 005067  173016                  CLR     CLK             ;CLEAR TIMER
006350' 004767  177150                  JSR     PC,CNTRDY       ;WAIT FOR CONTROLER READY
006354' 005777  173016                  TST     @RKCS1          ;ANY ERRORS?
006360' 100431                          BMI     3$              ;YES-GO REPORT
006362' 032777  040000  173020  1$:     BIT     #BIT14,@RKDS    ;ATTN SET-OPERATION COMPLETE?
006370' 001010                          BNE     2$              ;YES-CONTINUE
006372' 104407  000000'                 BREAK$,REGIN            ;TEMPORARY RETURN TO MONITOR....
006376' 104407  000000'                 BREAK$,REGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
006402' 005367  172760                  DEC     CLK
006406' 100765                          BMI     1$              ;TIME LEFT?
006410' 000421                          BR      4$              ;NO
006412' 004767  177636          2$:     JSR     PC,RDY1
006416' 000167  000040                  JMP     5$
006422'                         30$:
                                ;       MOV     #5,@RKCS1       ;CLEAR ATTN BITS
006422' 016777  173150  172746          MOV     CLRDRV,@RKCS1   ;CLEAR ATTN BITS
006430' 004767  176722                  JSR     PC,DRVRDY       ;DRIVE READY?
006434' 000403                          BR      3$              ;NO - ERROR
006436' 062716  000002                  ADD     #2,(SP)         ;ADJUST STACK
006442' 000407                          BR      5$
006444' 004767  000140          3$:     JSR     PC,DROP         ;GO DROP DRIVE
006450' 007747'                         DRVERR
006452' 000403                          BR      5$
006454'                         4$:
006454' 004767  000130                  JSR     PC,DROP         ;DROP THE DRIVE IF NOT READY
006460' 007675'                         CALIB
006462' 000207                  5$:     RTS     PC              ;RETURN
```

```
1255  006464' 005267  171662          DRVADR: INC     DRVVE           ;COUNT A DRIVE
1256  006470' 005267  171650                  INC     DSKADR          ;NEXT DRIVE
1257  006474' 005767  171646                  TST     DVICE           ;ANY DRIVES LEFT?
1258  006500' 001005                          BNE     1$              ;YES
1259  006502' 012767  007550' 172766          MOV     #GONE,DRPMSG
1260  006510' 000167  173309                  JMP     ENTEST          ;STOP THE RUN
1261  006514' 042767  177770  171622  1$:     BIC     #177770,DSKADR  ;UP TO ADDR 7 ONLY
1262  006522' 022767  000010  171622          CMP     #8.,DRVVE       ;ALL DRIVES CHECKED ?
1263  006530' 001406                          BEQ     2$              ;YES, GO FLAG END OF DRIVES
1264  006532' 006267  171612                  ASR     DRIVE           ;NO, IS NEXT DRIVE CHOSEN ?
1265  006536' 103352                          BCC     DRVADR          ;NO, GO TRY ANOTHER DRIVE
1266  006540' 062716  000002                  ADD     #2,(SP)         ;ADJUST STACK POINTER
1267  006544' 000412                          BR      3$
1268
1269  006546' 012767  177777  171576  2$:     MOV     #-1,DRVVE       ;RESET DRIVE COUNTER
1270  006554' 012767  177777  171562          MOV     #-1,DSKADR      ;ZERO THE SHIFTED DRIVE #
1271  006562' 016767  171560  171560          MOV     DVICE,DRIVE     ;RESTORE CHOSEN DRIVES
1272  006570' 000406                          BR      4$
1273
1274  006572' 016767  171554  172704  3$:     MOV     DRVVE,NUMB      ;CURRENT DRIVE NUMBER IN MESSAGE STRING
1275  006600' 052767  000060  172676          BIS     #60,NUMB        ;MAKE IT ASCII
1276  006606' 000207                  4$:     RTS     PC
1277
1278
```



