RK06

DEVICE ROUTINE (MPG)

MD-11-DTR6A-A

EP-DTR6A-A-DL-A

NOV 1976

digital

MADE IN U.S.A.

```
                .REM    %

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

                PRODUCT CODE:   MAINDEC-11-DTR6A-A

                PRODUCT NAME:   RK611 - RK06 DISK DEVICE ROUTINE
                                FOR MPG

                DATE:           JULY 1976

                MAINTAINED BY:  DIAGNOSTIC ENGINEERING

                AUTHOR:         A. W. LEIGH
```



```
                %
```

```
                        .SBTTL  REVISION HISTORY

;       JUL 76  DTR6A-A INITIAL RELEASE AS A FULL SUPPORT
;                       DEVICE ROUTINE FOR THE RK06 DISK.
```

```
                                .SBTTL  STANDARD DEVICE ROUTINE TABLE
                        .TITLE  MAINDEC-11-DTR6A-A  RK611 - RK06 DEVICE ROUTINE FOR MPG

                                ;REVISION 'A'

                                ;FILENAME OF "TR6AA0.MPG" ON MPG/XXDP MEDIA

                                ;MACY11:  DTR6AA,DTR6AA/CRF:SYM/DOC=DTR6AA.P11
                                ;LNKX11:  DTR6AA.MPG/B:0=DTR6AA/E
                                ;PAPER TAPE:  PUNCH DTR6AA.MPG/FILE:ELEV


000000'                         .CSECT  RJP11
                                .DSABL  GBL
                        ;
                        ;E Q U I V A L E N T   S T A T E M E N T S
                        ;

                        ;PROGRAM EQUALS
170200                  BUSMAP=170200                   ;UNIBUS ADDR OF BUS-MAP REGS
072460                  ITIME=30000.                    ;INTERRUPT TIME-COUNT
000400                  NRWORD=256.                     ;NR WORDS IN ONE SECTOR
001000                  NRBYTE=2*NRWORD                 ;NR BYTES IN ONE SECTOR
000633                  NRCYL=411.                      ;NR OF CYLINDERS
000003                  NRHEAD=3                        ;NR OF HEADS
177776                  PS=177776                       ;PSW
000020                  REGNUM=16.                      ;NR OF DEVICE REGISTERS
000003                  SDMAX=3                         ;HIGHEST VALUE OF SELECT-DRIVE CODE
002412                  STSLUP=2412                     ;LOOP CONTROL FOR "STSTAT" ROUTINE
000000                  XXXX=0                          ;VALUE TO BE TAYLORED BY DEVICE ROUTINE

                        ;DEVICE REGISTGER NAMES
000000                  RPCS1=0
000002                  RPWC=2
000004                  RPBA=4
000006                  RPDA=6
000010                  RPCS2=10
000012                  RPDS=12
000014                  RPER=14
000014                  RPER1=RPER
000016                  RPAS=16
000020                  RPDC=20
000022                  RKMU=22
000024                  RPDB=24
000026                  RPMR1=26
000030                  RPEC1=30
000032                  RPEC2=32
000034                  RPMR2=34
000036                  RPMR3=36


                        ;RKCS1 DEVICE BIT EQUATES
100000                  CCLR=100000
040000                  DI=40000
020000                  SPAR=20000
010000                  CFMT=10000
```

```
                        004000          CTO=4000
                        000200          RDY=200
                        000100          IE=100
                        000001          GO=1
                                        ;RKCS2 DEVICE BIT EQUATES
                        100000          DLT=100000
                        040000          MCE=40000
                        020000          UPE=20000
                        010000          NED=10000
                        004000          NEM=4000
                        002000          PGE=2000
                        001000          MDS=1000
                        000400          UFE=400
                        000200          OR=200
                        000040          SCLR=40
                        000020          BAI=20
                        000010          RLS=10
                                        ;RKDS DEVICE BIT EQUATES
                        100000          SVAL=100000
                        040000          DSC=40000
                        004000          WRL=4000
                        000200          DRDY=200
                        000100          VV=100
                        000040          DROT=40
                        000020          DSL=20
                        000010          ACLO=10
                        000001          DRA=1
                                        ;RKER DEVICE BIT EQUATES
                        100000          DCK=100000
                        040000          UNS=40000
                        020000          OPI=20000
                        010000          DTE=10000
                        004000          WLE=4000
                        002000          IDAE=2000
                        001000          COE=1000
                        000400          HVRC=400
                        000200          BSE=200
                        000100          ECH=100
                        000040          DTYE=40
                        000020          FMTE=20
                        000010          DRPAR=10
                        000004          NXF=4
                        000002          SKI=2
                        000001          ILF=1
                                        ;RKMR1 DEVICE BIT EQUATES
                        000020          PAT=20
                                        ;SYSTEM FLAG-WORD BIT DEFINITIONS
                        000001          MMVER=1
                        000002          USMTPS=2
                        000010          CPU70=10
                        000040          UNIMAP=40
```

```
                                ;DISK COMMAND CODES
000001                          SDCODE=1
000003                          PACODE=3
000005                          DCCODE=5
000007                          UCODE=7
000011                          SSCODE=11
000013                          RCCODE=13
000015                          OCODE=15
000017                          SCODE=17
000021                          RCODE=21
000023                          WCODE=23
000025                          RHCODE=25
000027                          WHCODE=27
000031                          WCCODE=31

                                ;DEVICE ROUTINE (DFLGWD) FLAG BITS
100000                          WAITMD=100000           ;WAIT MODE, 0=WAIT
002000                          CORFLG=2000             ;CORRECTION MODE, 0=CORON
000200                          ANYIOI=200              ;ANY I/O HAS BEEN ISSUED
000100                          CMDISU=100              ;I/O COMMAND HAS BEEN ISSUED
001000                          VVFLG=  1000            ;VV MODE - 0 = VVON
000400                          SPINFL= 400             ;SPIN COMMAND IN PROGRESS
000040                          SWOVTO= 40              ;TIMEOUT ON SWITCH OVER
000020                          SWOIER= 20              ;ERROR ON INT FOR SWITCH OVER
000010                          SWOVER= 10              ;ACQUIRING DISK - SWITCH OVER
000002                          IOTERM=2                ;PROCESS I/O TERMINATION
000001                          IOERR=1                 ;ERROR ON CURRENT I/O

                                ;MEMORY MANAGEMENT EQUATES
172350                          KPARH=172350
172310                          KPDRH=172310
100000                          P4CONS=100000
077406                          PDRCON=077406

                                ;DIAGNOSTIC FLAG EQUATES
000001                          DRIB=1                  ;DRIVE IS BUSY
000002                          NEE=2                   ;NO ERRORS EXPECTED
000004                          NDC=4                   ;NEED DRIVE CLEAR
000010                          NRC=10                  ;NEED RE-CALIBRATE
000020                          NCC=20                  ;NEED CONTROLLER/DRIVE CLEAR
000040                          PBSH=40                 ;PRINT BAD SECTOR HEADING
```

```
                        ;
                        ;M P G   I N T E R F A C E
                        ;
                        ;THE FOLLOWING TABLE IS IN THE STANDARDIZED FORMAT REQUIRED
                        ;  TO INTERFACE WITH MPG.

000000' 015544          LOC2:   .WORD   DVREND-.        ;DEVICE ROUT SIZE IN BYTES
000002' 000000          DFLGWD: .WORD   0               ;DEVICE ROUT FLAGWORD
000004' 000000          CYL:    .WORD   0               ;CYLINDER # (0 THRU 410.)
000006' 000000          HEAD:   .WORD   0               ;HEAD # (0 THRU 2.)
000010' 000000          SECT:   .WORD   0               ;SECTOR # (0 THRU 21./19.)
000012' 000003          RTRY:   .WORD   3               ;# OF RETRY ATTEMPTS
000014' 000000          MSGA:   .WORD   0               ;SAVE MSG A
000016' 000000          MSGB:   .WORD   0               ;SAVE MSG B
000020' 000001          SIZE:   .WORD   1               ;# OF BYTES TRANSFERRED / UNIMAP FLG
000022' 000000          ERRI:   .WORD   0               ;ERROR ON LAST I/O INDICATOR
000024' 177440          DREGAD: .WORD   177440          ;FIRST DEVICE REGISTER ADR
000026' 000210          IVCTAD: .WORD   210             ;INTERRUPT VECTOR ADR
000030' 000240          PSWD:   .WORD   240             ;INT PROC STATUS WORD (BR 5)
000032' 000000                  .WORD   0               ;NOT USED
000034' 002042                  .WORD   HSKEEP-.        ;HOUSEKEEPING ROUT REL ADR
000036' 002122                  .WORD   REPORT-.        ;REPORT ROUT REL ADR
000040' 002730                  .WORD   KILL-.          ;KILL ROUT REL ADR
000042' 001700                  .WORD   DATAER-.        ;DATA ERROR COUNTER REL ADR
000044' 002456                  .WORD   TOUTER-.        ;TIME OUT ERROR ROUT REL ADR
000046' 000000          CIOBSY: .WORD   0               ;I/O BUSY BRANCH ADR
000050' 000000          CUPGER: .WORD   0               ;DEVICE ERROR BRANCH ADR
000052' 000000          ULIST:  .WORD   0               ;USER MODE PRINT ROUTINE BRANCH ADR
000054' 000000          CLIST:  .WORD   0               ;CMND MODE PRINT ROUTINE BRANCH ADR
000056' 000000          BINASC: .WORD   0               ;CONVERT BINARY TO ASCII ROUT BR ADR
000060' 000000          BTASLZ: .WORD   0               ;CONVERT BINARY TO DECIMAL ASCII BR ADR
000062' 000000          DECASC: .WORD   0               ;CONVERT PACKED DECIMAL TO ASCII BR ADR
000064' 000000          CSYSFW: .WORD   0               ;MPG SYSTEM FLAGWORD ADR
000066' 000000          SETVEC: .WORD   0               ;SET INT VECT ROUT BR ADR
000070' 000000          CLRVEC: .WORD   0               ;CLEAR INT VECTOR ROUT BR ADR
000072' 000000          TSTVEC: .WORD   0               ;TEST INT VECTOR ROUT BR ADR
000074' 000000          RTNINT: .WORD   0               ;RETURN FROM INT ROUT BR ADR
000076' 000000          GETBYT: .WORD   0               ;GET DATA BYTE ROUT BR ADR
000100' 000000          PUTBYT: .WORD   0               ;PUT DATA BYTE ROUT BR ADR
000102' 000014                  .WORD   DVREGS-.        ;ADR OF DEVICE REGISTER NAMES
000104' 000152                  .WORD   DVCMDS-.        ;ADR OF DEVICE FUNCTIONS
000106' 000346                  .WORD   DVPKTE-.        ;ADR OF PACK TBL EXTENSION
000110' 000714                  .WORD   DVMVTE-.        ;ADR OF MODEL VECTOR TBL EXTEN.
000112' 001076                  .WORD   DVCPTE-.        ;ADR OF COMPILER TBL EXTEN.
000114' 001406                  .WORD   DVIWST-.        ;ADR OF DEV INTERFACE WD SYM TBL
```

```
                                .SBTTL  COMPILER TABLES & CONSTANT AREAS

                                ;
                                ;C O M P I L E R   T A B L E S   &   C O N S T A N T   A R E A S
                                ;
000116' 045522 030503   DVREGS: .ASCII  /RKC1/                  ;VALID DEVICE REGISTER NAMES &
000122' 000000                  .WORD   0                       ;THEIR POSITIONS RELATIVE TO
000124' 045522 041527           .ASCII  /RKWC/
000130' 000002                  .WORD   2
000132' 045522 040502           .ASCII  /RKBA/
000136' 000004                  .WORD   4
000140' 045522 040504           .ASCII  /RKDA/
000144' 000006                  .WORD   6
000146' 045522 031103           .ASCII  /RKC2/
000152' 000010                  .WORD   10
000154' 045522 051504           .ASCII  /RKDS/
000160' 000012                  .WORD   12
000162' 045522 051105           .ASCII  /RKER/
000166' 000014                  .WORD   14
000170' 045522 051501           .ASCII  /RKAS/
000174' 000016                  .WORD   16
000176' 045522 041504           .ASCII  /RKDC/
000202' 000020                  .WORD   20
000204' 047516 052524           .ASCII  /NOTU/
000210' 000022                  .WORD   22
000212' 045522 041104           .ASCII  /RKDB/
000216' 000024                  .WORD   24
000220' 045522 030515           .ASCII  /RKM1/
000224' 000026                  .WORD   26
000226' 045522 047520           .ASCII  /RKP0/
000232' 000030                  .WORD   30
000234' 045522 040520           .ASCII  /RKPA/
000240' 000032                  .WORD   32
000242' 045522 031115           .ASCII  /RKM2/
000246' 000034                  .WORD   34
000250' 045522 031515           .ASCII  /RKM3/
000254' 000036                  .WORD   36
        000256'         DVREGE= .

                                ;
000256'     120     201 DVCMDS: .BYTE   120,201                 ;VALID DEVICE FUNCTIONS
000260' 004130                  .WORD   READ-.                  ;  FLAG BYTE:
000262'     130     201         .BYTE   130,201                 ;    BIT 7 = NPR DEV
000264' 004156                  .WORD   WRITE-.                 ;    BIT 3 = MASSBUS DEV
000266'     376     000         .BYTE   376,0                   ;    BIT 0 = 2 WORDS FOR ADR
000270' 003154                  .WORD   NOWAIT-.                ;            (18 BIT ADRS)
000272'     375     000         .BYTE   375,0
000274' 003130                  .WORD   WAIT-.
000276'     374     000         .BYTE   374,0
000300' 001660                  .WORD   REPORT-.
000302'     373     000         .BYTE   373,0
000304' 001654                  .WORD   REPORT-.
000306'     372     000         .BYTE   372,0
000310' 004446                  .WORD   SEEK-.
000312'     371     000         .BYTE   371,0
```

```
000314' 004510                          .WORD   SELDRI-.
000316'    370  201                     .BYTE   370,201
000320' 004406                          .WORD   WRCK-.
000322'    367  000                     .BYTE   367,0
000324' 004356                          .WORD   RDHD-.
000326'    366  201                     .BYTE   366,201
000330' 004364                          .WORD   WRHD-.
000332'    365  000                     .BYTE   365,0
000334' 003220                          .WORD   CRESET-.
000336'    364  000                     .BYTE   364,0
000340' 003342                          .WORD   DRESET-.
000342'    363  000                     .BYTE   363,0
000344' 003260                          .WORD   SRESET-.
000346'    362  000                     .BYTE   362,0
000350' 002452                          .WORD   STEPUP-.
000352'    361  000                     .BYTE   361,0
000354' 002620                          .WORD   STEPDN-.
000356'    360  000                     .BYTE   360,0
000360' 004420                          .WORD   OFFSET-.
000362'    357  000                     .BYTE   357,0
000364' 003332                          .WORD   SPIN-.
000366'    356  000                     .BYTE   356,0
000370' 003320                          .WORD   UNLOAD-.
000372'    355  000                     .BYTE   355,0
000374' 004416                          .WORD   RECAL-.
000376'    354  000                     .BYTE   354,0
000400' 003324                          .WORD   PAKACK-.
000402'    353  000                     .BYTE   353,0
000404' 004160                          .WORD   BADSEC-.
000406'    352  000                     .BYTE   352,0
000410' 003322                          .WORD   REL-.
000412'    347  000                     .BYTE   347,0
000414' 003120                          .WORD   FMT22-.
000416'    346  000                     .BYTE   346,0
000420' 003124                          .WORD   FMT20-.
000422'    345  000                     .BYTE   345,0
000424' 003030                          .WORD   000-.
000426'    344  000                     .BYTE   344,0
000430' 003034                          .WORD   EVEN-.
000432'    337  000                     .BYTE   337,0
000434' 003040                          .WORD   BAION-.
000436'    336  000                     .BYTE   336,0
000440' 003044                          .WORD   BAIOFF-.
000442'    335  000                     .BYTE   335,0
000444' 003050                          .WORD   CORON-.
000446'    334  000                     .BYTE   334,0
000450' 003054                          .WORD   COROFF-.
000452' 177777                          .WORD   177777          ;TABLE TERMINATOR

000454' 047516  040527  052111  DVPKTE: .ASCII  /NOWAIT/        ;PACK TABLE EXTENSION
000462'    376     000                  .BYTE   376,0
000464' 020040  040527  052111          .ASCII  /  WAIT/
000472'    375     000                  .BYTE   375,0
000474' 052123  052101  051525          .ASCII  /STATUS/
000502'    374     000                  .BYTE   374,0
000504' 047503  047125  051524          .ASCII  /COUNTS/
```

```
000512'    373     000                      .BYTE   373,0
000514'  020040  042523  045505             .ASCII  /  SEEK/
000522'    372     000                      .BYTE   372,0
000524'  042523  042114  044522             .ASCII  /SELDRI/
000532'    371     000                      .BYTE   371,0
000534'  020040  051127  045503             .ASCII  /  WRCK/
000542'    370     000                      .BYTE   370,0
000544'  020040  042122  042110             .ASCII  /  RDHD/
000552'    367     000                      .BYTE   367,0
000554'  020040  051127  042110             .ASCII  /  WRHD/
000562'    366     000                      .BYTE   366,0
000564'  051103  051505  052105             .ASCII  /CRESET/
000572'    365     000                      .BYTE   365,0
000574'  051104  051505  052105             .ASCII  /DRESET/
000602'    364     000                      .BYTE   364,0
000604'  051123  051505  052105             .ASCII  /SRESET/
000612'    363     000                      .BYTE   363,0
000614'  052123  050105  050125             .ASCII  /STEPUP/
000622'    362     000                      .BYTE   362,0
000624'  052123  050105  047104             .ASCII  /STEPDN/
000632'    361     000                      .BYTE   361,0
000634'  043117  051506  052105             .ASCII  /OFFSET/
000642'    360     000                      .BYTE   360,0
000644'  020040  050123  047111             .ASCII  /  SPIN/
000652'    357     000                      .BYTE   357,0
000654'  047125  047514  042101             .ASCII  /UNLOAD/
000662'    356     000                      .BYTE   356,0
000664'  051040  041505  046101             .ASCII  / RECAL/
000672'    355     000                      .BYTE   355,0
000674'  040520  040513  045503             .ASCII  /PAKACK/
000702'    354     000                      .BYTE   354,0
000704'  040502  051504  041505             .ASCII  /BADSEC/
000712'    353     000                      .BYTE   353,0
000714'  020040  051040  046105             .ASCII  /   REL/
000722'    352     000                      .BYTE   352,0
000724'  043040  052115  031062             .ASCII  / FMT22/
000732'    347     000                      .BYTE   347,0
000734'  043040  052115  030062             .ASCII  / FMT20/
000742'    346     000                      .BYTE   346,0
000744'  020040  047440  042104             .ASCII  /   ODD/
000752'    345     000                      .BYTE   345,0
000754'  020040  053105  047105             .ASCII  /  EVEN/
000762'    344     000                      .BYTE   344,0
000764'  041040  044501  047117             .ASCII  / BAION/
000772'    337     000                      .BYTE   337,0
000774'  040502  047511  043106             .ASCII  /BAIOFF/
001002'    336     000                      .BYTE   336,0
001004'  041440  051117  047117             .ASCII  / CORON/
001012'    335     000                      .BYTE   335,0
001014'  047503  047522  043106             .ASCII  /COROFF/
001022'    334     000                      .BYTE   334,0

001024'  000376  001570            DVMVTE:  .WORD   376,MSFMT1-LOCZ        ;MODEL VECTOR TABLE EXTEN.
001030'  000375  001570                     .WORD   375,MSFMT1-LOCZ
001034'  000374  001570                     .WORD   374,MSFMT1-LOCZ
001040'  000373  001570                     .WORD   373,MSFMT1-LOCZ
```

```
001044' 000372  001570                  .WORD   372,MSFMT1-LOCZ
001050' 000371  001605                  .WORD   371,MSFMT5-LOCZ
001054' 000370  001571                  .WORD   370,MSFMT2-LOCZ
001060' 000367  001576                  .WORD   367,MSFMT3-LOCZ
001064' 000366  001605                  .WORD   366,MSFMT5-LOCZ
001070' 000365  001570                  .WORD   365,MSFMT1-LOCZ
001074' 000364  001570                  .WORD   364,MSFMT1-LOCZ
001100' 000363  001570                  .WORD   363,MSFMT1-LOCZ
001104' 000362  001605                  .WORD   362,MSFMT5-LOCZ
001110' 000361  001605                  .WORD   361,MSFMT5-LOCZ
001114' 000360  001605                  .WORD   360,MSFMT5-LOCZ
001120' 000357  001570                  .WORD   357,MSFMT1-LOCZ
001124' 000356  001570                  .WORD   356,MSFMT1-LOCZ
001130' 000355  001570                  .WORD   355,MSFMT1-LOCZ
001134' 000354  001570                  .WORD   354,MSFMT1-LOCZ
001140' 000353  001570                  .WORD   353,MSFMT1-LOCZ
001144' 000352  001570                  .WORD   352,MSFMT1-LOCZ
001150' 000347  001570                  .WORD   347,MSFMT1-LOCZ
001154' 000346  001570                  .WORD   346,MSFMT1-LOCZ
001160' 000345  001570                  .WORD   345,MSFMT1-LOCZ
001164' 000344  001570                  .WORD   344,MSFMT1-LOCZ
001170' 000337  001570                  .WORD   337,MSFMT1-LOCZ
001174' 000336  001570                  .WORD   336,MSFMT1-LOCZ
001200' 000335  001570                  .WORD   335,MSFMT1-LOCZ
001204' 000334  001570                  .WORD   334,MSFMT1-LOCZ


                        ;
                        ;       COMPILER TABLE EXTENSION
                        ;
001210'     003     376         DVCPTE: .BYTE   3,376                   ;NO WAIT
001212' 004537  000012                  .WORD   4537,10.
001216'     003     375                 .BYTE   3,375                   ;WAIT
001220' 004537  000012                  .WORD   4537,10.
001224'     004     374                 .BYTE   4,374                   ;STATUS
001226' 004537  000012  001002          .WORD   4537,10.,1002
001234'     004     373                 .BYTE   4,373                   ;COUNTS
001236' 004537  000012  001001          .WORD   4537,10.,1001
001244'     003     372                 .BYTE   3,372                   ;SEEK
001246' 004537  000012                  .WORD   4537,10.
001252'     004     371                 .BYTE   4,371                   ;SELECT DRIVE
001254' 004537  000012  000000          .WORD   4537,10.,0
001262'     006     370                 .BYTE   6,370                   ;WRITE CHECK DATA
001264' 004537  000012  000000          .WORD   4537,10.,0,2,2
001272' 000002  000002
001276'     005     367                 .BYTE   5,367                   ;READ HEADER
001300' 004537  000012  000000          .WORD   4537,10.,0,2
001306' 000002
001310'     006     366                 .BYTE   6,366                   ;WRITE HEADER
001312' 004537  000012  000204          .WORD   4537,10.,132.,2,2
001320' 000002  000002
001324'     003     365                 .BYTE   3,365                   ;CONTROL RESET
001326' 004537  000012                  .WORD   4537,10.