```
1279  006610'                         DROP:
1280  006610' 017667  000000  172660          MOV     @0(SP),DRPMSG
1281  006616' 104403  000000' 001470'         MSGNS,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
1282  006624' 012767  000006  171254          MOV     #6,ERRTYP       ;OFF LINE
1283                                          ;*****************************************************************
1284  006632' 104405  000000' 001376'         HPDEFS,BEGIN,RKCS1      ;
1285                                          ;*****************************************************************
1286  006640' 012701  000001                  MOV     #1,R1           ;INITIALIZE DROP PICKER
1287  006644' 016700  171502                  MOV     DRVVE,R0        ;GET THE DRIVE NUMBER
1288  006650' 001403                          BEQ     2$              ;IF DRIVE 0 GO DROP IT
1289  006652' 006301                  1$:     ASL     R1              ;NO, AIM AT THE NEXT DRIVE
1290  006654' 005300                          DEC     R0              ;IS THIS THE ONE ?
1291  006656' 001375                          BNE     1$              ;NO, LOOK AGAIN
1292  006660' 040167  171462          2$:     BIC     R1,DVICE        ;DROP THE DRIVE
1293  006664' 012777  100000  172504          MOV     #BIT15,@RKCS1   ;ISSUE CONTROLLER CLEAR
1294  006672' 004767  176626                  JSR     PC,CNTRDY       ;GO WAIT FOR CU READY
1295  006676' 005777  172474                  TST     @RKCS1          ;ANY ERRORS?
1296  006702' 100005                          BPL     3$
1297  006704' 012767  007725' 172564          MOV     #UNCLR,DRPMSG   ;SET ERROR MESSAGE
1298  006712' 000167  173076                  JMP     ENTEST          ;DROP MODULE
1299  006716' 062716  000002          3$:     ADD     #2,(SP)
1300  006722' 000207                          RTS     PC              ;RETURN
1301
1302
1303
```

```
1304  006724' 016700  171424          BLKADR: MOV     BLK1,R0         ;GET CURRENT BLOCK NUMBER
1305  006730' 042700  100000                  BIC     #100000,R0      ;CLEAR THE SIGN BIT
1306  006734' 012701  064734                  MOV     #27100.,R1      ;MAX BLOCK ADDRESS
1307  006740' 005002                          CLR     R2
1308  006742' 116702  171175                  MOVB    WBUFSZ+1,R2     ;GET NUMBER OF SECTORS
1309  006746' 160201                          SUB     R2,R1           ;SUBTRACK NUMBER OF SECTORS TRANSFERED
1310  006750' 020001                          CMP     R0,R1           ;MAKE SURE XFER WON'T RUN OFF END
1311  006752' 003401                          BLE     1$              ;ITS OK
1312  006754' 160100                          SUB     R1,R0           ;MAKE IT SOME SMALLER NUMBER
1313  006756'                         1$:
1314
1315  006756' 005067  171342                  CLR     CYLADR
1316  006762' 005067  171340                  CLR     SECADR
1317  006766' 022700  000102          2$:     CMP     #66.,R0         ;GET CYLINDER NUMBER
1318  006772' 101005                          BHI     3$              ;ON THIS TRACK
1319  006774' 005267  171324                  INC     CYLADR          ;NO - INCREMENT CYLINDER ADDRESS
1320  007000' 162700  000102                  SUB     #66.,R0         ;SECTORS PER CYLINDER
1321  007004' 000770                          BR      2$              ;FIND RIGHT CYLINDER
1322
1323  007006' 022700  000026          3$:     CMP     #22.,R0         ;FIND TRACK
1324  007012' 101005                          BHI     4$              ;ON RIGHT TRACK
1325  007014' 005267  171306                  INC     SECADR          ;BUMP TRACK COUNTER
1326  007020' 162700  000026                  SUB     #22.,R0         ;22 SECTORS PER TRACK
1327  007024' 000770                          BR      3$              ;CONTINUE
1328
1329  007026' 000367  171274          4$:     SWAB    SECADR          ;ALIGN TRACK BITS
1330  007032' 050067  171270                  BIS     R0,SECADR       ;COMBINE WITH SECTOR ADDRESS
1331  007036' 000207                          RTS     PC              ;RETURN
1332
```



```
1333  007040' 012700  000366'  CLRRB:  MOV     #RBUF,R0        ;CLEAR RBUF BUFFER
1334  007044' 016701  171062           MOV     RBUFSZ,R1       ;GET ITS ADDR AND SIZE
1335  007050' 005020           CLRCOM: CLR     (R0)+           ;CLEAR ANOTHER
1336  007052' 005301                   DEC     R1              ;COUNT ANOTHER
1337  007054' 001375                   BNE     CLRCOM          ;BR BACK TILL DONE
1338  007056' 000207                   RTS     PC
1339
1340
1341
1342
1343  007060' 016700  171260  RELEASE: MOV     DSKADR,R0
1344  007064' 052700  000010           BIS     #10,R0          ;PUT RELEASE BIT IN
1345  007070' 004767  177160           JSR     PC,RDY1
1346  007074' 000167  172560           JMP     RESTRT
1347
1348  007100' 010077  172302  1$:      MOV     R0,@RKCS2       ;WRITE THE COMMAND
1349                           ;        MOV     #1,@RKCS1       ;DO IT
1350  007104' 016777  172462  172264   MOV     SELDRV,@RKCS1   ;DO THE COMMAND
1351  007112' 000207                   RTS     PC
```