001332'     003     364                 .BYTE   3,364                   ;DRIVE RESET
001334' 004537  000012                  .WORD   4537,10.
001340'     003     363                 .BYTE   3,363                   ;SUBSYSTEM RESET
001342' 004537  000012                  .WORD   4537,10.
```

```
001346' 004 362                 .BYTE   4,362                   ;STEP UP
001350' 004537 000012 000000    .WORD   4537,10.,0
001356' 004 361                 .BYTE   4,361                   ;STEP DOWN
001360' 004537 000012 000000    .WORD   4537,10.,0
001366' 004 360                 .BYTE   4,360                   ;OFFSET
001370' 004537 000012 000000    .WORD   4537,10.,0
001376' 003 357                 .BYTE   3,357                   ;START SPINDLE
001400' 004537 000012           .WORD   4537,10.
001404' 003 356                 .BYTE   3,356                   ;UNLOAD
001406' 004537 000012           .WORD   4537,10.
001412' 003 355                 .BYTE   3,355                   ;RECALIBRATE
001414' 004537 000012           .WORD   4537,10.
001420' 003 354                 .BYTE   3,354                   ;PACK ACKNOWLEDGE
001422' 004537 000012           .WORD   4537,10.
001426' 003 353                 .BYTE   3,353                   ;PRINT LIST OF BAD SECTORS
001430' 004537 000012           .WORD   4537,10.
001434' 003 352                 .BYTE   3,352                   ;RELEASE
001436' 004537 000012           .WORD   4537,10.
001442' 003 347                 .BYTE   3,347                   ;FORMAT 22
001444' 004537 000012           .WORD   4537,10.
001450' 003 346                 .BYTE   3,346                   ;FORMAT 20
001452' 004537 000012           .WORD   4537,10.
001456' 003 345                 .BYTE   3,345                   ;ODD
001460' 004537 000012           .WORD   4537,10.
001464' 003 344                 .BYTE   3,344                   ;EVEN
001466' 004537 000012           .WORD   4537,10.
001472' 003 337                 .BYTE   3,337                   ;BAI ON
001474' 004537 000012           .WORD   4537,10.
001500' 003 336                 .BYTE   3,336                   ;BAI OFF
001502' 004537 000012           .WORD   4537,10.
001506' 003 335                 .BYTE   3,335                   ;CORRECTION ON
001510' 004537 000012           .WORD   4537,10.
001514' 003 334                 .BYTE   3,334                   ;CORRECTION OFF
001516' 004537 000012           .WORD   4537,10.


                        ;
                        ;       DEVICE INTERFACE WORD SYMBOL TABLE
                        ;
001522' 054503 020114   DVIWST: .ASCII  /CYL /
001526' 000004                  .WORD   DEVIW1
001530' 042510 042101           .ASCII  /HEAD/
001534' 000006                  .WORD   DEVIW2
001536' 042523 052103           .ASCII  /SECT/
001542' 000010                  .WORD   DEVIW3
001544' 052122 054522           .ASCII  /RTRY/
001550' 000012                  .WORD   DEVIW4
001552' 051515 040507           .ASCII  /MSGA/
001556' 000014                  .WORD   DEVIW5
001560' 051515 041107           .ASCII  /MSGB/
001564' 000016                  .WORD   DEVIW6
001566' 177777                  .WORD   177777                  ;END OF TABLE


                        ;
                        ;       MODEL STATEMENT TABLE EXTENSION
                        ;
001570'  000            MSFMT1: .BYTE   0                       ;NO OPERAND
```

```
001571' 377 052101 000377 MSFMT2: .ASCIZ  <377>/AT/<377>
001576' 044777 052116 177517 MSFMT3: .ASCIZ  <377>/INTO/<377>
001604' 000
001605' 377 000 MSFMT5: .BYTE  377,0  ;ONE OPERAND
001610'  .EVEN

;DEVICE ROUTINE CONSTANTS

001610'  MSKPST= .
001610'  .STAT= .  ;STORAGE FOR DEV REG'S AT INT
001610' 000000  ICS1:  .WORD  0  ;STORAGE FOR DISK REGISTERS
001612' 000000  IRKWC:  .WORD  0  ;  AS SAVED AT INTERRUPT TIME
001614' 000000  IRKBA:  .WORD  0
001616' 000000  .WORD  0
001620' 000000  ICS2:  .WORD  0
001622' 000000  IRKDS:  .WORD  0
001624' 000000  IRKER:  .WORD  0
001626' 000000  IRKAS:  .WORD  0
001630' 000000 000000  .WORD  0,0
001634' 000000  IRKDB:  .WORD  0
001636' 000000  .WORD  0
001640' 000000  IRKP0:  .WORD  0
001642' 000000  IRKPA:  .WORD  0
001644' 000000 000000  .WORD  0,0

001650' 000020  CSTAT:  .BLKW  16.  ;DEV REG CURRENT VALUES STORAGE
001710'  COUNTS:
001710' 000000  BYRD:  .WORD  0  ;BYTES READ COUNT
001712' 000000  .WORD  0
001714' 000000  BYWR:  .WORD  0  ;BYTES WRITTEN COUNT
001716' 000000  .WORD  0
001720' 000000  BYCK:  .WORD  0  ;BYTES CHECKED COUNT
001722' 000000  .WORD  0
001724' 000000  RDCNT:  .WORD  0  ;READ CMND COUNT
001726' 000000  WRCNT:  .WORD  0  ;WRITE CMND COUNT
001730' 000000  CKCNT:  .WORD  0  ;CHECK CMND COUNT
001732' 000000  SKCNT:  .WORD  0  ;SEEK/SEARCH CMD COUNT
001734' 000000  MISCNT:  .WORD  0  ;MISC. CMND COUNT
001736' 000000  ERRCNT:  .WORD  0  ;DEVICE ERRORS COUNT
001740' 000000  CECCER:  .WORD  0  ;CORRECTABLE ECC ERRORS
001742' 000000  DATAER:  .WORD  0  ;DATA/OPERATOR ERRORS COUNT
001744' 000000  DLTCNT:  .WORD  0  ;DATA LATE ERRORS
001746' 000000  DTECNT:  .WORD  0  ;DRIVE TIMING ERRORS
001750' 000000  HCRCNT:  .WORD  0  ;HEADER CRC ERRORS
001752' 000000  FERCNT:  .WORD  0  ;FORMAT ERRORS
001754' 000000  DCKCNT:  .WORD  0  ;DATA CHECK ERRORS
001756' 000000  WCECNT:  .WORD  0  ;WRITE CHECK ERRORS
001760' 000000  RETRYS:  .WORD  0  ;# OF RETRIES ON I/O CMNDS
001762' 000000  INTCNT:  .WORD  0  ;INTERRUPTS COUNT
001764'  CNTEND= .  ;END OF COUNT TABLE
000026  CNTNUM=CNTEND-COUNTS/2  ;NR OF COUNTERS

001764' 000000  ERRADR:  .WORD  0  ;CURR ADR IN USER PROG
001766' 000000  CNTADR:  .WORD  0  ;ADR OF BYTE COUNT TOTALS
```

N01


```
001770' 000000          CURFLG: .WORD   0               ;FLAG WORD OF CURR CMND
001772' 000000          CURCMD: .WORD   0               ;CURR CMND CODE
001774' 000000          CURADR: .WORD   0               ;CURR BUS ADDRESS
001776' 000000                  .WORD   0
002000' 000000          CURCNT: .WORD   0               ;NEG WORD CNT FOR CURR CMND
002002' 000000          CURPBC: .WORD   0               ;POSITIVE BYTE CNT FOR CURR CMND
002004' 000000          FINCNT: .WORD   0               ;FINAL WORD CNT (RPWC)
002006' 000000          CURRTY: .WORD   0               ;CURR RETRY COUNT
002010' 000000          RTRYIP: .WORD   0               ;RETRY IN PROGRESS FLAG
002012' 000000          CURPSW: .WORD   0               ;PSW STORAGE AREA
        002014'         HSKPEN= .

002014' 000000          RPCS1V: .WORD   0               ;BASE VALUE FOR RPCS1 REG
002016' 000000          RPCS2V: .WORD   0               ;BASE VALUE FOR RPCS2 REG
002020' 000000          RPMR1V: .WORD   0               ;BASE VALUE FOR RPMR1 REG
002022' 000000          DIAFLG: .WORD   0
002024' 000000          LUPCNT: .WORD   0               ;COUNT LOOPS
002026'                 PATCH:  .REPT   20.             ;PATCH AREA
                                .WORD   0
                                .ENDR
```

```
                                .SBTTL  RK06 SUPPORT ROUTINES ENTERED FROM MPG
                                ;
                                ;S U P P O R T   R O U T I N E S   E N T E R E D   F R O M   M P G
                                ;
                                ;DEVICE ROUTINE HOUSEKEEPING
                                ;       JSR     R5,HSKEEP       S/R CALL
                                ;       .WORD   0 OR 1          0 = DO HSKP PER OPSW
                                ;                               1 = UNCOND. DO HSKP
                                ;R2 = PROG'S OPSW
                                ;DESTROYS R0,R1
002076' 012767  000003  175706  HSKEEP: MOV     #3,RTRY         ;INIT # OF RETRY ATTEMPTS
002104' 005067  177704          CLR     RPCS1V          ;INITIALIZE RPCS1 VALUE
002110' 005067  177702          CLR     RPCS2V          ;INITIALIZE RPCS2 VALUE
002114' 005067  177700          CLR     RPMR1V          ;INITIALIZE RPMR1 VALUE
002120' 005067  177676          CLR     DIAFLG          ;INITIALIZE DIAGNOSTIC FLAGS
002124' 005[2]                  TST     (R5)+           ;CONDITIONALLY DO HSKP?
002126' 001003                  BNE     10$             ;N,Y-10$
002130' 032702  000004          BIT     #HSKPEP,R2      ;OPSW SPECIFY EACH PASS HSKP?
002134' 001010                  BNE     30$             ;Y,N-30$
002136' 010700          10$:    MOV     PC,R0           ;SET UP FIRST WD ADR
002140' 062700  177450          ADD     #HSKPST-.,R0
002144' 012701  000102          MOV     #HSKPEN-HSKPST/2,R1     ;SET UP # OF WORDS
002150' 005020          20$:    CLR     (R0)+           ;HSKP ALL NECESSARY AREAS
002152' 005301                  DEC     R1
002154' 001375                  BNE     20$
002156' 000205          30$:    RTS     R5              ;EXIT IN-LINE

                                ;RK06 REPORT ROUTINE
                                ;       JSR     R5,REPORT       S/R CALL
                                ;       .WORD   FLGWD           FLAGWORD
                                ;                                 BIT 15 = CMND MODE CALL
                                ;                                 BIT  9 = PROG STMNT CALL
                                ;                                 BIT  1 = DO STATUS REPORT
                                ;                                 BIT  0 = DO COUNTS REPORT
002160' 004067  010106  REPORT: JSR     R0,SAVREG       ;SAVE REG'S R0 - R5
002164' 004767  010134          JSR     PC,SUPTAD       ;SET UP PROG TBL ADR IN R3
002170' 011504                  MOV     (R5),R4         ;GET FLAGWORD
002172' 032704  000002          BIT     #2,R4           ;GOING TO DO STATUS DISPLAY?
002176' 001403                  BEQ     5$              ;Y,N-5$
002200' 004567  010140          JSR     R5,STSTAT       ;GO STORE STATUS REG'S
002204' 177444                  .WORD   CSTAT-.
002206' 032704  177776  5$:      BIT     #177776,R4      ;DISPLAYING CNTS AT END OF
002212' 001012                  BNE     15$             ;PROG PASS? (Y,N-15$)
002214' 010700                  MOV     PC,R0           ;SET UP ADR OF CNTS
002216' 062700  177472          ADD     #COUNTS-.,R0
002222' 012701  000026          MOV     #CNTNUM,R1      ;GET # OF CNT WORDS
002226' 005720          10$:    TST     (R0)+           ;THIS CNT WORD = 0?
002230' 001003                  BNE     15$             ;Y,N-15$
002232' 005301                  DEC     R1              ;DECR WORD CNT
```

```
                                                ;CK'ED ALL WORDS? (Y,N-10$)
                                                ;GO TO EXIT -- ALL CNTS ARE 0'S
                                                ;DISPLAY DEVICE I.D.
                                                ;DOING STATUS DISPLAY?
                                                ;Y,N-DISCNT
                                                ;SET UP ADR OF REG'S AT
                                                ;LAST INT
                                                ;SET UP # OF REG'S
                                                ;ALL REG'S = 0?
                                                ;N,Y-40$


                                                ;ISSUE 'AT LAST INT' MSG


                                                ;GO DISPLAY STATUS AT LAST INT

                                                ;ISSUE 'CURRENTLY' MSG


                                                ;GO DISPLAY CURRENT STATUS

                                                ;GO DISPLAY INFO WORDS
                                                ;DISPLAY COUNTS?
                                                ;Y,N-RPTEND
                                                ;SET UP # OF WORDS
                                                ;SET UP ADR OF CNTS

                                                ;SET UP TBL ADR

                                                ;MOV MSG ADR TO S/R LINKAGE
                                                ;SAVE ALL REG'S
                                                ;GET CURRENT COUNT
                                                ;CONVERT IT TO ASCII


                                                ;RESTORE REG'S
                                                ;POINT AT NXT CNT
                                                ;DONE ALL WORDS?
                                                ;Y,N-RPTLP
                                                ;GO ISSUE COUNTS MSG


                                                ;ISSUE "END OF REPORT" MSG


                                                ;RESTORE REGISTERS
                                                ;SET UP RETURN POINT
                                                ;EXIT IN-LINE


```

```
002460' 011124                          .WORD   BCMCK+6-RPTBAS
002462' 011151                          .WORD   CMDCRD-RPTBAS
002464' 011164                          .WORD   CMDCWR-RPTBAS
002466' 011177                          .WORD   CMDCCK-RPTBAS
002470' 011215                          .WORD   CMDCSK-RPTBAS
002472' 011232                          .WORD   CMDCMS-RPTBAS
002474' 011260                          .WORD   CNTERR-RPTBAS
002476' 011301                          .WORD   CNTCEC-RPTBAS
002500' 011322                          .WORD   CNTDER-RPTBAS
002502' 011350                          .WORD   CNTDLT-RPTBAS
002504' 011364                          .WORD   CNTDTE-RPTBAS
002506' 011401                          .WORD   CNTHCR-RPTBAS
002510' 011420                          .WORD   CNTFER-RPTBAS
002512' 011434                          .WORD   CNTDCK-RPTBAS
002514' 011453                          .WORD   CNTWCE-RPTBAS
002516' 011502                          .WORD   CNTRTY-RPTBAS
002520' 011530                          .WORD   CNTINT-RPTBAS


                                        ;TIMEOUT ERROR ROUTINE

                                                ;JSR    R5,TOUTER       S/R CALL

002522' 004067  007544          TOUTER: JSR     R0,SAVREG               ;SAVE ALL REGISTERS
002526' 004767  007572                  JSR     PC,SUPTAD               ;SET UP RPCS1 & PROG TBL ADR'S
002532' 032767  000400  175242          BIT     #SPINFL,DFLGWD          ;IS SPIN OR RECAL IN PROGRESS
002540' 001024                          BNE     CKSPIN                  ;YES
002542' 004567  007576                  JSR     R5,STSTAT               ;STORE CURRENT STATUS
002546' 177103                          .WORD   CSTAT-.
002550' 004567  006404                  JSR     R5,TVECT                ;CK IF I HAVE VECTOR CONTROL
002554' 000404                          BR      10$                     ;BR IF I DON'T
002556' 142714  000100                  BICB    #100,(R4)               ;RESET INT ENABLE
002562' 004767  006346                  JSR     PC,RINTV                ;RESET THE INTERRUPT VECTOR
002566' 042713  000010          10$:    BIC     #WT4IOT,(R3)            ;RESET WAITING FOR I/O FLG
002572' 004567  006420                  JSR     R5,ERRCS1               ;ISSUE I/O TIMEOUT ERROR MSG
002576' 003762                          .WORD   IOTO-ERMBAS
002600' 004567  007502                  JSR     R0,RESREG               ;RESTORE REGISTERS
002604' 012605                          MOV     (SP)+,R5                ;REMOVE RETURN ADR
002606' 000177  175236                  JMP     @CUPGER                 ;GO TO ERROR EXIT

002612' 032777  000002  175244  CKSPIN: BIT     #USMTPS,@CSYSFW         ;NEED TO USE MTPS INST?
002620' 001007                          BNE     32$                     ;N,Y-32$
002622' 113767  177776  177162          MOVB    @#PS,CURPSW             ;SAVE CURRENT PRIORITY
002630' 152737  000340  177776          BISB    #340,@#PS               ;SET PRIORITY TO 7
002636' 000404                          BR      34$                     ;GO SET UP UNIT #
002640' 106767  177146          32$:    MFPS    CURPSW                  ;SAVE CURR PRIORITY
002644' 106427  000340                  MTPS    #340                    ;SET PRIORITY TO 7
002650' 016400  000010          34$:    MOV     RPCS2(R4),R0            ;GET CURR UNIT #
002654' 010001                          MOV     R0,R1                   ;SAVE IT
002656' 042700  177747                  BIC     #177747,R0              ;RESET UNIT # & OTHER BITS
002662' 156300  000036                  BISB    PCURDV(R3),R0           ;SET IN MY UNIT #
002666' 110064  000010                  MOVB    R0,RPCS2(R4)            ;MOVE UNIT # TO RK611
002672' 005005                          CLR     R5
002674' 004767  010242                  JSR     PC,SD                   ;SELECT MY DRIVER
002700' 016400  000012                  MOV     RPDS(R4),R0             ;GET DRIVE'S STATUS
002704' 032700  000200                  BIT     #DRDY,R0                ;IS DRIVE BACK ON LINE?
```

```
002710' 001405                          BEQ     40$                     ;NO, WAIT LONGER
002712' 042713  000010          30$:    BIC     #WT4IOT,(R3)            ;RESET WAITING FOR I/O TERM FLAG
002716' 042767  000400  175056          BIC     #SPINFL,DFLGWD          ;RESET SPIN IN PROGRESS FLAG
002724' 012763  005670  000030  40$:     MOV     #3000.,PIOCNT(R3)       ;RESTORE 1 SECOND I/O COUNT
002732' 110164  000010                  MOVB    R1,RPCS2(R4)            ;RESTORE ORIG UNIT #
002736' 032777  000002  175120          BIT     #USMTPS,@CSYSFW         ;NEED TO USE MTPS INST'
002744' 001004                          BNE     50$                     ;N,Y-50$
002746' 116737  177040  177776          MOVB    CURPSW,@#PS             ;RESTORE ORIG PRIORITY
002754' 000402                          BR      60$                     ;GO TO EXIT
002756' 106467  177030          50$:    MTPS    CURPSW                  ;RESTORE ORIG PRIORITY
002762' 004067  007320          60$:    JSR     R0,RESREG               ;RESTORE REGISTERS
002766' 000205                          RTS     R5                      ;RETURN IN-LINE

                                ;KILL USER PROGRAM ROUTINE

                                        ;JSR    R5,KILL         S/R CALL
                                        ;R3 MUST CONTAIN PROG TBL ADR
                                        ;DESTROYS R0,R1

002770' 016701  175030          KILL:   MOV     DREGAD,R1               ;GET DEV REG ADR
002774' 004567  006160                  JSR     R5,IVECT                ;DO I HAVE VECTOR CONTROL'
003000' 000407                          BR      KILLEX                  ;BR IF I DON'T
003002' 132711  000100                  BITB    #100,(R1)               ;IS INT ENABLE SET'
003006' 001402                          BEQ     10$                     ;Y,N-10$
003010' 142711  000100                  BICB    #10C,(R1)               ;RESET INT ENABLE
003014' 004767  006114          10$:    JSR     PC,RINTV                ;RESET INT VECTOR INFO
003020' 000205                  KILLEX: RTS     R5                      ;EXIT IN-LINE
```

```
                                        .SBTTL  RK06 NON-I/O FUNCTION ROUTINES
                                ;
                                ;N O N - I N T E R R U P T   I / O   F U N C T I O N   R O U T I N E S
                                ;


                                        ;"STEPUP" FUNCTION ROUTINE

                                                ;JSR    R5,STEPUP       FUNCTION CALL
                                                ;.WORD  NBR             INCREMENT FACTOR

003022' 004767  000104          STEPUP: JSR     PC,STPCOM               ;DO COMMON SETUP
003026' 012567  000076                  MOV     (R5)+,STEIV             ;SAVE INCREMENT VALUE
003032' 004767  000252                  JSR     PC,UPSUB                ;GET NEW STARTING SECT,HEAD,CYL
003036' 010067  174742                  MOV     R0,CYL                  ;  AND SAVE THEM
003042' 010167  174740                  MOV     R1,HEAD
003046' 010267  174736                  MOV     R2,SECT
003052' 005767  000052                  TST     STEIV                   ;IS INCREMENT=0
003056' 001402                          BEQ     STPWRT                  ;YES
003060' 005367  000044                  DEC     STEIV                   ;GET NEW ENDING
003064' 004767  000220          STPWRT: JSR     PC,UPSUB                ;  VALUES
003070' 020067  174710                  CMP     R0,CYL                  ;IS ENDING CYL < START CYL
003074' 103406                          BLO     STE1                    ;YES, WRAP AROUND OCCURRED
003076' 020027  000632                  CMP     R0,#NRCYL-1             ;IS ENDING CYL = MAX CYL
003102' 001011                          BNE     STE2                    ;NO, EXIT
003104' 020127  000002                  CMP     R1,#NRHEAD-1            ;IS ENDING HEAD = MAX HEAD
003110' 001006                          BNE     STE2                    ;NO, EXIT
003112' 005067  174666          STE1:   CLR     CYL                     ;YES, FORCE WRAP AROUND
003116' 005067  174664                  CLR     HEAD                    ;  FOR CYL, HEAD, AND
003122' 005067  174662                  CLR     SECT                    ;    SECT
003126' 000205          STE2:   RTS     R5
003130' 000000          STEIV:  .WORD   0


003132' 016700  174646          STPCOM: MOV     CYL,R0                  ;GET CYL, HEAD, & SECT VALUES
003136' 016701  174644                  MOV     HEAD,R1                 ;IN REGISTERS
003142' 016702  174642                  MOV     SECT,R2
003146' 012704  000633                  MOV     #NRCYL,R4               ;GET MAX CYLINDER VALUE
003152' 012703  000024                  MOV     #20.,R3
003156' 032767  010000  176630          BIT     #CFMT,RPCS1V            ;ARE USING 20 SECTOR FORMAT?