```
1352  007114' 001542'                 ERTAB:  SVCS1                   ;POINTER TO ADDRESS OF REGISTER
1353  007116' 007324'                         EMCS1                   ;START OF ERROR MESSAGES FOR THIS REGISTER
1354  007120' 141777                         141777                  ;BIT MASK
1355
1356  007122' 001552'                         SVCS2
1357  007124' 007264'                         EMCS2
1358  007126' 000377                         000377
1359
1360  007130' 001554'                         SVDS
1361  007132' 007244'                         EMDS
1362  007134' 173707                         173707
1363
1364  007136' 001556'                         SVER
1365  007140' 007144'                         EMER
1366  007142' 000000                         000000
1367
1368  007144' 046111  000106          EMER:   .ASCIZ  'ILF'           ;ILLEGEAL FUNCTION
1369  007150' 045523  000111                  .ASCIZ  'SKI'           ;SEEK INCOMPLETE
1370  007154' 054116  000106                  .ASCIZ  'NXF'           ;NON-EXECTUABLE FUNCTION
1371  007160' 051104  000120                  .ASCIZ  'DRP'           ;DRIVE PARITY ERROR
1372  007164' 046506  000124                  .ASCIZ  'FMT'           ;FORMAT
1373  007170' 052104  000131                  .ASCIZ  'DTY'           ;DRIVE TYPE ERROR
1374  007174' 041505  000110                  .ASCIZ  'ECH'           ;ERROR CORRECTION HARD
1375  007200' 051502  000105                  .ASCIZ  'BSE'           ;BAD SECTOR ERROR
1376  007204' 053110  000122                  .ASCIZ  'HVR'           ;HEADER VRC ERROR
1377  007210' 047503  000105                  .ASCIZ  'COE'           ;CYLINDER OVERFLOW ERROR
1378  007214' 042111  000101                  .ASCIZ  'IDA'           ;INVALID DISK ADDRESS ERROR
1379  007220' 046127  000105                  .ASCIZ  'WLE'           ;WRITE LOCK
1380  007224' 052104  000105                  .ASCIZ  'DTE'           ;DRIVE TIMMING ERROR
1381  007230' 050117  000111                  .ASCIZ  'OPI'           ;OPERATION INCOMPLETE
1382  007234' 047125  000123                  .ASCIZ  'UNS'           ;DRIVE UNSAFE
1383  007240' 041504  000113                  .ASCIZ  'DCK'           ;DATA CHECK
1384
1385  007244' 041501  000114          EMDS:   .ASCIZ  'ACL'           ;DRIVE AC POWER LOW
1386  007250' 041504  000114                  .ASCIZ  'DCL'           ;DRIVE DC POWER LOW
1387  007254' 047504  000124                  .ASCIZ  'DOT'           ;DRIVE OFF TRACK
1388  007260' 051127  000114                  .ASCIZ  'WRL'           ;WRITE LOCK
1389
1390  007264' 043125  000105          EMCS2:  .ASCIZ  'UFE'           ;UNIT FIELD ERROR
1391  007270' 042115  000123                  .ASCIZ  'MDS'           ;MULTIPLE DRIVE SELECT
1392  007274' 043520  000105                  .ASCIZ  'PGE'           ;PROGRAMMING ERROR
1393  007300' 042516  000115                  .ASCIZ  'NEM'           ;NON EXISTANT MEMORY
1394  007304' 042516  000104                  .ASCIZ  'NED'           ;NON EXISTANT DRIVE
1395  007310' 050125  000105                  .ASCIZ  'UPE'           ;UNIBUS PARITY ERROR
1396  007314' 041527  000105                  .ASCIZ  'WCE'           ;WRITE CHECK ERROR
1397  007320' 046104  000124                  .ASCIZ  'DLT'           ;DATA LATE
1398
1399  007324' 042103  000124          EMCS1:  .ASCIZ  'CDT'           ;CONTROLLER DRIVE TYPE
1400  007330' 052103  000117                  .ASCIZ  'CTO'           ;CONTROLLER TIMEOUT
1401  007334' 043103  000115                  .ASCIZ  'CFM'           ;CONTROLLER FORMAT
1402  007340' 050123  000101                  .ASCIZ  'SPA'           ;SERIAL PARITY ERROR
1403  007344' 042503  000122                  .ASCIZ  'CEP'           ;CONTROLLER ERROR
1404
1405  007350' 020040      000         BLNKS:  .ASCIZ  '  '            ;TWO BLANKS
1406          007354'                         .EVEN
```



```
1407
1408  007354' 044040  051101  020104  HARD:   .ASCIZ  ' HARD ERROR'
1409  007362' 051105  047522  000122
1410  007370' 051440  043117  020124  SOFT:   .ASCIZ  ' SOFT ERROR'
1411  007376' 051105  047522  000122
1412  007404' 042040  044522  042526  DRVE:   .ASCIZ  ' DRIVE '
1413  007412' 000040
1414  007414' 042040  047522  050120  DRPED:  .ASCIZ  ' DROPPED '
1415  007422' 042105  000040
1416  007426' 042040  044522  042526  LOAD:   .ASCIZ  ' DRIVE 0 DROPEED LOAD MEDIUM%'
1417  007434' 030040  042040  047522
1418  007442' 042520  042105  046040
1419  007450' 046517  020104  042515
1420  007456' 044504  046525  000045
1421  007464' 053440  044522  042524  PROT:   .ASCIZ  ' WRITE PROT.%'
1422  007472' 050040  047522  027124
1423  007500' 000045
1424  007502' 042040  044522  042526  UNAVAL: .ASCIZ  ' DRIVE UNAVAILABLE%'
1425  007510' 052440  040516  046526
1426  007516' 046111  041101  042514
1427  007524' 000045
1428  007526' 042040  044522  042526  NOTRED: .ASCIZ  ' DRIVE NOT READY%'
1429  007534' 047040  052117  051040
1430  007542' 040505  054504  000045
1431  007550' 047040  020117  053101  GONE:   .ASCIZ  ' NO AVAILABLE DRIVES%'
1432  007556' 044501  040514  046102
1433  007564' 020105  051104  053111
1434  007572' 051505  000045
1435  007576' 051040  052105  054522  RTRY:   .ASCIZ  ' RETRY EXCEEDED%'
1436  007604' 042440  041530  042505
1437  007612' 042504  022504      000
1438  007617'     040  047503  052116  CNTRNV: .ASCIZ  ' CONTROLLER NOT READY%'
1439  007624' 047522  046114  051105
1440  007632' 047040  052117  051040
1441  007640' 040505  054504  000045
1442  007646' 042440  052130  040522  EXTRA:  .ASCIZ  ' EXTRANEOUS INTERRUPT%'
1443  007654' 042516  052517  020123
1444  007662' 047111  042524  051122
1445  007670' 050125  022524      000
1446  007675'     040  047125  041101  CALIB:  .ASCIZ  ' UNABLE TO RECALIBRATE%'
1447  007702' 042514  052040  020117
1448  007710' 042522  040503  044514
1449  007716' 051102  052101  022505
1450  007724'     000
1451  007725'     040  047125  041101  UNCLR:  .ASCIZ  ' UNABLE TO CLEAR%'
1452  007732' 042514  052040  020117
1453  007740' 046103  040505  022522
1454  007746'     000
1455  007747'     040  051104  053111  DRVERR: .ASCIZ  ' DRIVE ERROR%'
1456  007754' 020105  051105  047522
1457  007762' 022522      000
1458  007765'     117  044124  051105  PORTHG: .ASCIZ  'OTHER PORT NOT UPDATING BLOCK ZERO%'
1459  007772' 050040  051117  020124
1460  010000' 047515  020124  050125
1461  010006' 040504  044524  043516
1462  010014' 041040  047514  045503
```

```
 1463  010022' 055040  051105  022517
 1464  010030'    000
 1465  010031'    125  040516  046102  SPIND:  .ASCIZ  "UNABLE TO START SPINDLE?"
 1466  010036' 020105  047524  051440
 1467  010044' 040524  052122  051440
 1468  010052' 044520  042116  042514
 1469  010060' 000045
 1470                                  .EVEN
 1471          000001                  .END
```