003164' 001002                          BNE     STPCEX                  ;YES
003166' 012703  000026                  MOV     #22.,R3                 ;NO
003172' 000207          STPCEX: RTS     PC                              ;EXIT IN-LINE


                                        ;"STEPDN" FUNCTION ROUTINE

                                                ;JSR    R5,STEPDN       FUNCTION CALL
                                                ;.WORD  NBR             DECREMENT FACTOR

003174' 004767  177732          STEPDN: JSR     PC,STPCOM               ;DO COMMON SETUP
003200' 012567  177724                  MOV     (R5)+,STEIV             ;SAVE INCREMENT VALUE
003204' 004767  000146          STP2:   JSR     PC,DWNSUB               ;GET NEW STARTING SECT,HEAD,CYL
003210' 020067  174570                  CMP     R0,CYL                  ;IS ENDING GREATER THAN START
003214' 101014                          BHI     STP3                    ;YES, WRAP AROUND
003216' 010067  174562                  MOV     R0,CYL
```

```
                                                MOV     R1,HEAD
                                                MOV     R2,SECT
                                                CMP     R0,#NRCYL-1             ;IS ENDING CYL=MAX CYL
                                                BNE     STP1                    ;NO, EXIT
                                                CMP     R1,#NRHEAD-1            ;IS ENDING HEAD=MAX HEAD
                                                BNE     STP1                    ;NO, EXIT
                                        STP3:   MOV     #NPCYL-1,CYL            ;YES, FORCE WRAP-AROUND
                                                MOV     #NRHEAD-1,HEAD
                                                CLR     SECT
                                                TST     STEIV                   ;IS OFFSET=0
                                                BNE     STP31                   ;NO
                                                DEC     HEAD                    ;YES, CORRECT HEAD VALUE
                                        STP31:  JSR     PC,STPCOM
                                                BR      STP2                    ;START OVER FROM TOP DOWN
                                        STP1:   RTS     R5
                                        ;
                                        UPSUB:  ADD     STEIV,R2                ;GET NEW SECT
                                        UP4:    CMP     R2,R3                   ;IS SECT IN RANGE
                                                BLO     UP1                     ;YES
                                                SUB     R3,R2                   ;NO, REDUCE SECT
                                                INC     R1                      ;INCREMENT HEAD
                                                BR      UP4
                                        UP1:    CMP     R1,#NRHEAD              ;IS HEAD IN RANGE
                                                BLO     UP2                     ;YES
                                                SUB     #NRHEAD,R1              ;NO, REDUCE HEAD
                                                INC     R0                      ;INCREMENT CYL
                                                BR      UP1
                                        UP2:    CMP     R0,R4                   ;IS CYL IN RANGE
                                                BLO     UP3                     ;YES, EXIT
                                                SUB     R4,R0                   ;NO, REDUCE CYL
                                                BR      UP2
                                        UP3:    RTS     PC
                                        ;
                                        DWNSUB: SUB     STEIV,R2                ;GET NEW SECT
                                        DWN4:   CMP     R2,R3                   ;IS SECT IN RANGE
                                                BLO     DWN1                    ;YES
                                                ADD     R3,R2                   ;NO, INCREASE SECT
                                                DEC     R1                      ;DECREMENT HEAD
                                                BR      DWN4
                                        DWN1:   CMP     R1,#NRHEAD              ;IS HEAD IN RANGE
                                                BLO     DWN2                    ;YES
                                                ADD     #NRHEAD,R1              ;INCREASE HEAD
                                                DEC     R0                      ;DECREMENT CYL
                                                BR      DWN1
                                        DWN2:   CMP     R0,R4                   ;IS CYL IN RANGE?
                                                BLO     DWN3                    ;YES, EXIT
                                                ADD     R4,R0                   ;NO, INCREASE CYL
                                                BR      DWN2
                                        DWN3:   RTS     PC
                                        ;
                                        ;"WAIT" FUNCTION ROUTINE
                                        ;
                                                ;JSR    R5,WAIT          FUNCTION CALL
                                        ;
                                        WAIT:   BIC     #WAITMD,DFLGWD          ;RESET THE "NOWAIT" FLAG
```

```
003432' 004767 005232                  JSR     PC,CKDBSY               ;WAIT IF BUSY & DO TERMINATION
003436' 004767 005472                  JSR     PC,RINTV                ;RESET THE INTERRUPT VECTOR
003442' 000205                         RTS     R5                      ;EXIT IN-LINE


                                       ;"NOWAIT" FUNCTION ROUTINE

                                               ;JSR    R5,NOWAIT       FUNCTION CALL

003444' 052767 100000 174330 NOWAIT:   BIS     #WAITMD,DFLGWD          ;SET THE "NOWAIT" FLAG
003452' 000205                         RTS     R5                      ;EXIT IN-LINE


                                       ;"ODD" FUNCTION ROUTINE

                                               ;JSR    R5,ODD          FUNCTION CALL

003454' 042767 000020 176336 ODD:      BIC     #PAT,RPMR1V             ;SELECT ODD PARITY
003462' 000205                         RTS     R5                      ;EXIT IN-LINE
                                       ;"EVEN" FUNCTION ROUTINE

                                               ;JSR    R5,EVEN         FUNCTION CALL

003464' 052767 000020 176326 EVEN:     BIS     #PAT,RPMR1V             ;SELECT EVEN PARITY
003472' 000205                         RTS     R5                      ;EXIT IN-LINE


                                       ;"BAION" FUNCTION ROUTINE

                                               ;JSR    R5,BAION        FUNCTION CALL

003474' 052767 000020 176314 BAION:    BIS     #BAI,RPCS2V             ;SET THE BAI BIT
003502' 000205                         RTS     R5                      ;EXIT IN-LINE


                                       ;"BAIOFF" FUNCTION ROUTINE

                                               ;JSR    R5,BAIOFF       FUNCTION CALL

003504' 042767 000020 176304 BAIOFF:   BIC     #BAI,RPCS2V             ;RESET THE BAI BIT
003512' 000205                         RTS     R5                      ;EXIT IN-LINE
                                       ;"CORON" FUNCTION ROUTINE

                                               ;JSR    R5,CORON        FUNCTION CALL

003514' 042767 002000 174260 CORON:    BIC     #CORFLG,DFLGWD          ;RESET THE COR INH FLAG
003522' 000205                         RTS     R5                      ;EXIT IN-LINE


                                       ;"COROFF" FUNCTION ROUTINE

                                               ;JSR    R5,COROFF       FUNCTION CALL

003524' 052767 002000 174250 COROFF:   BIS     #CORFLG,DFLGWD          ;SET THE COR INH FLAG
003532' 000205                         RTS     R5                      ;EXIT IN-LINE
```

```
                                        ;"FMT22" FUNCTION ROUTINE
                                        ;
                                                ;JSR     R5,FMT22        FUNCTION CALL
                                        ;
003534' 042767  010000  176252  FMT22:  BIC     #CFMT,RPCS1V            ;SELECT 22 SECTORS
003542' 000205                          RTS     R5                      ;EXIT IN-LINE
                                        ;
                                        ;
                                        ;"FMT20" FUNCTION ROUTINE
                                        ;
                                                ;JSR     R5,FMT20        FUNCTION CALL
                                        ;
003544' 052767  010000  176242  FMT20:  BIS     #CFMT,RPCS1V            ;SELECT 20 SECTORS
003552' 000205                          RTS     R5                      ;EXIT IN-LINE
```

```
                                        .SBTTL  RK06 NON-INTERRUPT TYPE I/O FUNCTION ROUTINES

                                        ;"CRESET" FUNCTION ROUTINE

                                                ;JSR    R5,CRESET       FUNCTION CALL

003554' 004767  005110          CRESET: JSR     PC,CKDBSY               ;GO CK IF DEV IS BUSY
003560' 005267  176150                  INC     MISCNT                  ;ADD 1 TO MISC. CMND CNT
003564' 005067  174232                  CLR     ERRI                    ;RESET THE ERROR INDICATOR
003570' 005000                          CLR     R0                      ;INITIALIZE TIME OUT COUNT
003572' 052714  100000                  BIS     #CCLR,(R4)              ;SET THE CCLR BIT
003576' 032714  000200          CRE1:   BIT     #RDY,(R4)               ;IS CONTROLLER NOW READY?
003602' 001007                          BNE     20$                     ;YES
003604' 005300                          DEC     R0                      ;TIMED OUT?
003606' 100773                          BMI     CRE1                    ;NOT YET
003610' 004567  005374                  JSR     R5,ERRCS                ;GO ISSUE CRESET TIMEOUT ERROR
003614' 003013                          .WORD   CRTO-ERMBAS
003616' 000177  174226                  JMP     @CUPGER                 ;GO TO ERR RET POINT IN MPG
003622' 000205                  20$:    RTS     R5                      ;EXIT IN-LINE TO USER'S PROG


                                        ;"SRESET" FUNCTION ROUTINE

                                                ;JSR    R5,SRESET       FUNCTION CALL

003624' 004767  005040          SRESET: JSR     PC,CKDBSY               ;GO CK IF DEV IS BUSY
003630' 005267  176100                  INC     MISCNT                  ;COUNT MISC CMNDS
003634' 005067  174162                  CLR     ERRI                    ;CLEAR ERROR INDICATOR
003640' 005000                          CLR     R0
003642' 052764  000040  000010          BIS     #SCLR,RPCS2(R4)         ;PERFORM SRESET
003650' 004767  007306                  JSR     PC,DRVDLY
003654' 032714  000200          SRES1:  BIT     #RDY,(R4)               ;IS CONTROLLER READY?
003660' 001007                          BNE     SRES2                   ;YES
003662' 005300                          DEC     R0
003664' 100773                          BMI     SRES1
003666' 004567  005316                  JSR     R5,ERRCS                ;NO, ALARM AN ERROR
003672' 003031                          .WORD   SRTO-ERMBAS
003674' 000177  174150                  JMP     @CUPGER
003700' 000205                  SRES2:  RTS     R5


                                        ;"DRESET" FUNCTION ROUTINE

                                                ;JSR    R5,DRESET       FUNCTION CALL

003702' 012702  000005          DRESET: MOV     #DCCODE,R2              ;SETUP DRESET CMND CODE
003706' 000436                          BR      NOICOM                  ;GO TO NO INT CMND COM PROCESSING


                                        ;"UNLOAD" FUNCTION ROUTINE

                                                ;JSR    R5,UNLOAD       FUNCTION CALL

003710' 012702  000007          UNLOAD: MOV     #UCODE,R2               ;GET UNLOAD CODE
003714' 000433                          BR      NOICOM                  ;GO TO NO INT CMND COM PROCESSING
```

```
                                        ;"SPIN" FUNCTION ROUTINE
                                        ;
                                        ;JSR    R5,SPIN
                                        ;
003716' 012702  000011          SPIN:   MOV     #SSCODE,R2              ;GET SPIN CODE
003722' 000430                  SPIN1:  BR      NOICOM                  ;GO TO NO INT CMND COM PROCESSING
                                        ;
                                        ;"PAKACK" FUNCTION ROUTINE
                                        ;
                                        ;JSR    R5,PAKACK       FUNCTION CALL
                                        ;
003724' 012702  000003          PAKACK: MOV     #PACODE,R2              ;GET PACK ACK CODE
003730' 000425                  PAKAC1: BR      NOICOM                  ;GO TO NO INT CMND COM PROCESSING
                                        ;
                                        ;"REL" FUNCTION ROUTINE
                                        ;
                                        ;JSR    R5,REL          ;FUNCTION CALL
                                        ;
003732' 004767  004732          REL:    JSR     PC,CKDBSY               ;GO CK IF DEV IS BUSY
003736' 005267  175772                  INC     MISCNT                  ;COUNT MISC CMNDS
003742' 005067  174054                  CLR     ERRI                    ;CLEAR ERROR COUNT
003746' 005000                          CLR     R0
003750' 052764  000010  000010          BIS     #RLS,RPCS2(R4)          ;PERFORM REL
003756' 032714  000200          REL1:   BIT     #RDY,(R4)               ;IS CONTROLLER READY?
003762' 001007                          BNE     REL2                    ;YES
003764' 005300                          DEC     R0
003766' 100773                          BMI     REL1
003770' 004567  005214                  JSR     R5,ERRCS                ;NO, ALARM ERROR
003774' 003047                          .WORD   RTO-ERMBAS
003776' 000177  174046                  JMP     @CUPGER
004002' 000205                  REL2:   RTS     R5
                                        ;
                                        ;NON INTERRUPT I/O FUNCTION COMMON PROCESSING
                                        ;
                                        ;R5 = USER PROGRAM RETURN ADR
                                        ;R4 = RPCS1 ADR
                                        ;R3 = PROG TBL ADR
                                        ;R2 = FUNCTION'S COMMAND CODE
                                        ;
004004' 004767  004660          NOICOM: JSR     PC,CKDBSY               ;GO CK IF DEV IS BUSY
004010' 005267  175720                  INC     MISCNT                  ;ADD 1 TO MISC. CMND CNT
004014' 004767  000174          NOI1:   JSR     PC,ACQHSK               ;HOUSEKEEP THE DISK
004020' 032764  000100  000012          BIT     #VV,RPDS(R4)            ;IS THE VV BIT SET?
004026' 001004                          BNE     20$                     ;N,Y-20$
004030' 012714  000003                  MOV     #PACODE,(R4)            ;ISSUE THE PACK ACKNOWLEDGE CMND
004034' 004767  007122                  JSR     PC,DRVDLY
004040' 056702  175750          20$:    BIS     RPCS1V,R2               ;MERGE CFMT/CDT
004044' 052767  000300  173730          BIS     #CMDISU+ANYIOI,DFLGWD   ;SET CMND ISSUED FLAGS
004052' 010214                          MOV     R2,(R4)                 ;ISSUE SPECIFIED CMND
004054' 012700  000024                  MOV     #20.,R0                 ;SET UP DELAY CNT
004060' 005300                  30$:    DEC     R0                      ;DELAY FOR A FEW MICROSECONDS
004062' 001376                          BNE     30$
004064' 004767  007072                  JSR     PC,DRVDLY
004070' 032764  100000  000012          BIT     #SVAL,RPDS(R4)          ;IS RKDS VALID
004076' 001002                          BNE     14$                     ;YES
004100' 004767  007056                  JSR     PC,DRVDLY               ;NO, DELAY
```

L02


```
1096  004104' 032764 000001 000012 14$:   BIT     #DRA,RPDS(R4)              ;IS DRIVE AVAILABLE
1097  004112' 001010                       BNE     35$                        ;YES
1098  004114' 042767 000300 173660        BIC     #CMDISU!ONYIOI,DFLGWD      ;NO, WAIT FOR IT
1099  004122' 004577 173720               JSR     R5,@CIOBSY                 ;ALLOW OTHERS TO RUN
1100  004126' 004767 007116               JSR     PC,DRVCLR                  ;DO DRIVE CLEAR
1101  004132' 000730                       BR      NOI1                       ;REPEAT COMMAND
1102  004134' 005714               35$:    TST     (R4)                       ;IS 'CERR' ERROR BIT SET?
1103  004136' 100005                       BPL     50$                        ;Y,N-50$
1104  004140' 004567 005044               JSR     R5,ERRCS                   ;REPORT NON-INT TERM ERROR
1105  004144' 003101                       .WORD   NOITER-ERMBAS
1106  004146' 000177 173676               JMP     @CUPGER                    ;GO TO MPG'S ERROR RETURN POINT
1107  004152' 120227 000011        50$:    CMPB    R2,#SSCODE                 ;IS THIS THE SPIN COMMAND
1108  004156' 001015                       BNE     60$                        ;NO
1109  004160' 052767 000400 173614        BIS     #SPINFL,DFLGWD             ;SET SPIN IN PROGRESS FLAG
1110  004166' 012763 005670 000030        MOV     #3000.,PTOCNT(R3)          ;INITIALIZE 1 SECOND T/O CNT
1111  004174' 005767 173602               TST     DFLGWD                     ;"NOWAIT" FLAG SET?
1112  004200' 100404                       BMI     60$                        ;N,Y-60$
1113  004202' 052713 000010               BIS     #WT4IOT,(R3)               ;SET WAITING FOR I/O TERM FLAG
1114  004206' 004577 173634               JSR     R5,@CIOBSY                 ;RELEASE CNTRL UNTIL UNIT IS ON-LINE
1115  004212' 000205               60$:    RTS     R5                         ;EXIT TO USER PROG
```

```
                                                ;HOUSEKEEP DISK
                                                ;
                                                ;JSR    PC,ACQHSK       S/R CALL
                                                ;
                                                ;R5 = ADR AFTER USER PROG JSR
                                                ;R4 = RPCS1 ADR
                                                ;R3 = PROG TBL ADR
                                                ;
                                                ;DESTROYS R0
                                                ;
004214' 010146                  ACQHSK: MOV     R1,-(SP)                ;SAVE R1
004216' 042767  000561  173556  ACQRTY: BIC     #CMDISU!SPINFL!SWOVTO!SWDIER!IOERR,DFLGWD       ;HSKP FLAG BITS
004224' 012763  072460  000030          MOV     #ITIME,PTOCNT(R3)       ;SETUP INT TIMER
004232' 116300  000035                  MOVB    PCURDV(R3),R0           ;GET MY UNIT #
004236' 020027  000007                  CMP     R0,#7                   ;VALID UNIT #?