```
ACSR     000102R          303#    818*
ADDR     000006R          269#    588
ADDP22=  001000           327#
ASB      000106R          307#
ASTAT    000104R          305#
AWFLAG   000314R          347#
AWAS     000110R          308#
BADSPT   000252R          330#    725
BEGIN    000000R          266#    455     463     477     483     484     493     513     514     521     538     539     559
                          560     581     627     630     659     660     673     717     800     803     829     832     841
                          844     891     894    1074    1075    1082    1090    1091    1166    1167    1168    1169    1234
                         1235    1281    1294
BIT0   = 000001           327#    387     852    1108    1112    1118    1160
BIT1   = 000002           327#    387
BIT10  = 002000           327#    387     388     389    1134    1141
BIT11  = 004000           327#    388     389     390     702
BIT12  = 010000           327#    387     388     389    1121
BIT13  = 020000           327#    388     389
BIT14  = 040000           327#    387     808    1232
BIT15  = 100000           327#    389     439     440     646     657     801     808     819     874    1070    1124    1147
                         1153    1199    1293
BIT2   = 000004           327#    387     870
BIT3   = 000010           327#    387     390     885
BIT4   = 000020           327#    390
BIT5   = 000040           327#    387     390     623     834    1129
BIT6   = 000100           327#    439     715     801     876
BIT7   = 000200           327#    495     617     866    1072    1088
BIT8   = 000400           327#    389    1126
BIT9   = 001000           327#    387     389
BLK      001424R          384#
BLKADR   006724R          489     548    1304#
BLK1     000354R          363#    384     415     449*    502     547*    714*    718*    719     721     723*    727    1304
BLNKS    007350R          842    1405#
BREAK$=  104407           327#    513     514     538     539     559     560     659     660    1074    1075    1090    1091
                         1166    1167    1168    1169    1234    1235
BR1      000012R          271#    616
BR2      000013R          272#
BSRD     004772R          868     960#
BTODS  = 104421           327#
BUFADR   000334R          355#    736*    747*    754*    761*    768*    787
CALIB    007675R         1253    1446#
CDATA$=  104412           327#    493     521
CDERCT   000144R          322#
CDWDCT   000146R          323#
CHKDRV   002720R          452     623#
CLEAR    006170R         1173    1188#
CLK      001366R          369#    653*    651*   1057*   1076*   1152*   1170*   1228*   1236*
CLK1     001370R          370#   1087*   1092*
CLRCOM   007050R         1335#   1337
CLRCTR   001602R          440#
CLRDRV   001576R          438#    668    1190    1219    1243
CLRRB    007040R          500     552    1333#
CNT      000336R          356#    458*
CNTRDV   005524R          624     643     654     697     820     835    1058    1067    1087#   1106    1119    1154    1159
                         1191    1198    1220    1229    1294
CNTRNV   007617R         1094    1438#
```

```
CNTR1   001464R          393#    922*    951*
CNTR2   001466R          394#    931*    948*
CONFIG  000056R          291#
CSRA    000100R          301#    817*
CYCLE   002026R          469     486#
CYCLEB  002232R          487     525#
CYLADR  000324R          351#    781     1315*   1319*
CYLCNT  000332R          354#
DATCKS= 104411           327#
DATERS= 104404           327#
DISINT  001600R          439#
DPICNT  000360R          365#    619
DRIVE   000350R          361#    611*    1264*   1271*
DROP    006610R          508     544     675     682     704     707     823     854     1178    1248    1252    1279#
DRP     001470R          396#    1281
DRPED   007414R          398     1414#
DRPMSG  001476R          399#    483     580*    581     672*    673     828*    829     840*    841     1078*   1080*   1082
                         1094*   1259*   1280*   1297*
DRVADR  006464R          633     690     1255#   1265
DRVE    007404R          396     954     1412#
DRVERR  007747R          672     708     1080    1249    1455#
DRVRDY  005359R          662     700     989     1057#   1244
DRVVE   000352R          362#    460*    612*    1255*   1262    1269*   1274    1287
DSHARD= 004070           390#    915
DSKADR  000344R          359#    461*    613*    640     650     694     973     1063    1103    1117    1155    1188    1195
                         1256*   1261*   1270*   1343
DVICE   000346R          360#    473     567*    577     579*    611     1257    1271    1292*
DVID1   000014R          273#    567
ECCOR   005242R          684     1018#
EMCS1   007324R          1353    1399#
EMCS2   007264R          1357    1390#
EMDS    007244R          1361    1385#
EMER    007144R          1365    1368#
ENDIT$= 104413           327#    477
ENDS    = 104410         327#    484
ENTEST  002014R          475     482#    632     846     1095    1260    1298
ERBUF   001430R          392#    891     909     944
ERHARD= 053057           387#    917
ERRGEN  004550R          864     907#
ERRORS  004370R          848     863#
ERRTYP  000106R          306#    628*    630*    842*    892*    1282*
ERTAB   007114R          908     923     1352#
EXCUTE  003732R          738     742     749     756     763     781#
EXCUT1  003746R          772     783#
EXCUT2  003754R          784#    851
EXITS   = 104400         327#    860
EXTRA   007646R          828     1442#
FLAG    000312R          346#    450*    472*    488*    501     527     553*    554
FREE    000150R          324#
FUNC    000340R          357#    734*    741*    745*    752*    759*    766*    775*    796
GETPA$= 104415           327#    455
GONE    007550R          1259    1431#
GVBUF$= 104414           327#    463
HARD    007354R          910     1408#
HRDCNT  000044R          286#
HRDER$= 104405           327#    630     832     844     894     1284
```