004242' 101405                          BLOS    10$                     ;N,Y-10$
004244' 005267  175472                  INC     DATAER                  ;ADJ ERROR COUNTS
004250' 005367  175462                  DEC     ERRCNT
004254' 000427                          BR      ACQERR                  ;GO REPORT THE ERROR
004256' 010001                  10$:    MOV     R0,R1                   ;GET DISPLACEMENT INTO
004260' 006301                          ASL     R1                      ;THE ATA TABLE FOR
004262' 060701                          ADD     PC,R1                   ;THIS UNIT #
004264' 062701  000100                  ADD     #ATATBL-.,R1
004270' 112167  000110                  MOVB    (PC)+,MYATA             ;STORE ATA BIT MASKS FOR
004274' 111167  000106                  MOVB    (R1),OTHATA             ;THIS UNIT #
004300' 056700  175512                  BIS     RPCS2V,R0               ;SET BAI BIT IN UNIT #
004304' 010064  000010                  MOV     R0,RPCS2(R4)            ;SETUP RKCS2
004310' 016764  175504  000026          MOV     RPMR1V,RPMR1(R4)        ;  AND RPMR1
004316' 010501                          MOV     R5,R1                   ;SELECT DRIVE
004320' 005005                          CLR     R5
004322' 004767  006614                  JSR     PC,SD
004326' 010105                          MOV     R1,R5
004330' 012601                          MOV     (SP)+,R1                ;RESTORE R1
004332' 000207                          RTS     PC                      ;EXIT IN-LINE
004334' 010146                  ACQERR: MOV     R1,-(SP)                ;SAVE R1 & R2
004336' 010246                          MOV     R2,-(SP)
004340' 004567  004644                  JSR     R5,ERRCS                ;STORE CURR STATUS & REPORT
004344' 003562                          .WORD   INVDVN-ERMBAS           ;INVALID UNIT NR
004346' 012602                          MOV     (SP)+,R2                ;RESTORE R1 & R2
004350' 012601                          MOV     (SP)+,R1
004352' 004577  173472                  JSR     R5,@CUPGER              ;GO TO MPG'S ERROR RETURN POINT
004356' 004767  004306                  JSR     PC,CKDBSY               ;CHECK IF DISK IS BUSY
004362' 000715                          BR      ACQRTY
                                        
004364'     001     376         ATATBL: .BYTE   001,376
004366'     002     375                 .BYTE   002,375
004370'     004     373                 .BYTE   004,373
004372'     010     367                 .BYTE   010,367
004374'     020     357                 .BYTE   020,357
004376'     040     337                 .BYTE   040,337
004400'     100     277                 .BYTE   100,277
004402'     200     177                 .BYTE   200,177
                                        
004404' 000000          MYATA:  .WORD   0
004406' 000000          OTHATA: .WORD   0
```

```
                                        .SBTTL  RK06 INTERRUPT TYPE I/O FUNCTION ROUTINES
                                ;
                                ;I N T E R R U P T   T Y P E   I / O   F U N C T I O N   R O U T I N E S
                                ;


                                        ;"READ" FUNCTION ROUTINE

                                                ;JSR    R5,READ         FUNCTION CALL
                                                ;.WORD  ADR             DATA ADDRESS (BITS 16 - 21)
                                                ;.WORD  ADR             DATA ADDRESS (BITS 0 - 15)
                                                ;.WORD  CNT             BYTE COUNT
                                                ;.WORD  DEV             (NOT USED)

004410' 012702  000121          READ:   MOV     #RCODE!IE,R2            ;SETUP READ CODE
004414' 012701  000635                  MOV     #635,R1                 ;SET UP CMND FLAG WORD
004420' 004767  004244          RDCOM:  JSR     PC,CKDBSY               ;GO CK IF DEV IS BUSY
004424' 005267  175274                  INC     RDCNT                   ;ADD 1 TO READ CMND CNT
004430' 010700                          MOV     PC,R0                   ;SET UP ADR. OF BYTES READ CNT
004432' 062700  175260                  ADD     #BYRD+2-.,R0
004436' 000167  000454                  JMP     CMDCOM                  ;GO TO CMND COMMON PROCESSING


                                        ;"WRITE" FUNCTION ROUTINE

                                                ;JSR    R5,WRITE        FUNCTION CALL
                                                ;.WORD  ADR             DATA ADDRESS (BITS 16 - 21)
                                                ;.WORD  ADR             DATA ADDRESS (BITS 0 - 15)
                                                ;.WORD  CNT             BYTE COUNT
                                                ;.WORD  DEV             (NOT USED)

004442' 012702  000123          WRITE:  MOV     #WCODE!IE,R2            ;SETUP WRITE CODE
004446' 012701  000235                  MOV     #235,R1                 ;SET UP CMND FLAG WORD
004452' 004767  004212          WRCOM:  JSR     PC,CKDBSY               ;GO CK IF DEV IS BUSY
004456' 005267  175244                  INC     WRCNT                   ;ADD 1 TO WRITE CMND CNT
004462' 010700                          MOV     PC,R0                   ;SET UP ADR OF BYTES WRITTEN CNT
004464' 062700  175232                  ADD     #BYWR+2-.,R0
004470' 004067  005576                  JSR     R0,SAVREG               ;SAVE REGS
004474' 016746  173304                  MOV     CYL,-(SP)               ;SAVE STARTING DISK ADDR
004500' 016746  173302                  MOV     HEAD,-(SP)
004504' 016746  173300                  MOV     SECT,-(SP)
004510' 016567  000004  176412          MOV     4(R5),STEIV             ;GET SIZE OF XFER TO WRITE
004516' 006267  176406                  ASR     STEIV                   ;DIVIDE BY 512 BYTES/SECTOR
004522' 042767  100000  176400          BIC     #100000,STEIV           ;  TO GET NR SECTORS
004530' 012746  000010                  MOV     #8.,-(SP)
004534' 006267  176370          WR3:    ASR     STEIV
004540' 005316                          DEC     (SP)
004542' 001374                          BNE     WR3
004544' 005726                          TST     (SP)+
004546' 032765  000777  000004          BIT     #777,4(R5)              ;IS THERE A PARTIAL SECTOR
004554' 001402                          BEQ     WR1                     ;NO
004556' 005267  176346                  INC     STEIV                   ;YES, USE ONE MORE SECT
004562' 004767  176344          WR1:    JSR     PC,STPCOM               ;WILL THIS XFER DAMAGE LAST TRACK
004566' 005367  176336                  DEC     STEIV
004572' 004567  176266                  JSR     R5,STPWRT                   ;USE STEPUP TO GET ENDING ADDR
004576' 026667  000004  173200          CMP     4(SP),CYL                   ;DID CYL CHANGE
```

```
1229  004604' 001021                  BNE     HR2                     ;YES, ALARM
1230  004606' 026667  000002  173172  CMP     2(SP),HEAD                  ;DID HEAD CHANGE
1231  004614' 001015                  BNE     HR2                     ;YES, ALARM
1232  004616' 021667  173166          CMP     (SP),SECT                   ;DID SECT CHANGE
1233  004622' 001012                  BNE     HR2                     ;YES, ALARM
1234  004624' 012667  173160          MOV     (SP)+,SECT              ;NO, PERFORM XFER
1235  004630' 012667  173152          MOV     (SP)+,HEAD              ;  AFTER RESTORE ADDR
1236  004634' 012667  173144          MOV     (SP)+,CYL               ;  AND REGISTERS
1237  004640' 004067  005442          JSR     R0,RESREG
1238  004644' 000167  000246          JMP     CMDCOM                  ;GO TO COMMON
1239  004650' 012667  173134  HR2:    MOV     (SP)+,SECT              ;GET BAD ADDRESS AND
1240  004654' 012667  173126          MOV     (SP)+,HEAD
1241  004660' 012667  173120          MOV     (SP)+,CYL
1242  004664' 062706  000014          ADD     #12.,SP                 ;  CORRECT STACK
1243  004670' 004567  004314          JSR     R5,ERRCS                ;PRINT HAVE BAD ADDRESS
1244  004674' 003421                  .WORD   PAMBA-ERMBAS
1245  004676' 000177  173146          JMP     @CUPGER                 ;GO TO MPG ERROR RTN
1246
1247
1248                                  ;"RDHD" FUNCTION ROUTINE
1249
1250                                          ;JSR    R5,RDHD         FUNCTION CALL
1251                                          ;.WORD  ADR             DATA ADDRESS (BITS 16 - 21)
1252                                          ;.WORD  CNT             BYTE COUNT
1253
1254  004702' 012702  000125  RDHD:   MOV     #RHCODE!IE,R2           ;SETUP READ-HEADER CODE
1255  004706' 012701  001626          MOV     #1626,R1                ;SET UP CMND FLAG WORD
1256  004712' 000642                  BR      RDCOM                   ;GO TO COMMON READ PROCESSING
```

```
                                        ;"WRHD" FUNCTION ROUTINE

                                                ;JSR     R5,WRHD         FUNCTION CALL
                                                ;.WORD   ADR             DATA ADDRESS (BITS 16 - 21)
                                                ;.WORD   ADR             DATA ADDRESS (BITS 0 - 15)
                                                ;.WORD   CNT             BYTE COUNT

004714' 012702  000127          WRHD:   MOV     #WHCODE!IE,R2           ;SETUP WRITE-HEADER CODE
004720' 012701  000236                  MOV     #236,R1                 ;SET UP CMND FLAG WORD
004724' 000652                          BR      WRCOM                   ;GO TO WRITE COMMON PROCESSING


                                        ;"WRCK" FUNCTION ROUTINE

                                                ;JSR     R5,WRCK         FUNCTION CALL
                                                ;.WORD   ADR             DATA ADDRESS (BITS 16 - 21)
                                                ;.WORD   ADR             DATA ADDRESS (BITS 0 - 15)
                                                ;.WORD   CNT             BYTE COUNT

004726' 012702  000131          WRCK:   MOV     #WCCODE!IE,R2           ;SETUP WRITE-CHECK CODE
004732' 012701  000236                  MOV     #236,R1                 ;SET UP CMND FLAG WORD
004736' 004767  003726                  JSR     PC,CKDBSY               ;GO CK IF DEV IS BUSY
004742' 005267  174762                  INC     CKCNT                   ;ADD 1 TO CHECK CMND COUNT
004746' 010700                          MOV     PC,R0                   ;SET UP ADR OF BYTES
004750' 062700  174752                  ADD     #BYCK+2-.,R0            ;CHECKED COUNT
004754' 000460                          BR      CMDCOM                  ;GO TO CMND COM PROCESSING


                                        ;"SEEK" FUNCTION ROUTINE

                                                ;JSR     R5,SEEK         FUNCTION CALL

004756' 012702  000117          SEEK:   MOV     #SCODE!IE,R2            ;SETUP SEEK CODE
004762' 012701  002060                  MOV     #2060,R1                ;SET UP CMND FLAG WORD
004766' 004767  003676          SKCOM:  JSR     PC,CKDBSY               ;GO CK IF DEV IS BUSY
004772' 005267  174734                  INC     SKCNT                   ;ADD 1 TO SEEK CMND COUNT
004776' 000447                          BR      CMDCOM                  ;GO TO CMND COMMON PROCESSING


                                        ;"OFFSET" FUNCTION ROUTINE

                                                ;JSR     R5,OFFSET       FUNCTION CALL
                                                ;.WORD   VALUE           BINARY OFFSET VALUE

005000' 012702  000115          OFFSET: MOV     #OCODE!IE,R2            ;SETUP OFFSET CODE
005004' 012701  002140                  MOV     #2140,R1                ;SET UP CMND FLAG WORD
005010' 000766                          BR      SKCOM                   ;GO TO SEEK CMND COM PROCESSING


                                        ;"RECAL" FUNCTION ROUTINE

                                                ;JSR     R5,RECAL        FUNCTION CALL

005012' 012702  000113          RECAL:  MOV     #RCCODE!IE,R2           ;SETUP RECALIBRATE CODE
005016' 012701  002040                  MOV     #2040,R1                ;SET UP CMND FLAG WORD
005022' 000761                          BR      SKCOM                   ;GO TO SEEK CMND COM PROCESSING


                                        ;"SELDRI" FUNCTION ROUTINE
```

D03


```
                                                        ;JSR    R5,SELDRI
                                                        ;.WORD  0       CODE TO SELECT MESSAGE FROM DRIVE

005024' 004767 005274          SELDRI: JSR     PC,SUPTAD               ;SETUP R3 & R4
005030' 004767 003634                  JSR     PC,CKDBSY               ;CHECK IF DISK IS BUSY
005034' 004767 177154                  JSR     PC,QCHSK                ;HOW SECKEEP DISK
005040' 005267 174670                  INC     MISCNT                  ;COUNT MISC COMMANDS
005044' 012546                         MOV     (R5)+,-(SP)             ;GET CODE
005046' 100416                         BMI     SEL1                    ;CODE IS BAD
005050' 021627 000003                  CMP     (SP),#SDMAX             ;IS CODE IN RANGE?
005054' 101013                         BHI     SEL1                    ;NO
005056' 010502                         MOV     R5,R2                   ;YES, SAVE RETURN ADDRESS
005060' 012605                         MOV     (SP)+,R5                ;GET CODE AND SELECT
005062' 004767 006054                  JSR     PC,SD
005066' 016467 000034 172720           MOV     RPMR2(R4),MSGA          ;MSG A
005074' 016467 000036 172714           MOV     RPMR3(R4),MSGB          ;MSG B
005102' 000202                         RTS     R2
005104' 004567 004106          SEL1:   JSR     R5,ERRCS1               ;PRINT INVALID MSG-SEL
005110' 003575                         .WORD ISDMC-ERMBAS
005112' 000177 172732                  JMP     @CUPGER                 ;GO TO MPG'S ERROR RTN
```

```
                                        ;INTERRUPT TYPE I/O FUNCTION COMMON PROCESSING ROUTINE

                                                ;R4 = ADR OF RPCS1 DEV REG
                                                ;R3 = PROG TBL ADR
                                                ;R2 = COMMAND CODE
                                                ;R1 = COMMAND FLAG WORD
                                                ;R0 = ADR OF BYTE COUNT, IF APPLICABLE


                                                ;CMND FLAGWORD FORMAT:

                                                        ;BIT 12 = 10000 = WRHD COMMAND
                                                        ;BIT 11 = 4000 = RECALIBRATE COMMAND
                                                        ;BIT 10 = 2000 = TWO INTERRUPTS EXPECTED
                                                        ;BIT 9 = 1000 = RDHD COMMAND
                                                        ;BIT 8 = 400 = DO ECC CORRECTION IF ALLOWED
                                                        ;BIT 7 = 200 = PERFORM RETRIES ON CMND
                                                        ;BIT 6 = 100 = OFFSET COMMAND
                                                        ;BIT 5 = 040 = CMND TERMINATES WITH ATA
                                                        ;BIT 4 = 020 = SET UP CYL/HD/SECT #
                                                        ;BIT 3 = 010 = INCREMENT BYTE COUNTS
                                                        ;BIT 2 = 004 = DATA TRANSFER CMND
                                                        ;BIT 1 = 002 = 3 ARGUMENT CMND
                                                        ;BIT 0 = 001 = 4 ARGUMENT CMND


005116' 010067  174644          CMDCOM: MOV     R0,CNTADR               ;SAVE ADR OF BYTE COUNT
005122' 004767  177066          CMD1:   JSR     PC,ACQHSK               ;HOUSEKEEP THE DISK
005126' 010167  174636                  MOV     R1,CURFLG               ;SAVE FLAGWD FOR TERMINATION
005132' 056702  174656                  BIS     RPCS1V,R2               ;MERGE CFMT
005136' 010267  174630                  MOV     R2,CURCMD               ;SAVE CURR CMND CODE
005142' 032701  000003                  BIT     #3,R1                   ;THIS CMND HAVE BUS ADR & WD CNT?
005146' 001440                          BEQ     10$                     ;Y,N-10$
005150' 012567  174620                  MOV     (R5)+,CURADR            ;STORE 2 WORD BUS ADR
005154' 032701  001000                  BIT     #1000,R1                ;IS THIS A RDHD CMND
005160' 001002                          BNE     5$                      ;YES
005162' 012567  174610                  MOV     (R5)+,CURADR+2
005166' 012500                  5$:     MOV     (R5)+,R0                ;GET BYTE COUNT
005170' 032701  010000                  BIT     #10000,R1               ;IS THIS A WRHD CMND
005174' 001405                          BEQ     7$                      ;NO
005176' 032702  010000                  BIT     #CFMT,R2                ;SHOULD USE 20 SECTORS
005202' 001402                          BEQ     7$                      ;NO
005204' 012700  000170                  MOV     #120.,R0                ;YES, GET NR BYTES OF HEADERS
005210' 032701  001000          7$:     BIT     #1000,R1                ;IS THIS A RDHD
005214' 001002                          BNE     8$                      ;YES
005216' 042700  000001                  BIC     #1,R0                   ;MAKE SURE ITS EVEN
005222' 010067  174554          8$:     MOV     R0,CURPBC               ;SAVE POSITIVE BYTE CNT
005226' 000241                          CLC                             ;MAKE IT A WORD COUNT
005230' 006000                          ROR     R0
005232' 005400                          NEG     R0                      ;MAKE IT NEGATIVE
005234' 010067  174540                  MOV     R0,CURCNT               ;SAVE IT
005240' 032701  000001                  BIT     #1,R1                   ;THERE A 4TH WORD?
005244' 001401                          BEQ     10$                     ;Y,N-10$
005246' 005725                          TST     (R5)+                   ;BYPASS IT
005250' 032764  000100  000012  10$:    BIT     #VV,RPOS(R4)            ;IS THE VV BIT SET?
005256' 001004                          BNE     20$                     ;N,Y-20$
```

```
1391  005260' 012714 000003              MOV     #PACODE,(R4)            ;ISSUE PACK ACKNOWLEDGE CMND
1392  005264' 004767 005672              JSR     PC,DRVDLY
1393  005270' 004767 000176        20$:  JSR     PC,SUIORG               ;GO SET UP REGS FOR I/O
1394  005274' 032701 000100              BIT     #100,R1                 ;THIS THE OFFSET CMND?
1395  005300' 001403                     BEQ     30$                     ;Y,N-30$
1396  005302' 012500                     MOV     (R5)+,R0                ;RETRIEVE OFFSET VALUE
1397  005304' 010064 000016              MOV     R0,RPAS(R4)             ;SETUP OFFSET
1398  005310' 016767 172476 174470 30$:  MOV     RTRY,CURRTY             ;INITIALIZE RETRY COUNT
1399  005316' 005067 174466              CLR     RTRYIP                  ;CLEAR RETRY IN PROGRESS FLAG
1400  005322' 012767 003111 003552       MOV     #IOTERM-ERRBAS,INTEAD   ;INIT TERMINATION ERROR MSG
1401  005330' 052713 000010              BIS     #WT4IOT,(R3)            ;SET WAITING FOR I/O TERM FLAG
1402  005334' 052767 000302 172440       BIS     #CMDISU!ANYIOI!DOTERM,DFLGWD    ;SET CMD ISSUED FLAGS
1403  005342' 010214                     MOV     R2,(R4)                 ;ISSUE SPECIFIED CMND
1404  005344' 004767 005612              JSR     PC,DRVDLY               ;WAIT FOR DRIVE RESPONSE
1405  005350' 032767 000001 174444       BIT     #DRIB,DIAFLG            ;IS DRIVE BUSY?
1406  005356' 001413                     BEQ     60$                     ;NO
1407  005360' 042767 000001 174434       BIC     #DRIB,DIAFLG            ;YES, REMOVE COMMAND FROM MPG
1408  005366' 042767 000302 172406       BIC     #CMDISU!ANYIOI!DOTERM,DFLGWD
1409  005374' 004577 172446              JSR     R5,@CIOBSY              ;ALLOW OTHERS TO RUN
1410  005400' 004767 005644              JSR     PC,DRVCLR               ;DO DRIVE CLEAR
1411  005404' 000646                     BR      CMD1                    ;REPEAT COMMAND
1412  005406' 032701 004000        60$:  BIT     #4000,R1                ;IS THIS A RECAL CMND
1413  005412' 001406                     BEQ     61$                     ;NO
1414  005414' 052767 000400 172360       BIS     #SPINFL,DFLGWD          ;YES, FLAG WAIT FOR RDY
1415  005422' 012763 005670 000030       MOV     #3000.,PTOCNT(R3)       ;SETUP INT TIMER
1416  005430' 005767 172346        61$:  TST     DFLGWD                  ;"NOWAIT" BIT SET?
1417  005434' 100011                     BPL     40$                     ;Y,N-40$
1418  005436' 042713 000010              BIC     #WT4IOT,(R3)            ;RESET WAITING FOR I/O TERM
1419  005442' 032701 004000              BIT     #4000,R1                ;IS THIS THE RECAL CMND
1420  005446' 001410                     BEQ     50$                     ;NO
1421  005450' 042767 000400 172324       BIC     #SPINFL,DFLGWD          ;YES
1422  005456' 000404                     BR      50$                     ;GO TO EXIT
1423  005460' 004577 172362        40$:  JSR     R5,@CIOBSY              ;WAIT FOR I/O TO COMPLETE
1424  005464' 004767 003316              JSR     PC,PROCTM               ;GO PROCESS TERMINATION
1425  005470' 000205               50$:  RTS     R5                      ;EXIT IN-LINE TO USER PROG
1426
1427
1428
1429
1430                                     ;SET UP DEVICE REGS FOR I/O
1431
1432                                             ;JSR    PC,SUIORG       S/R CALL
1433                                             ;
1434                                             ;R4 = RPCS1 ADR
1435                                             ;R3 = PROG TBL ADR
1436                                             ;R2 = CMND CODE
1437                                             ;R1 = CMND FLAGWORD
1438                                             ;
1439                                             ;DESTROYS R0
1440
1441  005472' 032701 000020        SUIORG: BIT     #20,R1                  ;NEED TO SET UP CYL/HEAD/SECT?
1442  005476' 001412                     BEQ     10$                     ;Y,N-10$
1443  005500' 016764 172300 000020       MOV     CYL,RPDC(R4)            ;LOAD CYL #
1444  005506' 016746 172274              MOV     HEAD,-(SP)              ;GET HEAD #
1445  005512' 000316                     SWAB    (SP)                    ;PUT IN CORRECT BIT POSITION
1446  005514' 116716 172270              MOVB    SECT,(SP)               ;SET IN SECT #
```

```
 1447  005520' 012664 000006          MOV     (SP)+,RPDA(R4)          ;LOAD HEAD & SECT #'S
 1448  005524' 032701 000004      10$: BIT     #4,R1                   ;DATA XFER CMND?
 1449  005530' 001414                  BEQ     30$                     ;Y,N-30$
 1450  005532' 016700 174236          MOV     CURADR,R0               ;GET HIGH BITS OF ADR
 1451  005536' 042700 177774          BIC     #177774,R0              ;RESET BITS ABOVE A17
 1452  005542' 000300                  SWAB    R0                      ;ALIGN BITS A16 & A17
 1453  005544' 050002                  BIS     R0,R2                   ;SET THEM INTO CMND CODE WORD
 1454  005546' 016764 174224 000004   MOV     CURADR+2,RPBA(R4)       ;LOAD BITS 0-15 OF ADR
 1455  005554' 016764 174220 000002   MOV     CURCNT,RPWC(R4)         ;LOAD WORD COUNT
 1456  005562' 000207              30$: RTS     PC                      ;EXIT IN-LINE
```

```
                                        ;"BADSEC" FUNCTION ROUTINE
                                        ;
                                        ;JSR    R5,BADSEC
005564' 004767 003100          BADSEC: JSR     PC,CKDBSY               ;CHECK IF DISK IS BUSY
005570' 005267 174140                  INC     MISCNT
005574' 004767 176414                  JSR     PC,ACQMSK               ;HOUSEKEEP DISK
005600' 010700                         MOV     PC,R0                   ;GET ADDR OF BYRD
0056?2' 062700 174110                  ADD     #BYRD+2-.,R0
0056?6' 010067 174154                  MOV     R0,CNTADR
005612' 016702 174176                  MOV     RPCS1V,R2               ;SETUP RKCS1
005616' 052702 000121                  BIS     #RCODE!IE,R2            ;  WITH READ COMMAND
005622' 010267 174144                  MOV     R2,CURCMD               ;SAVE COMMAND
0056?6' 016300 000020                  MOV     PRDIOA(R3),R0           ;IS R0IO LARGE ENOUGH
0057?2' 016301 000016                  MOV     PRDIOA(R3),R1           ;  FOR ONE SECTOR
0057?6' 160100                         SUB     R1,R0
0058?0' 020027 001000                  CMP     R0,#NRBYTE
0058?4' 1C  35                         BHIS    BAD1                    ;YES
0058?6' 0  67 003344                   JSR     R5,ERRCS1               ;NO, ALARM
0059?2' 003635                         .WORD   RIOTS-ERMBAS
0059?4' 000177 172170                  JMP     @CUPGER
0059?0' 032777 000001  172176  BAD1:   BIT     #MMTER,@CSYSFW          ;USING MEMMGMNT
005?06' 001422                         BEQ     BAD11                   ;NO
005?70' 032777 000040  172166          BIT     #UNIMAP,@CSYSFW         ;USING UNIBUS MAP
005?76' 001407                         BEQ     BAD12                   ;NO
006?00' 016367 000244  174066          MOV     PRDIOV(R3),CURADR       ;YES
006?06' 016367 000246  174062          MOV     PRDIOV+2(R3),CURADR+2
006?14' 000414                         BR      BAD13
006?16' 016367 000240  174050  BAD12:  MOV     PRDIOX(R3),CURADR
006?24' 016367 000242  174044          MOV     PRDIOX+2(R3),CURADR+2
006?32' 000405                         BR      BAD13
006?34' 005067 174034          BAD11:  CLR     CURADR
006?40' 016367 000016  174030          MOV     PRDIOA(R3),CURADR+2
006?46' 0  767 000400  174024  BAD13:  MOV     #NRWORD,CURCNT          ;SETUP WORD COUNT
006?54' 0  5467 174020                 NEG     CURCNT
006?60' 0 2764 000100  000012          BIT     #VV,RPDS(R4)            ;IS VOLUME VALID
006?66' 001004                         BNE     BAD14                   ;YES
006?70' 012714 000003                  MOV     #PACODE,(R4)
006?74' 004767 005162                  JSR     PC,DRVDLY
007?00' 012767 000632  171776  BAD14:  MOV     #NRCYL-1,CYL            ;INIT ADDRESS
007?06' 012767 000002  171772          MOV     #NRHEAD-1,HEAD
007?14' 032767 010000  173772          BIT     #CFMT,RPCS1V
007?22' 001003                         BNE     BAD141
007?24' 005067 171760                  CLR     SECT
007?30' 000403                         BR      BAD142
007?32' 012767 000001  171750  BAD141: MOV     #1,SECT
007?40' 042767 000001  171734  BAD142: BIC     #IOERR,DFLGWD
007?46' 012701 000635                  MOV     #635,R1                 ;GET COMMAND FLAG
007?52' 010167 173712                  MOV     R1,CURFLG               ;  AND SAVE
007?56' 004767 177410                  JSR     PC,SUIORG               ;SET UP REGS
010?02' 005064 000016                  CLR     RPAS(R4)
010?06' 016767 171720  173712          MOV     RTRY,CURRTY
010074' 005067 173710                  CLR     RTRYIP
010100' 012767 003111  002774          MOV     #IOTERM-ERMBAS,INTEAD
010106' 052713 000010                  BIS     #WT4IOT,(R3)            ;SET WAITING FLAG
010112' 052767 000102  171662          BIS     #CMDISU!DOTERM,DFLGWD
```

```
1514  006120' 010214                          MOV     R2,(R4)                 ;ISSUE READ DATA
1515  006122' 004767  005034                  JSR     PC,DRVDLY
1516  006126' 032767  000001  173666          BIT     @DRIB,DIAFLG            ;IS DRIVE BUSY
1517  006134' 001411                          BEQ     BAD2                    ;NO
1518  006136' 042767  000001  173656          BIC     @DRIB,DIAFLG            ;YES
1519  006144' 042767  000302  171630          BIC     @CMDISU!ANYIOI!DOTERM,DFLGHD
1520  006152' 004577  171670                  JSR     R5,@CIOBSY              ;ALLOW OTHERS TO RUN
1521  006156' 000730                          BR      BAD142                  ;TRY AGAIN
1522  006160' 004577  171662          BAD2:   JSR     R5,@CIOBSY              ;WAIT FOR XFER COMPLETE
1523  006164' 042767  000002  171610          BIC     @DOTERM,DFLGHD
1524  006172' 032767  000001  171602          BIT     @IOERR,DFLGHD           ;DID ERRORS OCCUR
1525  006200' 001414                          BEQ     BAD3                    ;NO
1526  006202' 062767  000002  171600          ADD     @2,SECT                 ;YES, GET NEXT SECTOR
1527  006210' 026727  171574  000026          CMP     SECT,@22.
1528  006216' 002710                          BLT     BAD142
1529  006220' 004567  002764                  JSR     R5,ERRCS                ;CAN NOT READ LAST TRACK
1530  006224' 003310                          .WORD   CRLT-ERMBAS
1531  006226' 000177  171616                  JMP     @CUPGER                 ;GO TO MPG'S ERROR RTN
1532  006232' 004767  000450          BAD3:   JSR     PC,CLRCOD               ;CLEAR OUTPUT BUFFER FOR MSG
1533  006236' 004567  004636                  JSR     R5,PRINTX               ;PRINT HEADING
1534  006242' 001224                          .WORD   BSH-PRIX1
1535  006244' 000030                          .WORD   BSHX-BSH
1536  006246' 016304  000016                  MOV     PRDIOA(R3),R4           ;GET RDIO ADDR
1537  006252' 012400                          MOV     (R4)+,R0                ;GET CARTRIDGE #
1538  006254' 004577  171576                  JSR     R5,@BINASC              ;  AND CONVERT TO ASCII
1539  006260' 007140                          .WORD   PNUM2-.
1540  006262' 012400                          MOV     (R4)+,R0
1541  006264' 004577  171566                  JSR     R5,@BINASC
1542  006270' 007122                          .WORD   PNUM1-.
1543  006272' 004567  004602                  JSR     R5,PRINTX               ;PRINT CARTRIDGE #
1544  006276' 001254                          .WORD   PNUM-PRIX1
1545  006300' 001224                          .WORD   PNUMX-PNUM
1546  006302' 005724                          TST     (R4)+                   ;IGNORE UNUSED WORD OF DATA
1547  006304' 012400                          MOV     (R4)+,R0                ;IS PACK AN ALIGNMENT ONE
1548  006306' 001407                          BEQ     BAD321                  ;NO
1549  006310' 004767  004010                  JSR     PC,SUPTAD               ;SETUP R3 & R4
1550  006314' 004567  002676                  JSR     R5,ERRCS1               ;YES, ALARM
1551  006320' 004053                          .WORD   TIAAP-ERMBAS
1552  006322' 000177  171522                  JMP     @CUPGER                 ;GO TO MPG ERROR RTN
1553  006326' 004767  000354          BAD321: JSR     PC,CLRCOD               ;CLR OUTPUT BUFFER FOR MSG
1554  006332' 010703                          MOV     PC,R3                   ;GET ADDR OF OUTPUT BUFFER
1555  006334' 062703  005706                  ADD     #COUFLD-.,R3
1556  006340' 005067  000032                  CLR     BAD32X                  ;FLAG NO BAD-SECTORS
1557  006344' 005067  173454                  CLR     LUFCNT
1558  006350' 012400          BAD32:  MOV     (R4)+,R0                ;GET NEXT WORD OF DATA
1559  006352' 100512                          BMI     BAD33                   ;DONE
1560  006354' 010367  000016                  MOV     R3,BAD32X               ;PACK CYL
1561  006360' 010701                          MOV     PC,R1                   ;GET ADDR OF ASCII STORAGE
1562  006362' 062701  000014                  ADD     #BAD32X-.,R1
1563  006366' 160167  000004                  SUB     R1,BAD32X               ;  AND MAKE RELATIVE
1564  006372' 004577  171460                  JSR     R5,@BINASC              ;CONVERT CYL TO ASCII
1565  006376' 000000          BAD32X: .WORD   XXXX                    ;STORE RELATIVE ADDR HERE
1566  006400' 062703  000006                  ADD     #6,R3                   ;GET ADDR OF NEXT VALUE
1567  006404' 112723  000054                  MOVB    #',,(R3)+               ;PACK COMMA
1568  006410' 116400  000001                  MOVB    1(R4),R0                ;PACK TRACK
1569  006414' 010367  000016                  MOV     R3,BAD32Y               ;GET REL ADDR FOR DATA
```

```
1570  006420' 010701                  MOV     PC,R1
1571  006422' 062701  000014          ADD     #BAD32Y-.,R1
1572  006426' 160167  000004          SUB     R1,BAD32Y
1573  006432' 004577  171420          JSR     R5,@BINASC              ;CONVERT TRACK TO ASCII
1574  006436' 000000          BAD32Y: .WORD   XXXX                    ;STORE REL ADDR HERE
1575  006440' 062703  000006          ADD     #6,R3
1576  006444' 112723  000054          MOVB    #',,(R3)+               ;PACK COMMA
1577  006450' 012400                  MOV     (R4)+,R0                ;PACK SECTOR
1578  006452' 042700  177400          BIC     #177400,R0              ;REMOVE TRACK
1579  006456' 010367  000016          MOV     R3,BAD32Z               ;GET REL ADDR FOR DATA
1580  006462' 010701                  MOV     PC,R1
1581  006464' 062701  000014          ADD     #BAD32Z-.,R1
1582  006470' 160167  000004          SUB     R1,BAD32Z
1583  006474' 004577  171356          JSR     R5,@BINASC              ;CONVERT SECTOR TO ASCII
1584  006500' 000000          BAD32Z: .WORD   XXXX
1585  006502' 062703  000006          ADD     #6,R3
1586  006506' 112723  000040          MOVB    #' ,(R3)+               ;PACK
1587  006512' 112723  000040          MOVB    #' ,(R3)+               ;  SPACES
1588  006516' 005267  173302          INC     LUPCNT
1589  006522' 026727  173276  000003  CMP     LUPCNT,#3               ;IS LINE RDY FOR PRINTING
1590  006530' 001307                  BNE     BAD32                   ;NO
1591  006532' 005767  177640          TST     BAD32X                  ;WERE ANY BAD SEC FOUND
1592  006536' 001447                  BEQ     BAD4                    ;NO
1593  006540' 032767  000040  173254  BIT     #PBSH,DIAFLG            ;SHOULD PRINT HEADING
1594  006546' 001007                  BNE     BAD322                  ;NO
1595  006550' 052767  000040  173244  BIS     #PBSH,DIAFLG            ;YES
1596  006556' 004567  004316          JSR     R5,PRINTX               ;PRINT HEADING
1597  006562' 002300                  .WORD   BSHEAD-PRIX1
1598  006564' 000024                  .WORD   BSHEX-BSHEAD
1599  006566' 004567  004306  BAD322: JSR     R5,PRINTX               ;PRINT ONE LINE
1600  006572' 001112                  .WORD   CODFLD-PRIX1
1601  006574' 000102                  .WORD   66.
1602  006576' 000653                  BR      BAD321
1603  006600' 005767  177572  BAD33:  TST     BAD32X                  ;WERE BAD SEC FOUND
1604  006604' 001424                  BEQ     BAD4                    ;NO
1605  006606' 032767  000040  173206  BIT     #PBSH,DIAFLG            ;SHOULD PRINT HEADING
1606  006614' 001007                  BNE     BAD331                  ;NO
1607  006616' 052767  000040  173176  BIS     #PBSH,DIAFLG            ;YES
1608  006624' 004567  004250          JSR     R5,PRINTX               ;PRINT HEADING
1609  006630' 002300                  .WORD   BSHEAD-PRIX1
1610  006632' 000024                  .WORD   BSHEX-BSHEAD
1611  006634' 004567  004240  BAD331: JSR     R5,PRINTX               ;YES, PRINT ONE LINE
1612  006640' 001112                  .WORD   CODFLD-PRIX1
1613  006642' 000102                  .WORD   66.
1614  006644' 004567  004230          JSR     R5,PRINTX               ;PRINT END OF BAD SEC
1615  006650' 002367                  .WORD   EOBS-PRIX1
1616  006652' 177753                  .WORD   EOBS-EOBSX
1617  006654' 000404                  BR      BAD41
1618  006656' 004567  004216  BAD4:   JSR     R5,PRINTX               ;PRINT "NONE"
1619  006662' 002324                  .WORD   NONE-PRIX1
1620  006664' 177771                  .WORD   NONE-NONEX
1621  006666' 004767  003432  BAD41:  JSR     PC,SUPTAD               ;SETUP R3 & R4
1622  006672' 004767  002110          JSR     PC,PROCTM               ;PROCESS TERMINATION
1623  006676' 042767  000040  173116  BIC     #PBSH,DIAFLG
1624  006704' 000205                  RTS     R5
1625                          ;
```

```
1626                                    ;SUBROUTINE TO ZERO BUFFER USED TO CONTAIN ASCII OF BAD SECTOR INFORMATION
1627                                    ;
1628  006706' 012700  000041   CLRCOD: MOV     #33.,R0
1629  006712' 010701                   MOV     PC,R1                   ;GET ADDR OF BUFFER
1630  006714' 062701  005326           ADD     #CODFLD-.,R1
1631  006720' 005021           CLRC1:  CLR     (R1)+
1632  006722' 005300                   DEC     R0
1633  006724' 001375                   BNE     CLRC1
1634  006726' 000207                   RTS     PC
```

```
                                        .SBTTL  RK06 INTERRUPT SERVICE ROUTINE
                
006730' 004067 003336          RHPINT: JSR     R0,SAVREG               ;SAVE ALL REGISTERS
006734' 004567 003404                  JSR     R5,STSTAT               ;GO STORE ALL DEV REG'S
006740' 172650                         .WORD   ISTAT-.
006742' 005267 173014                  INC     INTCNT                  ;ADD 1 TO INTERRUPT CNT
006746' 004767 003352                  JSR     PC,SUPTAD               ;SET UP PROG TBL & RPCS1 ADR'S
006752' 004767 004204                  JSR     PC,DRVDLY
006756' 005767 172640                  TST     IRKDS                   ;IS REGISTER VALID?
006762' 100420                         BMI     RHP1                    ;YES
006764' 032701 002000                  BIT     #2000,R1                ;IS THIS 2ND INTERRUPT
006770' 001[illegible]                 BEQ     RHP11                   ;NO, ALARM
006772' 042[illegible] 002000          BIC     #2000,R1                ;YES, SELECT DRIVE
006776' 00[illegible]                  CLR     RS
007000' 004[illegible] 004136          JSR     PC,SD
007004' 000[illegible]                 BR      RHP1                    ;CHECK DI & ATN BITS
007006' 012767 003[illegible] 002066 RHP11:    MOV     #NSVAL-ERMBAS,INTEAD    ;ALARM
007014' 052767 000004 173000          BIS     #NDC,DIAFLG             ;FLAG NEED DRIVE CLR
007022' 000467                         BR      RHP6
007024' 032767 000001 172570 RHP1:     BIT     #DRA,IRKDS              ;IS DRIVE AVAILABLE?
007032' 001006                         BNE     RHP2                    ;YES
007034' 042713 000010                  BIC     #WT4IOT,(R3)            ;CLEAR WAITING FOR I/O
007040' 052767 000001 172754          BIS     #DRIB,DIAFLG            ;FLAG DRIVE IS BUSY
007046' 000554                         BR      RHP62                   ;EXIT UNTIL LATER
007050' 016701 172714          RHP2:   MOV     CURFLG,R1               ;GET THIS CMND'S FLGWD
007054' 005714                         TST     (R4)                    ;IS 'CERR' BIT SET?
007056' 100554                         BMI     RHP3                    ;YES
007060' 016700 172542                  MOV     IRKAS,R0                ;GET ATA BITS
007064' 000300                         SWAB    R0
007066' 032701 000040                  BIT     #40,R1                  ;IS COMMAND SUPPOSED TO SET ATA?
007072' 001422                         BEQ     RHP24                   ;NO, CHECK EXTRA ATA
007074' 032767 040000 172506          BIT     #DI,ICS1                ;IS DI SET?
007102' 001007                         BNE     RHP23                   ;YES, CHECK FOR ATA BIT
007104' 032701 002000                  BIT     #2000,R1                ;IS ANOTHER INTERRUPT TO ARRIVE
007110' 001133                         BNE     RHP62                   ;YES, WAIT FOR 2ND INT
007112' 012767 003157 001762          MOV     #NODI-ERMBAS,INTEAD     ;NO, SETUP ERROR MSG
007120' 000430                         BR      RHP6
007122' 036700 175256          RHP23:  BIT     MYATA,R0                ;IS MY ATA SET?
007126' 001016                         BNE     RHP211                  ;YES, TEST FOR ERRORS
007130' 012767 003137 001744          MOV     #NOATA-ERMBAS,INTEAD    ;NO, ALARM
007136' 000421                         BR      RHP6
007140' 036700 175240          RHP24:  BIT     MYATA,R0                ;IS MY ATA SET?
007144' 001412                         BEQ     RHP21                   ;NO, TEST FOR ERRORS
007146' 012767 003176 001726          MOV     #UXPATA-ERMBAS,INTEAD   ;YES, ALARM
007154' 052767 000004 172640          BIS     #NDC,DIAFLG             ;FLAG NEED DRIVE CLR
007162' 000407                         BR      RHP6
007164' 052767 000004 172630 RHP211:   BIS     #NDC,DIAFLG             ;FLAG NEED DRIVE CLR
007172' 052767 000002 172622 RHP21:    BIS     #NEE,DIAFLG             ;FLAG NO ERRORS EXPECTED
007200' 000506                         BR      RHP313
                                       ;
                                       ;EXIT
                                       ;
007202' 052767 000001 170572 RHP6:     BIS     #IOERR,DFLGWD           ;SET TERM I/O ERROR FLAG
007210' 042713 000010          RHP61:  BIC     #WT4IOT,(R3)            ;RESET WAITING FOR I/O TERM
007214' 042767 000002 172600          BIC     #NEE,DIAFLG
007222' 032767 000004 172572          BIT     #NDC,DIAFLG             ;SHOULD DO A DRIVE CLR
```

```
1692  007230' 001405                          BEQ     RHP612                  ;NO
1693  007232' 004767  004012                  JSR     PC,DRVCLR               ;YES
1694  007236' 042767  000004  172556          BIC     #NDC,DIAFLG
1695  007244' 032767  000010  172550  RHP612: BIT     #NRC,DIAFLG             ;DO I NEED A RECAL
1696  007252' 001426                          BEQ     RHP613                  ;NO
1697  007254' 016446  000010                  MOV     RPCS2(R4),-(SP)         ;YES
1698  007260' 052764  000040  000010          BIS     #SCLR,RPCS2(R4)
1699  007266' 004767  003670                  JSR     PC,DRVDLY
1700  007272' 012664  000010                  MOV     (SP)+,RPCS2(R4)
1701  007276' 012714  000013                  MOV     #RCCODE,(R4)            ;  OUTPUT RECAL
1702  007302' 052767  000400  170472          BIS     #SPINFL,DFLGWD          ;PREPARE FOR INT TIMEOUT
1703  007310' 042767  000010  172504          BIC     #NRC,DIAFLG             ;  TO TEST DRDY BIT
1704  007316' 012763  005670  000030          MOV     #3000.,PTOCNT(R3)       ;  WHEN HEADS ARE HOME
1705  007324' 052713  000010                  BIS     #WT4IOT,(R3)            ;FLAG STAY IN MPG UNTIL DONE
1706  007330' 032767  000020  172464  RHP613: BIT     #NCC,DIAFLG             ;DO NEED CONTROL CLR
1707  007336' 001407                          BEQ     RHP611                  ;NO
1708  007340' 042767  000020  172454          BIC     #NCC,DIAFLG             ;YES
1709  007346' 004767  003634                  JSR     PC,CTRLCL               ;CCLR
1710  007352' 004767  003672                  JSR     PC,DRVCLR               ;DRIVE CLR
1711  007356' 032767  000004  172404  RHP611: BIT     #4,CURFLG               ;THIS A DATA TRANSFER CMND?
1712  007364' 001403                          BEQ     RHP614                  ;NO
1713  007366' 016467  000002  172410          MOV     RPWC(R4),FINCNT         ;SAVE FINAL WORD COUNT
1714  007374' 042714  000100          RHP614: BIC     #IE,(R4)                ;REMOVE IE
1715  007400' 004067  002702          RHP62:  JSR     R0,RESREG               ;RESTORE ALL REGISTERS
1716  007404' 000177  170464                  JMP     @RTNINT                 ;EXIT FROM INTERRUPT
1717                                          ;
1718                                          ;ANALIZE ERROR CONDITION
1719                                          ;
1720  007410' 052767  000020  172404  RHP3:   BIS     #NCC,DIAFLG             ;FLAG NEED CCLR
1721  007416' 032767  010420  172200  RHP313: BIT     #DTE!HVRC!FMTE,IRKER    ;DID OTHER ERRORS OCCUR?
1722  007424' 001141                          BNE     CKRTRY                  ;YES, CHECK RETRIES
1723  007426' 032767  100000  172170          BIT     #DCK,IRKER              ;DID DATA ERROR OCCUR?
1724  007434' 001402                          BEQ     RHP315                  ;NO
1725  007436' 000167  000530                  JMP     CKCORR
1726  007442' 032767  024000  172140  RHP315: BIT     #CTO!SPAR,ICS1          ;DID CTO OR SPAR ERRORS OCCUR?
1727  007450' 001402                          BEQ     RHP31                   ;NO
1728  007452' 000167  000472                  JMP     HARDER                  ;YES
1729  007456' 032767  037400  172134  RHP31:  BIT     #UFE!MDS!PGE!NEM!NED!UPE,ICS2 ;ANY RKCS2 ERRORS?
1730  007464' 001402                          BEQ     RHP311                  ;NO
1731  007466' 000167  000456                  JMP     HARDER                  ;YES
1732  007472' 032767  140000  172120  RHP311: BIT     #DLT!WCE,ICS2           ;DID DLT OR WCE OCCUR?
1733  007500' 001113                          BNE     CKRTRY                  ;YES, CHECK RETRIES
1734  007502' 032767  004070  172112          BIT     #ACLO!DSL!DROT!WRL,IRKDS ;ANY RKDS ERRORS?
1735  007510' 001402                          BEQ     RHP312                  ;NO
1736  007512' 000167  000432                  JMP     HARDER                  ;YES
1737          067357          RKERBT=ILF!SKI!NXF!DRPAR!DTYE!ECH!BSE!COE!IDAE!WLE!OPI!UNS
1738  007516' 032767  067357  172100  RHP312: BIT     #RKERBT,IRKER           ;ANY RKER ERRORS?
1739  007524' 001402                          BEQ     RHP314                  ;NO
1740  007526' 000167  000416                  JMP     HARDER                  ;YES
1741  007532' 032767  000002  172262  RHP314: BIT     #NEE,DIAFLG             ;ARE ERRORS EXPECTED
1742  007540' 001010                          BNE     RHP4                    ;NO
1743  007542' 012767  003340  001332          MOV     #UNKERR-ERMBAS,INTEAD   ;SETUP FOR UNKNOWN ERROR
1744  007550' 052767  000020  172244          BIS     #NCC,DIAFLG             ;FLAG NEED CCLR
1745  007556' 000167  177420                  JMP     RHP6
1746                                          ;
1747                                          ;SERVICE RDHD COMMAND
```

```
1748                                                    ;
1749  007562' 032701  001000          RHP4:   BIT     #1000,R1                ;IS THIS A RDHD COMMAND
1750  007566' 001610                  BEQ     RHP61                   ;NO, FINISHED
1751  007570' 032767  000200  172022          BIT     #DR,ICS2                ;IS DATA OUTPUT RDY
1752  007576' 001005                  BNE     RHP41                   ;YES, GET HEADER DATA
1753  007600' 012767  003521  001274          MOV     #DNRRH-ERMBAS,INTEAD    ;NO, ALARM
1754  007606' 000167  177370          JMP     RHP6
1755  007612' 016700  172156  RHP41: MOV     CURADR,R0               ;GET STORAGE ADDRESS
1756  007616' 016701  172012          MOV     IRKDB,R1                ;GET 1ST HEADER WORD
1757  007622' 004777  170252          JSR     PC,@PUTBYT              ;LET MPG STORE IT
1758  007626' 000301                  SWAB    R1
1759  007630' 004777  170244          JSR     PC,@PUTBYT
1760  007634' 016401  000024          MOV     RPDB(R4),R1             ;GET 2ND HEADER WORD
1761  007640' 004777  170234          JSR     PC,@PUTBYT              ;LET MPG STORE IT
1762  007644' 000301                  SWAB    R1
1763  007646' 004777  170226          JSR     PC,@PUTBYT
1764  007652' 016401  000024          MOV     RPDB(R4),R1             ;GET 3RD HEADER WORD
1765  007656' 004777  170216          JSR     PC,@PUTBYT              ;LET MPG STORE IT
1766  007662' 000301                  SWAB    R1
1767  007664' 004777  170210          JSR     PC,@PUTBYT
1768  007670' 062767  000006  172014          ADD     #6,BYRD+2               ;COUNT NR BYTES REAJ
1769  007676' 005567  172006          ADC     BYRD
1770  007702' 062767  000006  170110          ADD     #6,SIZE
1771  007710' 005367  172066          DEC     CURPBC                  ;SHOULD I GET MORE HEADERS
1772  007714' 003002                  BGT     RHP42                   ;YES
1773  007716' 000167  177266          JMP     RHP61                   ;NO
1774  007722' 010067  172046  RHP42: MOV     R0,CURADR               ;SAVE STORAGE ADDR
1775  007726' 000476                  BR      CKR1                    ;YES
1776                                                    ;
1777                                                    ;DETERMINE WHICH ERROR SHOULD BE RETRIED
1778                                                    ;
1779  007730' 005767  172054  CKRTRY: TST     RTRYIP                  ;ALREADY DONE RETRIES ON THIS CMND?
1780  007734' 001054                  BNE     55$                     ;N,Y-55$
1781  007736' 005767  172044          TST     CURRTY                  ;ARE RETRIES SPECIFIED?
1782  007742' 001502                  BEQ     HARDER                  ;Y,N-HARDER
1783  007744' 032767  100000  171646          BIT     #DLT,ICS2               ;DLT ERROR?
1784
1785  007752' 001403                  BEQ     42$                     ;Y,N-42$
1786  007754' 005267  171764          INC     DLTCNT                  ;ADD 1 TO DLT COUNT
1787  007760' 000442                  BR      55$                     ;GO CK RETRY COUNT
1788  007762' 032767  010000  171634  42$:    BIT     #DTE,IRKER              ;DTE ERROR?