```
HRDPAS  000050R          288#
HSRD    001502R          401#    960*    966*    1009
ICNT    000356R          364#    619*
ICONT   000036R          293#
ICOUNT  000040R          294#
IDNUM   000122R          313#
IMODX.= 000000           325#    464
INIT    000030R          290#
INTFLG  001374R          372#    459*    784*    799*    806
INTR    006120R          312#    448*
INTSRV  004062R          615     801#
LOAD    007426R          580     1416#
LOOP1   001714R          460#    479
LOOP2   001752R          466#    470
LOOP3   001760R          469#    839
LOOP4   001766R          468     472#
MAP22$= 104416           327#
MODNAM  000000R          267#
MODSP   000252R          281     325#    454     838
MSGNS   = 104403         327#    483     581     627     673     829     841     891     1082    1281
MSGS$   = 104402         327#
MSGS    = 104401         327#
NOTRDY  000030P          1152#   1210
NOTRED  007526R          1078    1179    1428#
NULL    = 000000         327#
NUMB    001504R          397     402#    955     1274*   1275*
NXTDRV  003236R          467     690#    693     706     709
OPEN    = 000000         268     274     275     276     277     294     295     296     297     298     299     300     301
                         303     305     307     308     310     311     312     315     316     318     319     321     322
                         323     324     327#
OTOA$ = 104420           327#
PACKAC  001606R          442#    642     1197
PASCNT  000034P          282#
PICKBK  003346R          462     714#    720     724     728
PIRQS   = 000004         327#    803
POPSP   = 005726         327#
POPSP2= 022626           327#
PORTHG  007765R          509     545     1458#
PROT    007464R          705     1421#
PRTY    = 000000         327#
PRTY0 = 000000           272     327#
PRTY1 = 000040           327#
PRTY2 = 000100           327#
PRTY3 = 000140           327#
PRTY4 = 000200           327#
PRTY5 = 000240           271     327#
PRTY6 = 000300           327#
PRTY7 = 000340           327#
PS      = 177776         327#
PSW     = 177776         327#
PUSH    = 005746         327#
PUSH2 = 024646           327#
RAND$ = 104417           327#    717
RANNUM  000054R          290#    718
RBUF    000366R          314     368#    501*    502*    518     529     531     533     547     554*    995     1333
RBUFEA  000130R          316#    748     755     762     769     981
```