1789  007770' 001403                  BEQ     44$                     ;Y,N-44$
1790  007772' 005267  171750          INC     DTECNT                  ;ADD 1 TO DTE COUNT
1791  007776' 000433                  BR      55$                     ;GO CK RETRY COUNT
1792  010000' 032767  000400  171616  44$:    BIT     #HVRC,IRKER             ;HVRC ERROR?
1793  010006' 001403                  BEQ     46$                     ;Y,N-46$
1794  010010' 005267  171734          INC     HCRCNT                  ;ADD 1 TO HCRC COUNT
1795  010014' 000424                  BR      55$                     ;GO CK RETRY COUNT
1796  010016' 032767  000020  171600  46$:    BIT     #FMTE,IRKER             ;FER ERROR?
1797  010024' 001403                  BEQ     50$                     ;NO
1798  010026' 005267  171720          INC     FERCNT                  ;ADD 1 TO FER COUNT
1799  010032' 000415                  BR      55$                     ;GO CK RETRY COUNT
1800  010034' 032767  100000  171562  50$:    BIT     #DCK,IRKER              ;DCK ERROR?
1801  010042' 001403                  BEQ     52$                     ;Y,N-52$
1802  010044' 005267  171704          INC     DCKCNT                  ;ADD 1 TO DCK COUNT
1803  010050' 000406                  BR      55$                     ;GO CK RETRY COUNT
```

```
 1804  010052' 032767  040000  171540  52$:    BIT     #WCE,ICS2               ;WCE ERROR?
 1805  010060' 001402                          BEQ     55$                     ;Y,N-55$
 1806  010062' 005267  171670                  INC     WCECNT                  ;ADD 1 TO WCE COUNT
 1807  010066' 005367  171714          55$:    DEC     CURRTY                  ;DECREMENT RETRY COUNT
 1808  010072' 100004                          BPL     60$                     ;CNT EXHAUSTED? (Y,N-60$)
 1809  010074' 012767  003215  001000          MOV     #RTYEXH-ERMBAS,INTEAD   ;SET UP EXHAUSTED RETRIES ERR MSG ADR
 1810  010102' 000431                          BR      JSETER                  ;GO TO ERROR EXIT
 1811  010104' 005267  171650          60$:    INC     RETRYS                  ;ADD 1 TO RETRY TOTAL CNT
 1812  010110' 005267  171674                  INC     RTRYIP                  ;SET RETRY IN PROGRESS FLAG
 1813  010114' 004767  003066                  JSR     PC,CTRLCL               ;CLEAR CONTROLLER
 1814  010120' 004767  003124                  JSR     PC,DRVCLR               ;CLEAR DRIVE
 1815  010124' 016702  171642          CKR1:   MOV     CURCMD,R2               ;GET CURR CMND IN R2
 1816  010130' 004767  175336                  JSR     PC,SUIORG               ;SET UP DEV REGS
 1817  010134' 012763  072460  000030          MOV     #TIME,PTOCNT(R3)        ;HSKP T/O COUNT
 1818  010142' 010214                          MOV     R2,(R4)                 ;RE-ISSUE THE ORIG CMND
 1819  010144' 000167  177230                  JMP     RHP62                   ;GO TO INT EXIT
 1820
 1821  010150' 032767  002002  171446  HARDER: BIT     #IDAE!SKI,IRKER         ;DID ADDRESSING ERROR OCCUR
 1822  010156' 001403                          BEQ     JSETER                  ;NO
 1823  010160' 052767  000010  171634          BIS     #MRC,DIAFLG             ;YES, FLAG NEED RECAL
 1824  010166' 000167  177010          JSETER: JMP     RHP6                    ;GO TO ERROR EXIT
 1825
 1826  010172' 032767  002000  167602  CKCORR: BIT     #CORFLG,DFLGWD          ;CORCON STATEMENT IN EFFEC.
 1827  010200' 001253                          BNE     CKRTRY                  ;Y,N-CKRTRY
 1828  010202' 032714  010000                  BIT     #CFMT,(R4)              ;IN 18 BIT WORD MODE?
 1829  010206' 001250                          BNE     CKRTRY                  ;N,Y-CKRTRY
 1830  010210' 032764  000100  000014          BIT     #ECH,RPER(R4)           ;IS THIS A HARD ERROR
 1831  010216' 001244                          BNE     CKRTRY                  ;YES, RETRY BUT NO CORRECTION
 1832  010220' 005764  000032                  TST     RPEC2(R4)               ;ECC BIT PATTERN = 0?
 1833  010224' 001004                          BNE     70$                     ;Y,N-70$
 1834  010226' 012767  003237  000646          MOV     #INVPAT-ERMBAS,INTEAD   ;SET UP INV BIT PATTERN ERR MSG ADR
 1835  010234' 000754                          BR      JSETER                  ;GO TO ERROR EXIT
 1836  010236' 026427  000030  010041  70$:    CMP     RPEC1(R4),#10041        ;ECC BIT POSITION TOO BIG?
 1837  010244' 101404                          BLOS    80$                     ;Y,N-80$
 1838  010246' 012767  003263  000626          MOV     #INVPOS-ERMBAS,INTEAD   ;SET UP INV BIT POSITION ERR MSG ADR
 1839  010254' 000744                          BR      JSETER                  ;GO TO ERROR EXIT
 1840  010256' 016405  000002          80$:    MOV     RPWC(R4),R5             ;GET REMAINING NEG WORD CNT
 1841  010262' 006305                          ASL     R5                      ;MAKE IT A BYTE CNT
 1842  010264' 066705  171512                  ADD     CURPBC,R5               ;ADD IN ORIG POS BYTE CNT
 1843  010270' 001617                          BEQ     CKRTRY                  ;ANY BYTES READ? (Y,N-CKRTRY)
 1844  010272' 005001                          CLR     R1                      ;RESET DATA START DISPL VALUE
 1845  010274' 012746  001000                  MOV     #512.,-(SP)             ;INITIALIZE BLOCK LENGTH
 1846  010300' 020516                  85$:    CMP     R5,(SP)                 ;REACHED START OF BAD BLOCK?
 1847  010302' 101403                          BLOS    90$                     ;N,Y-90$
 1848  010304' 061601                          ADD     (SP),R1                 ;ADD BLK LNGTH TO DATA DISPL
 1849  010306' 161605                          SUB     (SP),R5                 ;REDUCE BYTE READ CNT
 1850  010310' 000773                          BR      85$                     ;GO CK IT NOW
 1851  010312' 005726                  90$:    TST     (SP)+                   ;CORRECT STACK
 1852  010314' 016400  000030                  MOV     RPEC1(R4),R0            ;GET ECC BIT POSITION VALUE
 1853  010320' 005300                          DEC     R0                      ;ADJ IT FOR SHIFTING
 1854  010322' 010002                          MOV     R0,R2                   ;SET UP A SHIFT COUNT
 1855  010324' 042702  177760                  BIC     #177760,R2              ;ISOLATE SHIFT CNT BITS
 1856  010330' 040200                          BIC     R2,R0                   ;CLEAR SHIFT CNT BITS IN BIT POSITION
 1857  010332' 006200                          ASR     R0                      ;CONVERT BIT POSITION TO A WORD
 1858  010334' 006200                          ASR     R0                      ;DISPL INTO THE DATA IN THIS
 1859  010336' 006200                          ASR     R0                      ;BLOCK
```

```
 1860  010340' 060001                        ADD     R0,R1                   ;ADD IT TO DATA DISPL VALUE
 1861  010342' 020167 171434                 CMP     R1,CURPBC               ;CORRECTION WITHIN USER'S DATA?
 1862  010346' 103[ ]                        BHIS    110$                    ;Y,N-110$
 1863  010350' 005267 171364                 INC     CECCER                  ;ADD 1 TO CORRECTABLE ECC ERROR CNT
 1864  010354' 016405 000032                 MOV     RPEC2(R4),R5            ;GET ECC BIT PATTERN
 1865  010360' 005046                        CLR     -(SP)                   ;CLEAR SHIFT INTO WORD
 1866  010362' 005302                 95$:   DEC     R2                      ;DECREMENT SHIFT COUNT
 1867  010364' 0[ ]403                       BLT     100$                    ;FINISHED SHIFTING? (N,Y-100$)
 1868  010366' 006305                        ASL     R5                      ;DO A BIT SHIFT ON 2 WORDS
 1869  010370' 006116                        ROL     (SP)
 1870  010372' 000773                        BR      95$                     ;GO CK SHIFT CNT
 1871  010374' 004767 000052          100$:  JSR     PC,DOCORR               ;CORRECT 1ST WORD OF PAIR
 1872  010400' 012605                        MOV     (SP)+,R5                ;GET 2ND CORRECTION WORD
 1873  010402' 062701 000002                 ADD     #2,R1                   ;INCR DATA DISPL VALUE
 1874  010406' 020167 171370                 CMP     R1,CURPBC               ;STILL WITHIN HIS DATA?
 1875  010412' 103002                        BHIS    105$                    ;Y,N-105$
 1876  010414' 004767 000032                 JSR     PC,DOCORR               ;CORRECT 2ND WORD OF PAIR
 1877  010420' 004767 001700          105$:  JSR     PC,SUPTAD               ;SET UP R3 & R4 AGAIN
 1878  010424' 005764 000002          110$:  TST     RPWC(R4)                ;ANY MORE DATA TO READ?
 1879  010430' 001002                        BNE     115$                    ;Y,N-115$
 1880  010432' 000167 176552                 JMP     RHP61                   ;GO CLEAR WAIT FLG & EXIT
 1881  010436' 004767 002544          115$:  JSR     PC,CTRLCL               ;ISSUE CTRL CLR CMND
 1882  010442' 016714 171324                 MOV     CURCMD,(R4)             ;RESUME ORIG CMND
 1883  010446' 000167 176726                 JMP     RHP62                   ;GO TO INT EXIT
 1884
 1885
 1886
 1887                                         ;DO ECC DATA CORRECTION
 1888
 1889                                                 ;JSR    PC,DOCORR       S/R CALL
 1890                                                 ;
 1891                                                 ;R1 = DATA WORD DISPLACEMENT
 1892                                                 ;R5 = BIT PATTERN
 1893
 1894  010452' 016703 171316          DOCORR: MOV     CURADR,R3               ;GET DATA'S ABS STARTING ADR
 1895  010456' 016704 171314                 MOV     CURADR+2,R4
 1896  010462' 060104                        ADD     R1,R4                   ;ADD DATA DISPL INTO IT
 1897  010464' 005503                        ADC     R3
 1898  010466' 032777 000001 167370         BIT     #MMVER,@CSYSFW          ;RUNNING UNDER MEM MGMNT?
 1899  010474' 001460                        BEQ     20$                     ;Y,N-20$
 1900  010476' 032777 000040 167360         BIT     #UNIMAP,@CSYSFW         ;IS UNIBUS MAP USED?
 1901  010504' 001430                        BEQ     5$                      ;NO
 1902  010506' 010446                        MOV     R4,-(SP)                ;YES, GET UPPER 5 BITS
 1903  010510' 006203                        ASR     R3                      ;  OF VIRTUAL ADDRESS
 1904  010512' 006016                        ROR     (SP)                    ;MERGE BIT 16 WITH BITS15-13
 1905  010514' 006203                        ASR     R3                      ;PUT BIT 17 IN CARRY
 1906  010516' 006016                        ROR     (SP)                    ;MERGE BIT 17 IN WITH BITS16-13
 1907  010520' 006216                        ASR     (SP)                    ;POSITION SUCH THAT IS X4
 1908  010522' 042716 101777                 BIC     #101777,(SP)            ;REMOVE ANY SIGN BIT
 1909  010526' 000316                        SWAB    (SP)                    ;MOVE 5 BITS TO LOWER BYTE
 1910  010530' 062716 170200                 ADD     #BUSMAP,(SP)            ;GET BASE ADR OF REG
 1911  010534' 010446                        MOV     R4,-(SP)                ;SAVE ORIGINAL OFFSET
 1912  010536' 017604 000002                 MOV     @2(SP),R4               ;GET CONTENTS OF MAP REGS
 1913  010542' 062766 000002 000002         ADD     #2,2(SP)
 1914  010550' 017603 000002                 MOV     @2(SP),R3
 1915  010554' 042716 160000                 BIC     #160000,(SP)            ;MERGE IN ORIGINAL OFFSET
```

```
1916  010560' 062604                      ADD     (SP)+,R4
1917  010562' 005503                      ADC     R3
1918  010564' 005726                      TST     (SP)+                   ;CORRECT STACK
1919  010566' 010400              5$:     MOV     R4,R0                   ;SAVE LOW 16 BITS IN WORK AREA
1920  010570' 042704 177700              BIC     #177700,R4              ;ISOLATE UP TO 32 WORD OFFSET
1921  010574' 052704 100000              BIS     #P4CONS,R4              ;SET ADR TO SELECT PAGE 4
1922  010600' 012702 000006              MOV     #6,R2                   ;SET UP SHIFT CNT
1923  010604' 006203              10$:    ASR     R3                      ;SHIFT ADR 1 BIT
1924  010606' 006000                      ROR     R0
1925  010610' 005302                      DEC     R2                      ;GOT HI 16 BITS IN 1 WORD?
1926  010612' 001374                      BNE     10$                     ;Y,N-10$
1927  010614' 013746 172350               MOV     @#KPAR4,-(SP)           ;SAVE PAGE 4 STUFF
1928  010620' 013746 172310               MOV     @#KPDR4,-(SP)
1929  010624' 010037 172350               MOV     R0,@#KPAR4              ;SET PAGE 4 TO USER'S ARREA
1930  010630' 012737 077406 172310        MOV     #PDRCON,@#KPDR4
1931  010636' 011400              20$:    MOV     (R4),R0                 ;GET DATA WORD READ
1932  010640' 040514                      BIC     R5,(R4)                 ;RESET PATTERN'S BITS IN WORD READ
1933  010642' 040005                      BIC     R0,R5                   ;RESET WORD READ BITS IN PAT WORD
1934  010644' 050514                      BIS     R5,(R4)                 ;SET REMAINING SINGLE BITS BACK IN
1935  010646' 032777 000001 167210        BIT     #MMVER,@CSYSFW          ;RUNNING UNDER MEM MGMNT?
1936  010654' 001404                      BEQ     30$                     ;Y,N-30$
1937  010656' 012637 172310               MOV     (SP)+,@#KPDR4           ;RESTORE PAGE 4'S REGS
1938  010662' 012637 172350               MOV     (SP)+,@#KPAR4
1939  010666' 000207              30$:    RTS     PC                      ;EXIT IN-LINE
```

```
                                        .SBTTL  SUBROUTINES FOR RK06 FUNCTION ROUTINES


                                        ;CHECK IF DEVICE IS BUSY AND WAIT IF IT IS

                                                ;JSR    PC,CKDBSY       S/R CALL
                                                ;
                                                ;DESTROYS R0,R3,R4
                                                ;ON EXIT:  R3 = PROG TBL ADR
                                                ;          R4 = RPCS1 ADR

010670' 004767  001430          CKDBSY: JSR     PC,SUPTAD               ;SET UP PROG TBL & RPCS1 ADR'S
010674' 032714  000100          10$:    BIT     #IE,(R4)                ;INT ENABLE ON'
010700' 001403                  BEQ     20$                             ;Y,N-20$
010702' 004577  167140          15$:    JSR     R5,@CIOBSY              ;RELEASE CONTROL
010706' 000772                  BR      10$                             ;GO CK AGAIN
010710' 032767  000400  167064  20$:    BIT     #SPINFL,DFLGWD          ;DO I HAVE AN SPIN IN PROGRESS'
010716' 001403                  BEQ     25$                             ;Y,N-25$
010720' 052713  000010                  BIS     #WT4IOT,(R3)            ;SET WAITING FOR I/O TERM
010724' 000766                  BR      15$                             ;GO RELEASE CONTROL
010726' 032767  000002  167046  25$:    BIT     #DOTERM,DFLGWD          ;HAVE TO PROCESS PREV TERMINATION'
010734' 001403                  BEQ     30$                             ;Y,N-30$
010736' 004767  000044                  JSR     PC,PROCTM               ;GO PROCESS TERMINATION
010742' 000754                  BR      10$                             ;GO CK INT ENABLE AGAIN
010744' 016767  167056  000012  30$:    MOV     IVCTAD,4@$              ;STORE INT VECTOR ADR
010752' 016767  167052  000006          MOV     PSWD,4$                 ;STORE PROC STATUS WORD
010760' 004577  167102                  JSR     R5,@SETVEC              ;GO SET UP INTERRUPT VECTOR
010764' 000000                  4@$:    .WORD   XXXX                    ;  INT VECTOR ADR
010766' 000000                  4$:     .WORD   XXXX                    ;  PSW
010770' 175740                          .WORD   RHPINT-.                ;  REL INT ROUT ADR
010772' 010567  170766                  MOV     R5,ERRADR               ;SAVE CURR USER STMNT ADR
010776' 162767  000004  170760          SUB     #4,ERRADR
011004' 000207                          RTS     PC                      ;EXIT IN-LINE


                                        ;PROCESS TERMINATION OF PREVIOUS I/O FUNCTION

                                                ;JSR    PC,PROCTM       S/R CALL
                                                ;
                                                ;R3 = PROG TABLE ADR
                                                ;
                                                ;DESTROYS R0

011006' 010146          PROCTM: MOV     R1,-(SP)                        ;SAVE R1 & R2
011010' 010246                  MOV     R2,-(SP)
011012' 042767  000002  166762          BIC     #DOTERM,DFLGWD          ;RESET PROCESS TERMINATION FLAG
011020' 032767  000010  170742          BIT     #10,CURFLG              ;INCR BYTE COUNT'
011026' 001417                  BEQ     20$                             ;Y,N-20$
011030' 016700  170744                  MOV     CURCNT,R0               ;GET INITIAL WORD CNT
011034' 005400                  NEG     R0                              ;MAKE IT POSITIVE AGAIN
011036' 016701  170742                  MOV     FINCNT,R1               ;GET FINAL WORD CNT
011042' 100001                  BPL     10$                             ;IS IT NEGATIVE' (Y,N-10$)
011044' 005401                  NEG     R1                              ;MAKE IT POSITIVE
011046' 160100          10$:    SUB     R1,R0                           ;SUB REMAINING CNT FROM INITIAL CNT
011050' 006300                  ASL     R0                              ;MAKE IT A BYTE CNT
011052' 010067  166742                  MOV     R0,SIZE                 ;STORE # OF BYTES ACTUALLY XFERRED
```

```
011056' 016701  170704                  MOV     CNTADR,R1               ;GET ADR OF BYTE CNT TOTALS
011062' 060011                          ADD     R0,(R1)                 ;ADD IN THIS CNT
011064' 005541                          ADC     -(R1)                   ;UPDATE MOST SIGNF WORD OF CNT
011066' 032767  000001  166706  20$:    BIT     #IOERR,DFLGWD           ;WAS THERE AN ERROR?
011074' 001412                          BEQ     PROCEX                  ;Y,N-PROCEX
011076' 004567  000132                  JSR     R5,ERRIS                ;GO ISSUE I/O TERMINATION
011102' 003111                  INTEAD: .WORD   IOTERM-ERMBAS           ;ERROR MSG
011104' 004767  000024                  JSR     PC,RINTV                ;RESET THE INT VECTOR
011110' 012602                          MOV     (SP)+,R2                ;RESTORE R1 & R2
011112' 012601                          MOV     (SP)+,R1
011114' 004577  166730                  JSR     R5,@CUPGER              ;GO TO MPG ERR RETN POINT
011120' 000207                          RTS     PC                      ;RETURN IN-LINE
011122' 004767  000006          PROCEX: JSR     PC,RINTV                ;GO RESET INT VECTOR
011126' 012602                          MOV     (SP)+,R2                ;RESTORE R1 & R2
011130' 012601                          MOV     (SP)+,R1
011132' 000207                          RTS     PC                      ;EXIT IN-LINE


                                        ;RESET INTERRUPT VECTOR S/R

                                                ;JSR    PC,RINTV        S/R CALL
                                                ;R3 MUST CONTAIN PROG TBL ADR
                                                ;DESTROYS R0

011134' 004567  000020          RINTV:  JSR     R5,TVECT                ;GO CK IF I HAVE VECTOR CONTROL
011140' 000406                          BR      RINTEX                  ;BR IF I DON'T
011142' 016767  166660  000004          MOV     IVCTAD,10$              ;GET CURR INT VECT ADR
011150' 004577  166714                  JSR     R5,@CLRVEC              ;GO HAVE MPG CLEAR IT
011154' 000000                  10$:    .WORD   XXXX
011156' 000207                  RINTEX: RTS     PC                      ;EXIT IN-LINE


                                        ;TEST INTERRUPT VECTOR S/R

                                                ;JSR    R5,TVECT        S/R CALL
                                                ;BR     LABEL           EXECUTED IF NOT SAME
                                                ;R3 MUST CONTAIN PROG TBL ADR
                                                ;DESTROYS R0

011160' 016767  166642  000010  TVECT:  MOV     IVCTAD,20$              ;GET CURR INT VECT ADR
011166' 016346  000004                  MOV     PFWADR(R3),-(SP)        ;STORE FLGWD ADR TO IDENTIFY ME
011172' 004577  166674                  JSR     R5,@TSTVEC              ;DO I HAVE VECTOR CONTROL?
011176' 000000                  20$:    .WORD   XXXX                    ;  MPG WILL TELL ME SINCE I CAN'T
011200' 175530                          .WORD   RHPINT-.                ;  GET AT LOWER MEM IF MEM MGMNT
011202' 000401                          BR      TVECTX                  ;BR IF I DON'T HAVE CNTRL
011204' 005725                          TST     (R5)+                   ;BYPASS BR INST IN S/R CALL
011206' 000205                  TVECTX: RTS     R5                      ;EXIT IN-LINE
```

```
                                        ;ERROR INFORMATION DISPLAY S/R
                                        ;
                                        ;JSR     R5,ERRCS        S/R CALL FOR CURR STATUS
                                        ;JSR     R5,ERRCS1       S/R CALL FOR CURR STATUS W/O STORING
                                        ;JSR     R5,ERRIS        S/R CALL FOR INT STATUS
                                        ;.WORD   MSGADR-ERMBAS   REL ADR OF ERROR MSG
                                        ;
                                        ;R3 = PROG TABLE ADR
                                        ;DESTROYS R0,R1,R2

011210' 004567  001130          ERRCS:  JSR     R5,STSTAT               ;STORE CURR STATUS
011214' 170434                          .WORD   CSTAT-.
011216' 012767  170020  000404  ERRCS1: MOV     #CSTAT-ERSTAD,ERSTAD    ;STORE ADR OF CURR STATUS
011224' 012767  170206  000212          MOV     #CSTAT-EBSTAD,EBSTAT
011232' 000406                          BR      ERRCOM                  ;GO TO COMMON POINT
011234' 012767  167760  000366  ERRIS:  MOV     #ISTAT-ERSTAD,ERSTAD    ;STORE ADR OF LAST INT STATUS
011242' 012767  170146  000174          MOV     #ISTAT-EBSTAD,EBSTAT
011250' 012567  000134          ERRCOM: MOV     (R5)+,ERMBAS            ;STORE MSG ADR
011254' 005267  170456                  INC     ERRCNT                  ;ADD 1 TO ERROR CNT
011260' 012767  000001  166534          MOV     #1,ERRI                 ;SET THE ERROR INDICATOR
011266' 032763  000400  000002          BIT     #DOERCK,POPSW(R3)       ;SUPPOSED TO DO ERROR CHECKING?
011274' 001004                          BNE     2$                      ;Y,N-2$
011276' 032763  020000  000002          BIT     #PRONER,POPSW(R3)       ;ERROR PRINTING INHIBITED?
011304' 001402                          BEQ     4$                      ;N,Y-4$
011306' 000167  000404          2$:     JMP     ERREX                   ;GO TO EXIT
011312' 010446                  4$:     MOV     R4,-(SP)                ;SAVE R4 & R5
011314' 010546                          MOV     R5,-(SP)
011316' 005004                          CLR     R4                      ;SET USER MODE PRINT FLAG
011320' 004767  001114                  JSR     PC,DEVID                ;DISPLAY DEVICE I.D.
011324' 032767  000100  166450          BIT     #CMDISU,DFLGWD          ;HAS THE CMND BEEN ISSUED?
011332' 001005                          BNE     6$                      ;N,Y-6$
011334' 004567  001424                  JSR     R5,PRINT                ;PRINT THE "BEFORE ISSUING I/O" MSG
011340' 003004                          .WORD   BEFIO-.
011342' 000030                          .WORD   24.
011344' 000404                          BR      8$                      ;GO CALC MSG LNGTH
011346' 004567  001412          6$:     JSR     R5,PRINT                ;PRINT THE "AFTER ISSUING I/O" MSG
011352' 003022                          .WORD   AFTIO-.
011354' 000027                          .WORD   23.
011356' 010700                  8$:     MOV     PC,R0                   ;GET START ADR OF ERROR MSG
011360' 062700  000030                  ADD     #ERMBAS-.,R0
011364' 061000                          ADD     (R0),R0
011366' 012701  177777                  MOV     #-1,R1                  ;INITIALIZE MSG LENGTH
011372' 005201                  10$:    INC     R1                      ;ADD 1 TO MSG LENGTH
011374' 105720                          TSTB    (R0)+                   ;MSG TERMINATOR?
011376' 001375                          BNE     10$                     ;Y,N-10$
011400' 010167  000006                  MOV     R1,ERMBAS+2             ;STORE MSG LENGTH
011404' 004567  001354                  JSR     R5,PRINT                ;PRINT ERROR MSG SPECIFIED
011410' 000000          ERMBAS: .WORD   XXXX
011412' 000000                  .WORD   XXXX
011414' 026727  177770  003562          CMP     ERMBAS,#INVDVN-ERMSGS   ;INVALID UNIT # MSG OR HIGHER?
011422' 103105                          BHIS    ERRSNM                  ;N,Y-ERRSNM
011424' 010701                          MOV     PC,R1                   ;GET ADR OF CODE  AREA IN ERR MSG
011426' 062701  002614                  ADD     #CODFLD-.,R1
011432' 010700                          MOV     PC,R0                   ;SET UP ADR OF ERROR CODE TBL
011434' 062700  000264                  ADD     #ERCDTB-.,R0
011440' 010702                          MOV     PC,R2                   ;SET UP ADR OF STORED DEV REG'S
```

```
2101  011442' 062702                    EBSBAS: ADD     (PC)+,R2
2102  011444' 170206                    EBSTAT: .WORD   CSTAT-EBSBAS
2103  011446' 012767  000015  000146            MOV     #13.,70$        ;INITIALIZE MSG LENGTH
2104  011454' 012746  000100                    MOV     #64.,-(SP)      ;INITIALIZE CODE FIELD CNT
2105  011460' 012205                    15$:    MOV     (R2)+,R5        ;GET NEXT DEV REG WORD
2106  011462' 000305                    17$:    SWAB    R5              ;GET DESIRED BYTE IN LOW BYTE
2107  011464' 112004                    20$:    MOVB    (R0)+,R4        ;GET FLAG & LENGTH BYTE
2108  011466' 005704                            TST     R4              ;END OF THE CODE TBL?
2109  011470' 001445                            BEQ     60$             ;N,Y-60$
2110  011472' 122704  000377                    CMPB    #377,R4         ;GO TO NXT DEV REG WORD?
2111  011476' 001770                            BEQ     15$             ;N,Y-15$
2112  011500' 122704  000376                    CMPB    #376,R4         ;GO TO NXT BYTE IN DEV REG WORD?
2113  011504' 001766                            BEQ     17$             ;N,Y-17$
2114  011506' 032704  000100                    BIT     #100,R4         ;BIT VALUE OF 0 = AN ERROR CONDITION?
2115  011512' 001403                            BEQ     30$             ;Y,N-30$
2116  011514' 131005                            BITB    (R0),R5         ;THIS BIT RESET IN DEV REG BYTE?
2117  011516' 001407                            BEQ     40$             ;N,Y-40$
2118  011520' 000402                            BR      35$             ;GO TO NXT TBL ENTRY
2119  011522' 131005                    30$:    BITB    (R0),R5         ;THIS ERROR BIT SET IN DEV REG BYTE?
2120  011524' 001004                            BNE     40$             ;N,Y-40$
2121  011526' 042704  177770            35$:    BIC     #177770,R4      ;ISOLATE ENTRY LENGTH
2122  011532' 060400                            ADD     R4,R0           ;POINT AT NXT CODE TBL ENTRY
2123  011534' 000753                            BR      20$             ;GO CK FOR NXT CODE
2124  011536' 042704  177770            40$:    BIC     #177770,R4      ;ISOLATE I.D. NAME LENGTH + 1
2125  011542' 020416                            CMP     R4,(SP)         ;ENOUGH ROOM FOR NAME?
2126  011544' 101017                            BHI     60$             ;Y,N-60$
2127  011546' 060467  000050                    ADD     R4,70$          ;ADJ MSG LENGTH FOR NAME
2128  011552' 005304                            DEC     R4              ;ADJ FOR BIT MASK CHAR
2129  011554' 005200                            INC     R0              ;POINT PAST BIT MASK
2130  011556' 021627  000100                    CMP     (SP),#64.       ;FIRST ERROR CODE IN MSG?
2131  011562' 001403                            BEQ     50$             ;N,Y-50$
2132  011564' 112721  000054                    MOVB    #',,(R1)+       ;MOVE COMMA TO MSG
2133  011570' 005316                            DEC     (SP)            ;ADJ REMAINING ROOM IN MSG
2134  011572' 112021                    50$:    MOVB    (R0)+,(R1)+     ;MOVE ERROR CODE TO MSG
2135  011574' 005316                            DEC     (SP)            ;ADJ REMAINING ROOM IN MSG
2136  011576' 005304                            DEC     R4              ;MOVED ALL NAME CHARS?
2137  011600' 001374                            BNE     50$             ;Y,N-50$
2138  011602' 000730                            BR      20$             ;GO CK FOR MORE ERROR BITS
2139  011604' 005004                    60$:    CLR     R4              ;SET USER MODE PRINT
2140  011606' 022627  000100                    CMP     (SP)+,#64.      ;ANY ERROR CODES PUT IN MSG?
2141  011612' 001404                            BEQ     80$             ;Y,N-80$
2142  011614' 004567  001144                    JSR     R5,PRINT        ;GO ISSUE ERROR BITS MSG
2143  011620' 002404                            .WORD   DKEMSG-.
2144  011622' 000116                    70$:    .WORD   78.
2145  011624' 004567  000700            80$:    JSR     R5,DISPST       ;DISPLAY DEVICE REG'S
2146  011630' 000000                    ERSTAD: .WORD   XXXX
2147  011632' 004767  001056                    JSR     PC,PRTIHD       ;DISPLAY CYL,HEAD,SECT VALUES
2148  011636' 016300  000022            ERRSNM: MOV     PSRCST(R3),R0   ;GET ADR OF SRC STMNTS
2149  011642' 111001                    110$:   MOVB    (R0),R1         ;SAVE STMNT LENGTH
2150  011644' 026067  000004  170112            CMP     4(R0),ERRADR    ;ERROR OCCUR ON THIS STMNT?
2151  011652' 001402                            BEQ     120$            ;N,Y-120$
2152  011654' 060100                            ADD     R1,R0           ;POINT AT NXT STMNT
2153  011656' 000771                            BR      110$            ;GO CK NXT STMNT
2154  011660' 005720                    120$:   TST     (R0)+           ;SET UP ADR OF STMNT # DATA
2155  011662' 010701                            MOV     PC,R1           ;SET UP DATA OUTPUT ADR
2156  011664' 062701  002332                    ADD     #STMNUM-.,R1
```

```
2157  011670' 004577  166166                    JSR     R5,$DECASC              ;CONVERT IT TO ASCII
2158  011674' 012767  020040  002320            MOV     #20040,STMNUM+4         ;SET 2 LOW DIGITS TO SPACES
2159  011702' 004567  001056                    JSR     R5,PRINT                ;ISSUE STMNT # MSG
2160  011706' 002300                            .WORD   STMNMG-.
2161  011710' 177762                            .WORD   -14.
2162  011712' 012605                            MOV     (SP)+,R5                ;RESTORE R5 & R4
2163  011714' 012604                            MOV     (SP)+,R4
2164  011716' 000205                  ERREX:    RTS     R5                      ;EXIT IN-LINE
2165
2166
2167                                            ;ERROR MESSAGE CODE TABLE
2168
2169                                                    ;377 = GO TO NEXT DEVICE REGISTER WORD
2170                                                    ;376 = GO TO NEXT DEVICE REGISTER BYTE
2171                                                    ;
2172                                                    ;BYTE 0 CONTAINS FLAG BITS & I.D. NAME LENGTH
2173                                                    ;
2174                                                    ;       BITS 0-2 = LENGTH OF BIT MASK + I.D. NAME
2175                                                    ;       BIT 3 = RP04 ONLY ERROR BIT
2176                                                    ;       BIT 4 = RP05/RP06 ONLY ERROR BIT
2177                                                    ;       BIT 6 = BIT = 0 IS AN ERROR CONDITION
2178                                                    ;
2179                                                    ;BYTE 1 IS THE BIT MASK
2180                                                    ;
2181                                                    ;BYTES 2 THRU * ARE THE BIT'S ASCII I.D.
2182
2183  011720' 100005  042503  051122  ERCDTB: .ASCII  <005><200>/CERR/                ;RPCS1 - BYTE 1
2184  011726' 020005  050123  051101            .ASCII  <005><040>/SPAR/
2185  011734' 004004  052103     117            .ASCII  <004><010>/CTO/
2186  011741'    377                            .BYTE   377
2187  011742'    377                            .BYTE   377
2188  011743'    377                            .BYTE   377
2189  011744'    377                            .BYTE   377
2190  011745'    004  042200  052114            .ASCII  <004><200>/DLT/         ;RPCS2 - BYTE 1
2191  011752' 040004  041527     105            .ASCII  <004><100>/NCE/
2192  011757'    004  052440  042520            .ASCII  <004><040>/UPE/
2193  011764' 010004  042516     104            .ASCII  <004><020>/NED/
2194  011771'    004  047010  046505            .ASCII  <004><010>/NEM/
2195  011776' 002004  043520     105            .ASCII  <004><004>/PGE/
2196  012003'    004  046402  051504            .ASCII  <004><002>/MDS/
2197  012010' 000404  043125     105            .ASCII  <004><001>/UFE/
2198  012015'    377                            .BYTE   377
2199  012016' 004004  051127     114            .ASCII  <004><010>/WRL/         ;RPDS - BYTE 1
2200  012023'    376                            .BYTE   376
2201  012024' 100105  051104  054504            .ASCII  <105><200>/DRDY/        ;RPDS - BYTE 0
2202  012032' 040103  053126                     .ASCII  <103><100>/VV/
2203  012036' 020005  051104  052117            .ASCII  <005><040>/DROT/
2204  012044' 010004  051504     114            .ASCII  <004><020>/DSL/
2205  012051'    005  040410  046103            .ASCII  <005><010>/ACLO/
      012056'    117
2206  012057'    377                            .BYTE   377
2207  012060' 100004  041504     113            .ASCII  <004><200>/DCK/         ;RPER1 - BYTE 1
2208  012065'    004  052500  051516            .ASCII  <004><100>/UNS/
2209  012072' 020004  050117     111            .ASCII  <004><040>/OPI/
2210  012077'    004  042020  042524            .ASCII  <004><020>/DTE/
2211  012104' 004004  046127     105            .ASCII  <004><010>/WLE/
```

```
2212  012111'    005  044404  040504          .ASCII  <005><004>/IDAE/
      012116'    105
2213  012117'    004  041402  042517          .ASCII  <004><002>/COE/
2214  012124'  000405  053110  041522          .ASCII  <005><001>/HVRC/
2215  012132'    376                          .BYTE   376
2216  012133'    004  041200  042523          .ASCII  <004><200>/BSE/         ;RPER1 - BYTE 0
2217  012140'  040004  041505    110          .ASCII  <004><100>/ECH/
2218  012145'    005  042040  054524          .ASCII  <005><040>/DTYE/
      012152'    105
2219  012153'    005  043020  052115          .ASCII  <005><020>/FMTE/
      012160'    105
2220  012161'    006  042010  050122          .ASCII  <006><010>/DRPAR/
      012166'  051101
2221  012170'  002004  054116    106          .ASCII  <004><004>/NXF/
2222  012175'    004  051402  044513          .ASCII  <004><002>/SKI/
2223  012202'  000404  046111    106          .ASCII  <004><001>/ILF/
2224  012207'    377                          .BYTE   377
2225  012210'  100005  052101  033501          .ASCII  <005><200>/ATA7/        ;RPAS - BYTE 0
2226  012216'  040005  052101  033101          .ASCII  <005><100>/ATA6/
2227  012224'  020005  052101  032501          .ASCII  <005><040>/ATA5/
2228  012232'  010005  052101  032101          .ASCII  <005><020>/ATA4/
2229  012240'  004005  052101  031501          .ASCII  <005><010>/ATA3/
2230  012246'  002005  052101  031101          .ASCII  <005><004>/ATA2/
2231  012254'  001005  052101  030501          .ASCII  <005><002>/ATA1/
2232  012262'  000405  052101  030101          .ASCII  <005><001>/ATA0/
2233  012270'    000                          .BYTE   0                       ;TABLE TERMINATOR
2234          012272'                         .EVEN
```

```
                                        .SBTTL  SUBROUTINES FOR RK06 DEVICE ROUTINE


                                        ;SAVE REGISTERS R0 THRU R5

                                                ;JSR    R0,SAVREG       S/R CALL

012272' 010146                  SAVREG: MOV     R1,-(SP)                ;SAVE R0 THRU R5
012274' 010246                          MOV     R2,-(SP)
012276' 010346                          MOV     R3,-(SP)
012300' 010446                          MOV     R4,-(SP)
012302' 010546                          MOV     R5,-(SP)
012304' 010007                          MOV     R0,PC                   ;EXIT IN-LINE


                                        ;RESTORE REGISTERS R0 THRU R5

                                                ;JSR    R0,RESREG       S/R CALL

012306' 005726                  RESREG: TST     (SP)+                   ;RESTORE R5 THRU R0
012310' 012605                          MOV     (SP)+,R5
012312' 012604                          MOV     (SP)+,R4
012314' 012603                          MOV     (SP)+,R3
012316' 012602                          MOV     (SP)+,R2
012320' 012601                          MOV     (SP)+,R1
012322' 000200                          RTS     R0                      ;EXIT IN-LINE


                                        ;SET PROGRAM'S PROG TABLE ADR IN R3 & RPCS1 ADR IN R4

                                                ;JSR    PC,SUPTAD       S/R CALL

012324' 010703                  SUPTAD: MOV     PC,R3                   ;SET UP LOCATION ZERO ADR
012326' 062703  165452                  ADD     #LOCZ-.,R3
012332' 166303  177776                  SUB     -2(R3),R3               ;SUBTRACT PROG TBL LENGTH
012336' 016704  165462                  MOV     DREGAD,R4               ;GET DEV REG BASE ADR
012342' 000207                          RTS     PC                      ;EXIT IN-LINE


                                        ;STORE DEVICE'S STATUS REGISTERS

                                                ;JSR    R5,STSTAT       S/R CALL
                                                ;.WORD  STADR-.         REL STORAGE ADR
                                                ;DESTROYS R0,R1,R2

012344' 010501                  STSTAT: MOV     R5,R1                   ;GET REL STORAGE ADR & MAKE
012346' 062501                          ADD     (R5)+,R1                ;IT ABSOLUTE
012350' 016700  165450                  MOV     DREGAD,R0               ;GET DEV REG ADR
012354' 010046                          MOV     R0,-(SP)                ;SET UP ADR OF RPCS2 REG
012356' 062716  000010                  ADD     #RPCS2,(SP)             ;FOR LATER USE
012362' 012702  002412                  MOV     #STSLUP,R2              ;SETUP TWO LOOP COUNTS
012366' 012021                  10$:    MOV     (R0)+,(R1)+             ;STORE DEV REG
012370' 105302                          DECB    R2                      ;FINISHED WITH THIS GROUP OF REGS?
012372' 001375                          BNE     10$                     ;Y,N-10$
012374' 000302                          SWAB    R2                      ;SET UP NEXT LOOP CNT
```

```
2292  012376' 001417                          BEQ     30$                     ;DONE 2 PASSES? (N,Y-30$)
2293  012400' 010746                          MOV     PC,-(SP)                ;SET UP CURRENT STATUS
2294  012402' 062716  167246                  ADD     #CSTAT-.,(SP)           ;STORAGE ADR
2295  012406' 020126                          CMP     R1,(SP)+                ;STORING STATUS FOR INTERRUPT?
2296  012410' 101005                          BHI     15$                     ;Y,N-15$
2297  012412' 032736  000200                  BIT     #OR,@(SP)+              ;IS OUTPUT READY?
2298  012416' 001403                          BEQ     20$                     ;Y,N-20$
2299  012420' 012021                          MOV     (R0)+,(R1)+             ;STORE RPDB'S CONTENTS
2300  012422' 000761                          BR      10$                     ;GO DO SECOND PASS
2301  012424' 005726                  15$:    TST     (SP)+                   ;TAKE UNUSED ADR OFF STACK
2302  012426' 062700  000002          20$:    ADD     #2,R0                   ;BYPASS READ OF RPDB
2303  012432' 005021                          CLR     (R1)+                   ;SET ITS STORAGE TO 0'S
2304  012434' 000754                          BR      10$                     ;GO DO SECOND PASS
2305  012436' 000205                  30$:    RTS     R5                      ;EXIT IN-LINE
2306
2307                                          ;DISPLAY DEVICE I.D. & UNIT #
2308
2309                                                  ;JSR    PC,DEVID        S/R CALL
2310
2311                                                  ;R3 MUST CONTAIN PROG TBL ADR
2312                                                  ;DESTROYS R0,R1,R2
2313
2314  012440' 012767  000026  000056  DEVID:  MOV     #22.,DEVIML             ;INITIALIZE TO NORMAL MSG LNGTH
2315  012446' 116300  000035                  MOVB    PCURDV(R3),R0           ;GET CURR UNIT #
2316  012452' 020027  000007                  CMP     R0,#7                   ;VALID UNIT #?
2317  012456' 101007                          BHI     DEVIIV                  ;Y,N-DEVIIV
2318  012460' 004577  165374                  JSR     R5,@#BTASLZ             ;CONVERT # TO DECIMAL ASCII
2319  012464' 000674                          .WORD   UNASCI-.
2320  012466' 016767  000672  000664          MOV     UNASCI+4,UNPSCI         ;MOVE ASCII # TO 1ST TWO DIGITS
2321  012474' 000410                          BR      DEVIPR                  ;GO ISSUE MSG
2322  012476' 012767  000032  000020  DEVIIV: MOV     #26.,DEVIML             ;SET UP ERR COND MSG LNGTH
2323  012504' 042700  177400                  BIC     #177400,R0              ;RESET HIGH BYTE
2324  012510' 004577  165342                  JSR     R5,@#INASC              ;CONVERT BINARY # TO ASCII
2325  012514' 000644                          .WORD   UNASCI-.
2326  012516' 004567  000242          DEVIPR: JSR     R5,PRINT                ;GO ISSUE UNIT # MSG
2327  012522' 000612                          .WORD   UNITMG-.
2328  012524' 000026                  DEVIML: .WORD   22.
2329  012526' 000207                          RTS     PC                      ;EXIT IN-LINE
```

```
                                        ;TAILOR STATUS MSG & PRINT IT
                                        ;
                                        ;JSR    R5,DISPST       S/R CALL
                                        ;WORD   STATADR-.       REL ADR OF STATUS DATA
                                        ;DESTROYS R0,R1,R2
                                        ;
012530' 010346          DISPST: MOV     R3,-(SP)                ;SAVE R3
012532' 010503                  MOV     R5,R3                   ;GET REL DATA ADR
012534' 062503                  ADD     (R5)+,R3                ;MAKE IT ABS
012536' 010546                  MOV     R5,-(SP)                ;SAVE R5
012540' 010705                  MOV     PC,R5                   ;SET UP ADR OF REG NAMES IN ASCII
012542' 062705  165354          ADD     #DVREGS-.,R5
012546' 012746  000020          MOV     #REGNUM,-(SP)           ;SET UP # OF REGISTERS TO DISPLAY
012552' 012700  000003  10$:    MOV     #3,R0                   ;SET UP 3 REG LOOP CNT
012556' 010701                  MOV     PC,R1                   ;POINT AT REG NAME IN MSG
012560' 062701  000606          ADD     #DVRGMG-.,R1
012564' 012521          15$:    MOV     (R5)+,(R1)+             ;MOVE REG NAME TO MSG
012566' 012521                  MOV     (R5)+,(R1)+
012570' 005725                  TST     (R5)+                   ;POINT TO NEXT NAME
012572' 062701  000012          ADD     #10.,R1                 ;POINT TO NEXT FIELD IN MSG
012576' 005300                  DEC     R0                      ;DONE 3 REGS?
012600' 001371                  BNE     15$                     ;Y,N-15$
012602' 012300                  MOV     (R3)+,R0                ;CONVERT OCTAL REGISTER CONTENTS
012604' 004577  165246          JSR     R5,@#INASC              ;FOR 3 REGISTERS TO ASCII
012610' 000564                  .WORD   DVRDT1-.                ;AND PLACE IN THE MSG
012612' 012300                  MOV     (R3)+,R0
012614' 004577  165236          JSR     R5,@#INASC
012620' 000572                  .WORD   DVRDT2-.