```
RBUFPA  000126R          315#    493     521     747     754     761     768     975
RBUFSZ  000132R          317#    456    1334
RBUFVA  000124R          314#    455
RDY1    006254R          638     648     664     692     795     970    1061    1208#   1215    1223    1239    1345
RDOO    003722R          517     526     775#
READ    003522R          492     520     549     745#
READY   005572R          635    1100#   1174    1208
RECAL   006276R          690     900    1215#
RECALX  001604R          441#   1227
RELEAS  007060R          505     535     556    1343#
RESTRT  001660R          309     453#    786     825     860     971    1346
RES1    000056R          292#
RES2    000060R          293#
RETRY   001506R          403#    783*    850*    855     888
RKASOF  001414R          390#    603*
RKBA    001402R          375#    593*    787*    975*
RKCHK   005654R         1101    1117#
RKCS1   001376R          373#    580*    625     630     642*    644     652*    655     668*    696*    698     798*    801*
                         808     812     817     819*    821     836     844     988*    991    1059    1064*   1068    1088
                        1105*   1118*   1147*   1153*   1158*   1190*   1197*   1199*   1219*   1221    1227*   1230    1243*
                        1284    1293*   1295    1350*
RKCS2   001406R          377#    597*    623*    640*    650*    694*    834*    973*   1063*   1103*   1117*   1120    1155*
                        1188*   1195*   1348*
RKDA    001404R          376#    595*    790*    980*
RKDC    001416R          381#    605*    789*    976*    979*
RKDS    001410R          378#    599*    646     657     702    1070    1072    1107    1108    1123    1160    1232
RKECPS  001420R          382#    608*
RKECPT  001422R          383#    610*
RKER    001412R          379#    601*   1128
RKFLG   001610R          443#    797     977     987
RKWC    001400R          374#    591*    788*    974*
RSTRT   000112R          309#
RTRY    007576R         1435#
SBADR   000102R          302#
SECADR  000326R          352#    782    1005    1316*   1325*   1329*   1330*
SELDRV  001572R          436#    696    1064    1105    1158    1350
SETUP   002446R          451     567#
SFTERR  001372R          371#    898     907*    920*
SOFCNT  000042R          285#
SOFERS= 104406           327#
SOFPAS  000046R          287#
SOFT    007370R          919    1410#
SPIND   010031R          676    1465#
SPINDL  001574R          437#    652
SPOINT  000032R          281#
SPSIZ = 000040             1#    325
SR1     000016R          274#    486     495     617     715     852     870     885
SR2     000020R          275#
SR3     000022R          276#
SR4     000024R          277#
START   001612R          280     446#
STAT    000026R          279#
SVASOF  001560R          411     425#
SVBA    001546R          406     420#   1018    1045
SVCS1   001542R          404     418#    813     818     826     879     913    1352
SVCS2   001552R          408     422#    911    1356
```



```
SVDA    001550R          407     421#
SVDC    001562R          412     426#   1003
SVDS    001554R          409     423#    915    1360
SVECPS  001564R          413     427#   1020
SVECPT  001566R          414     428#   1030
SVER    001556R          410     424#    866     874     876     917    1364
SVRQ    000062R          294#
SVR1    000064R          295#
SVR2    000066R          296#
SVR3    000070R          297#
SVR4    000072R          298#
SVR5    000074R          299#
SVR6    000076R          300#
SVWC    001544R          405     419#
SYSCNT  000052R          289#
S1HARD= 036000           388#    913
S2HARD= 137400           389#    911
TABLE   001510R          404#    832     894
TCYL    000316R          348#    770*    791*    789
TIMER   000322R          350#    504*    506*    525*    542*
TRPDFD= 000022           327#
TSEC    000320R          349#    771*    782*    790
UNAVAL  007562R          683    1424#
UNCLR   007725R          627     824     840    1297    1451#
UNLOAD  001570R          435#   1134*   1141*
VECTOR  000010R          270#    614
WASADR  000104R          304#
WBUFEA  000136R          319#    737
WBUFPA  000134R          318#    736
WBUFRQ  000140R          320#
WBUFSZ  000142R          321#    464    1308
WCNT1   000362R          366#    464*    465*    735
WCNT2   000364R          367#    456*    457*    746     753     760     767
WDCNT   000330R          353#    735*    746*    753*    760*    767*    788
WDFR    000116R          311#    446*
WDTO    000114R          310#    447*
WRITCB  003616R          551     759#
WRITCK  003506R          491     741#
WRITE   003450R          490     734#
WRITEB  003560R          550     752#
WRT00   003654R          503     555     766#
WRT00R  003662R          767#    776
XFLAG   000005R          268#
XMEM    000342R          358#    737*    748*    755*    762*    769*    791
.     = 010062R          368#    392#    494     522     978    1406#

. ABS.  000000     000
        010062     001

ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  3

XRKBD0,XRKBD0/SOL/CRF:SYM=DDXCOM,XRKBD0
RUN-TIME: 2 3 .5 SECONDS
```

RUN-TIME RATIO: 26/6=3.7
CORE USED: 7K (13 PAGES)