012622' 012300                  MOV     (R3)+,R0
012624' 004577  165226          JSR     R5,@#INASC
012630' 000600                  .WORD   DVRDT3-.
012632' 012767  000050  000034  MOV     #40.,30$                ;INITIALIZE MSG LENGTH TO 3 REGS
012640' 162716  000003          SUB     #3,(SP)                 ;DECR REGISTER CNT
012644' 100005                  BPL     25$                     ;< 3 REGS? (Y,N-25$)
012646' 162767  000016  000020  20$:    SUB     #14.,30$        ;SHORTEN MSG LENGTH BY 1 REG
012654' 005216                  INC     (SP)                    ;INCR NEG REG CNT
012656' 100773                  BMI     20$                     ;CNT BACK TO 0? (Y,N-20$)
012660' 010346          25$:    MOV     R3,-(SP)                ;SAVE REG DATA PNTR
012662' 016603  000006          MOV     6(SP),R3                ;RESTORE PROG TBL ADR
012666' 004567  000072          JSR     R5,PRINT                ;GO PRINT THE MSG
012672' 000474                  .WORD   DVRGMG-.
012674' 000050          30$:    .WORD   40.
012676' 012603                  MOV     (SP)+,R3                ;RESTORE REG DATA PNTR
012700' 005716                  TST     (SP)                    ;MORE REGS TO GO?
012702' 001323                  BNE     10$                     ;N,Y-10$
012704' 005726                  TST     (SP)+                   ;REMOVE CNT FROM STACK
012706' 012605                  MOV     (SP)+,R5                ;RESTORE R5 & R3
012710' 012603                  MOV     (SP)+,R3
012712' 000205                  RTS     R5                      ;EXIT IN-LINE
```

```
                                        ;DISPLAY CYL/HEAD/SECT WORDS' VALUES
                                        ;
                                        ;       JSR     PC,PRTIWD       S/R CALL
                                        ;       DESTROYS R0,R1,R2
                                        ;
012714' 016700  165064          PRTIWD: MOV     CYL,R0                  ;GET CYL VALUE
012720' 004577  165132                  JSR     R5,@BINASC              ;CONVERT ITS VALUE TO ASCII
012724' 001221                          .WORD   IFCYL-.
012726' 016700  165054                  MOV     HEAD,R0                 ;GET & CONVERT HEAD VALUE
012732' 004577  165120                  JSR     R5,@BINASC
012736' 001224                          .WORD   IFHEAD-.
012740' 016700  165044                  MOV     SECT,R0                 ;GET & CONVERT SECT VALUE
012744' 004577  165106                  JSR     R5,@BINASC
012750' 001227                          .WORD   IFSECT-.
012752' 004567  000006                  JSR     R5,PRINT                ;PRINT MSG WITH THEIR VALUES
012756' 001162                          .WORD   INFOMG-.
012760' 000045                          .WORD   37.
012762' 000207                  RTS     PC                              ;EXIT IN-LINE
                                        ;
                                        ;ISSUE MSG TO LIST DEVICE SUBROUTINE
                                        ;
                                        ;       JSR     R5,PRINT        S/R CALL
                                        ;       .WORD   MSGADR-.        REL ADR OF MSG
                                        ;       .WORD   BYTCNT          MSG BYTE CNT (IF NEGATIVE,
                                        ;                                 RESET PRT DEV DEDICATED.)
                                        ;
                                        ;R3 = PROG TBL ADR
                                        ;R4 = FLAGWORD -- IF NEGATIVE, USE CMND MODE PRINT
                                        ;DESTROYS R0,R1,R2
                                        ;
012764' 010500          PRINT:  MOV     R5,R0                   ;GET MSG ADR & MAKE IT ABS
012766' 062500                  ADD     (R5)+,R0
012770' 012501                  MOV     (R5)+,R1                ;GET BYTE COUNT
012772' 005704                  TST     R4                      ;USE CMND MODE PRINT?
012774' 100030                  BPL     40$                     ;Y,N-40$
012776' 010702                  MOV     PC,R2                   ;SET UP LINK INFO ADR
013000' 062702  000040          ADD     #20$-.,R2
013004' 160200                  SUB     R2,R0                   ;MAKE MSG ADR REL
013006' 010022                  MOV     R0,(R2)+                ;STORE MSG ADR
013010' 010112                  MOV     R1,(R2)                 ;STORE MSG'S BYTE COUNT
013012' 100401                  BPL     10$                     ;CNT NEG? (Y,N-10$)
013014' 005412                  NEG     (R2)                    ;MAKE IT POSITIVE
013016' 016367  000006  000240  10$:    MOV     PASCIN(R3),PROGNM       ;STORE PROG'S # IN MSG
013024' 004577  165024                  JSR     R5,@CLIST               ;ISSUE PROG #
013030' 000232                          .WORD   PNMMSG-.
013032' 000005                          .WORD   5
013034' 004577  165014                  JSR     R5,@CLIST               ;ISSUE MSG SPECIFIED
013040' 000000          20$:    .WORD   XXXX
013042' 000000                  .WORD   XXXX
013044' 004577  165004                  JSR     R5,@CLIST               ;ISSUE A <CR> & <LF>
013050' 000440                          .WORD   CRLF-.
013052' 000002                          .WORD   2
013054' 000410                  BR      PRTEX                   ;GO TO EXIT
013056' 010067  000010          40$:    MOV     R0,50$                  ;STORE MSG'S ABS ADR
013062' 010167  000006                  MOV     R1,60$                  ;STORE ITS BYTE CNT
013066' 004577  164760                  JSR     R5,@ULIST               ;GO TO MPG TO ISSUE THE MSG
```

```
013072' 000000                  SOS:    .WORD   XXXX
013074' 000000                  SOS:    .WORD   XXXX
013076' 000205                  PRTEX:  RTS     R5              ;EXIT IN-LINE
                                ;
                                ;SUBROUTINE TO PRINT ON THE LIST DEVICE
                                ;CALL AS FOLLOWS
                                ;  JSR R5,PRINTX
                                ;  .WORD MSGADR-PRIX1
                                ;  .WORD BYTE COUNT
                                ;  RETURN
                                ;THIS SUBROUTINE DOES NOT DESTROY R3 & R4
                                ;
013100' 010446                  PRINTX: MOV     R4,-(SP)
013102' 010346                          MOV     R3,-(SP)
013104' 004767  177214                  JSR     PC,SUPTAD       ;SETUP R3
013110' 012704  000000                  MOV     #0,R4           ;  AND R4 FOR USER MODE
013114' 011567  000010                  MOV     (R5)+,PRIX1     ;RELOCATE ADDRESSES
013120' 012567  000006                  MOV     (R5)+,PRIX2
013124' 004567  177634                  JSR     R5,PRINT        ;GO TO MPG PRINT
013130' 000000                  PRIX1:  .WORD   XXXX
013132' 000000                  PRIX2:  .WORD   XXXX
013134' 012603                          MOV     (SP)+,R3
013136' 012604                          MOV     (SP)+,R4
013140' 000205                          RTS     R5
                                ;
                                ;SUBROUTINE TO SELECT A DRIVE
                                ;CALL AS FOLLOWS
                                ;  PUT  MESSAGE SELECT CODE IN R5 (R4 MUST = BASE ADDR)
                                ;  JSR PC,SD
                                ;  RETURN
                                ;
013142' 050564  000026          SD:     BIS     R5,RPMR1(R4)    ;SET UP MESSAGE SELECT CODE
013146' 012764  000001  000000          MOV     #SDCODE,RPCS1(R4)       ;SELECT DRIVE
013154' 004767  000002                  JSR     PC,DRVDLY       ;WAIT FOR DRIVE
013160' 000207                          RTS     PC
                                ;
                                ;SUBROUTINE TO WAIT FOR DRIVE DELAY TIME
                                ;CALL WITH JSR PC,DRVDLY
                                ;
013162' 010546                  DRVDLY: MOV     R5,-(SP)
013164' 012705  000140                  MOV     #96.,R5
013170' 032714  000200          DRV1:   BIT     #RDY,(R4)       ;IS CONTROLLER READY
013174' 001002                          BNE     DRV2            ;YES, EXIT
013176' 005305                          DEC     R5
013200' 001373                          BNE     DRV1
013202' 012605                  DRV2:   MOV     (SP)+,R5
013204' 000207                          RTS     PC              ;DEFAULT IF CONTRL NOT RDY
                                ;
                                ;SUBROUTINE TO ISSUE CONTROLLER CLEAR
                                ;SUBROUTINE SAVES AND RESTORES RXCS2, RXBA, AND RXDA
                                ;R4 MUST HAVE BASE ADDRESS OF DISK ADDRESSES
                                ;CALL WITH JSR PC,CTRLCL
                                ;
013206' 016446  000010          CTRLCL: MOV     RPCS2(R4),-(SP) ;SAVE DISK PARAMATERS
013212' 016446  000004                  MOV     RPBA(R4),-(SP)
013216' 016446  000006                  MOV     RPDA(R4),-(SP)
```

```
2493  013222' 052714  100000          BIS     #CCLR,(R4)              ;CLEAR CONTROLLER
2494  013226' 004767  177730          JSR     PC,DRVDLY
2495  013232' 012664  000006          MOV     (SP)+,RPDA(R4)          ;RESTORE DISK PARAMATERS
2496  013236' 012664  000004          MOV     (SP)+,RPBA(R4)
2497  013242' 012664  000010          MOV     (SP)+,RPCS2(R4)
2498  013246' 000207                  RTS     PC
2499
2500                                  ;SUBROUTINE TO ISSUE DRIVE CLEAR
2501                                  ;CALL WITH JSR PC,DRVCLR
2502                                  ;R4 MUST CONTAIN THE BASE ADDRESS OF THE DISK REGISTERS
2503                                  ;
2504  013250' 012714  000005  DRVCLR: MOV     #DCCODE,(R4)            ;CLEAR DRIVE
2505  013254' 004767  177702          JSR     PC,DRVDLY
2506  013260' 000207                  RTS     PC
```

```
                                                .SBTTL  RK06 MESSAGE STORAGE AREA
                                        ;
                                        ;M E S S A G E   S T O R A G E   A R E A
                                        ;


                                                .NLIST  BEX

                                                .EVEN
013262' 021520                          PNAMSG: .ASCII  /Pa/
013264' 054130      011                 PROGNM: .ASCII  /XX/<011>
013267'    101  020124  040514          ATIMSG: .ASCII  /AT LAST INT:/
013303'    103  051125  042522          CURMSG: .ASCII  /CURRENTLY:/
013315'    105  042116  047440          RENDMG: .ASCII  /END OF REPORT/<15><12>
                                                .EVEN
013334' 025052  025052  045522          UNITMG: .ASCII  /####RK06 DISK UNIT: /
013360' 054130  054130  054130          UNASCI: .ASCII  /XXXXXX/
                                                .EVEN
013366' 054130  054130  020075          DVRGMG: .ASCII  /XXXX= /
013374' 054130  054130  054130          DVROT1: .ASCII  /XXXXXX  XXXX= /
013412' 054130  054130  054130          DVROT2: .ASCII  /XXXXXX  XXXX= /
013430' 054130  054130  054130          DVROT3: .ASCII  /XXXXXX/
013436' 054502  042524  035123          CNTSMG: .ASCII  /BYTES:  RD= /
013452' 054130  054130  054130          BCMRD:  .ASCII  /XXXXXXXXXXXX  WR= /
013474' 054130  054130  054130          BCMWR:  .ASCII  /XXXXXXXXXXXX/
013510' 005015                          CRLF:   .ASCII  <015><012>
013512' 004411  045503  020075                  .ASCII  <011><011>/CK= /
013520' 054130  054130  054130          BCMCK:  .ASCII  /XXXXXXXXXXXX/<015><012><011>/CMNDS:  RD= /
013553'    130  054130  054130          CMDCRD: .ASCII  /XXXXXX WR= /
013566' 054130  054130  054130          CMDCWR: .ASCII  /XXXXXX CK= /
013601'    130  054130  054130          CMDCCK: .ASCII  /XXXXXX/<015><012><011><011>/SK= /
013617'    130  054130  054130          CMDCSK: .ASCII  /XXXXXX MISC= /
013634' 054130  054130  054130          CMDERS: .ASCII  /XXXXXX/<015><012><011>/ERRORS: DEV= /
013662' 054130  054130  054130          CNTDER: .ASCII  /XXXXXX CORR ECC= /
013703'    130  054130  054130          CNTCEC: .ASCII  #XXXXXX DATA/OPR= #
013724' 054130  054130  054130          CNTDER: .ASCII  /XXXXXX/<015><012><011>/RETRYS: DLT= /
013752' 054130  054130  054130          CNTDLT: .ASCII  /XXXXXX DTE= /
013776' 054130  054130  054130          CNTDTE: .ASCII  /XXXXXX HVRC= /
014003'    130  054130  054130          CNTHCR: .ASCII  /XXXXXX/<015><012><011><011>/FER= /
014022' 054130  054130  054130          CNTFER: .ASCII  /XXXXXX DCK= /
014036' 054130  054130  054130          CNTDCK: .ASCII  /XXXXXX/<015><012><011><011>/WCE= /
014055'    130  054130  054130          CNTWCE: .ASCII  /XXXXXX/<015><012><011>/TOTAL RETRYS: /
014104' 054130  054130  054130          CNTRTY: .ASCII  /XXXXXX/<015><012><011>/INTERRUPTS:  /
014132' 054130  054130  054130          CNTINT: .ASCII  /XXXXXX/
                    014140'             CNTSEM= .
014140' 054503  036514     040          INFOMG: .ASCII  /CYL= /
014145'    130  054130  054130          IFCYL:  .ASCII  /XXXXXX HEAD= /
014162' 054130  054130  054130          IFHEAD: .ASCII  /XXXXXX SECT= /
014177'    130  054130  054130          IFSECT: .ASCII  /XXXXXX/
                    014206'                     .EVEN
014206' 052123  047115  020124          STMTMG: .ASCII  /STMNT # /
014216' 054130  054130  054130          STMNUM: .ASCII  /XXXXXX/
014224' 051105  047522  020122          DKEMSG: .ASCII  /ERROR BITS:/<015><012><011>
014242' 000102                          CODFLD: .BLKB   #2.
014344' 042502  047506  042522          BEFIO:  .ASCII  'BEFORE ISSUING I/O CMND:'
014374' 043101  042524  020122          AFTIO:  .ASCII  'AFTER ISSUING I/O CMND:'
```

```
2564  014423'    124  047457  047440  CRTO:   .ASCIZ  'I/O ON CRFSET'
2565  014441'    124  047457  047440  SRTO:   .ASCIZ  'I/O ON SRESET'
2566  014457'    124  047457  047440  RTO:    .ASCIZ  'I/O ON REL'
2567  014472'  044524  042515  042517  IOTO:   .ASCIZ  'TIMEOUT ON I/O'
2568  014511'    116  047117  042505  NOITER: .ASCII  'NON-INT '
2569  014521'    111  047457  042440  IOTERM: .ASCIZ  'I/O TERMINATION ERROR'
2570  014547'    111  052116  042440  NOATA:  .ASCIZ  'INT WITHOUT ATA'
2571  014567'    111  052116  043440  NODI:   .ASCIZ  'INT WITHOUT DI'
2572  014606'  047125  054105  042120  UXPATA: .ASCIZ  'UNEXP ATA COND'
2573  014625'    105  044130  042501  RTYEXH: .ASCIZ  'EXHAUSTED RETRIES'
2574  014647'    111  053116  042440  INVPAT: .ASCIZ  'INV ECC BIT PATTERN'
2575  014673'    111  053116  042440  INVPOS: .ASCIZ  'INV ECC BIT POSITION'
2576  014720'  040503  020116  047516  CRLT:   .ASCIZ  'CAN NOT READ LAST TRACK'
2577  014750'  047122  047113  053517  UNKERR: .ASCIZ  'UNKNOWN ERROR CONDITION'
2578  015000'  053123  046101  041040  NSVAL:  .ASCIZ  'SVAL BIT NOT SET IN RKDS'
2579  015031'    120  047522  051107  PABRA:  .ASCIZ  'PROGRAM ABORT:  HAVE DISK ADDRESS WHICH WOULD CHANGE LAST TRACK'
2580  015131'    117  052125  052520  ONRDY:  .ASCIZ  'OUTPUT NOT READY ON RDMD COMMAND'
2581                                    ;
2582                                    ;MESSAGES WHICH FOLLOW WILL NOT HAVE DEVICE REGISTERS PRINTED
2583                                    ;
2584  015172'  047111  020126  047125  INVDVN: .ASCIZ  /INV UNIT #/
2585  015205'    123  046105  051104  ISDMC:  .ASCIZ  'SELDRI COMMAND HAS INVALID CODE'
2586  015245'    122  047524  020117  RIOTS:  .ASCII  'RDIO IS TOO SMALL FOR "BADSEC" COMMAND'<15><12>
2587  015315'    011  030465  042062          .ASCIZ  <11>'512(DECIMAL) BYTES ARE NEEDED'
2588  015354'  040502  020104  042523  BSH:    .ASCII  /BAD SECTORS (OCTAL DATA)/
2589          015404'                  BSHSX=.
2590  015404'  040520  045503  021440  PNUM:   .ASCII  /PACK #/
2591  015412'  000006                  PNUM1:  .BLKB   6
2592  015420'  000006                  PNUM2:  .BLKB   6
2593  015426'     015     012                  .BYTE   15,12
2594          015430'                  PNUMSX=.
2595  015430'  020040  041440  046131  BHEAD:  .ASCII  /   CYL,  HEAD,  SECT/
2596          015454'                  BHEXSX=.
2597  015454'  047516  042516  005015  NONE:   .ASCII  /NONE/<15><12><12>
2598          015463'                  NONESX=.
2599  015463'     124  044510  020123  TIMAP:  .ASCIZ  /THIS IS AN ALIGNMENT PACK/<15><12>
2600  015517'     105  042116  047440  EOBS:   .ASCII  /END OF BAD SECTORS/<15><12><12>
2601          015544'                  EOBSX=.
2602                                           .EVEN
2603                                           .LIST   BEX
2604
2605          015544'                  DVREND= .
```

```
                                .SBTTL  FORMATS FOR PROGRAM & DEVICE ROUTINE TABLES

                        ;       PROGRAM TABLE FORMAT

000242                  PTLGTH= 162.    ;PROGRAM TABLE LENGTH - NON MEM MGMNT VERSION OF MPG

                        ;(PTLGTH= 212.  ;PROGRAM TABLE LENGTH - MEM MGMNT VERSION OF MPG)

000000                  PFLGWD= +0.     ;PROGRAM FLAG WORD - 1 WORD

000002                          URSTOP= 2       ; 1 = USER HAS STOPPED THIS PROGRAM
000004                          ERSTOP= 4       ; 1 = AN E   R HAS STOPPED THIS PROGRAM
000010                          WT4IOT= 10      ; 1 = W IT  G FOR I/O TERMINATION
000020                          CTPRIO= 20      ; 1 = C     E OR PRINTER I/O IN PPOGRESS
000040                          SETDD= 40       ; 1 = THIS F  G SET T E PRT DEV DEDICATED FLAG
000100                          OCP  S= 100     ; 1 = OBJ CO E IS PRE ENT
000200                          USE  M= 200     ; 1 = THIS P G USES THE UNIBUS MAP (MEM MGMNT ONLY)
100000                          ACTI E= 100000  ; 1 = PROGRAM IS ACTIVE (SPECIFIED FOR EXECUTION)

000002                  POPSW=  +2.     ;PROGRAM'S OPERATION SWITCHES - 1 WORD

100000                          STOMER= 10    0  ; 1 = STOP PROG EXECUTION UPON ERROR
040000                          CYCPRG= 40    0  ; 1 = CYCLE PROGRAM (ON CURRENT DEVICE)
020000                          PRNMER= 20000   ; 1 = DO NOT PRINT ON ERROR
010000                          BIT12=  10000   ; 0 = NOT USED
004000                          BIT11=  4000    ; 0 = NOT U ED
002000                          CYCDVL= 2000    ; 1 = CYCLE THE DEVICE LIST
001000                          GTNXTD= 1000    ; 1 = CYCLE ON S E DEVICE UPON ERROR
000400                          DOERCK= 400     ; 1 = DON'T DO   R CHECKING
000200                          SPOPER= 200     ; 1 = DEVICE SPECIAL OPERATION
000100                          BIT6=   100     ; 0 = NOT USED
000040                          DOIOT=  40      ; 1 = DO NOT PERFORM I/O TIMEOUT
000020                          AUTORP= 20      ; 1 = DO NOT AUTOMATICALLY DISPLAY COUNTS
000010                          AUR  P= 10      ; 1 = AUTO DISPLAY COUNTS AT END OF FINAL PASS ONLY
000004                          HS EP= 4        ; 1 = H  EKEEP COUNTS ONLY AT RUN COMMAND
000002                          PF  OV= 2       ; 1 = PRINT FIRST BAD BYTE ONLY ON VERIFY
000001                          NOCOMP= 1       ; 1 = DO NOT PRINT PROG COMPLETED MSG

000004                  PFWADR= +4.     ;#;PROGRAM FLAGWORD ADDRESS - 1 WORD

000006                  PASCIN= +6.     ;PROGRAM'S NUMBER IN ASCII - 1 WORD

000010                  PNAME=  +8.     ;PROGRAM'S NAME IN ASCII - 6 BYTES

000016                  PRDIOA= +14.    ;ADDRESS OF READ I/O AREA - 1 WORD

000020                  PWRIOA= +16.    ;ADDRESS OF WRITE I/O AREA - 1 WORD

000022                  PSRCST= +18.    ;SOURCE STATEMENTS START ADDRESS - 1 WORD

000024                  POBJST= +20.    ;OBJECT CODE START ADDRESS - 1 WORD

000026                  PLNGTH= +22.    ;PROG AREA LENGTH (OBJ END MINUS PROG TBL START) - 1 WORD

000030                  PTOCNT= +24.    ;I/O TIMEOUT COUNT - 1 WORD
```

```
2663    000032                  PMDLCD= +26.    ;DEV ROUT MODEL # CODE - 1 WORD
2664
2665    000034                  PDPNTR= +28.    ;CURRENT DEVICE NUMBER POINTER - 1 BYTE
2666
2667    000035                  PCURDV= +29.    ;CURRENT DEVICE # - 1 BYTE
2668
2669    000036                  PDNUMS= +30.    ;DEVICE NUMBERS - 16 BYTES
2670
2671    000056                  PTEMJ=  +46.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2672
2673    000060                  PTEM1=  +48.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2674
2675    000062                  PTEM2=  +50.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2676
2677    000064                  PTEM3=  +52.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2678
2679    000066                  PTEM4=  +54.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2680
2681    000070                  PTEM5=  +56.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2682
2683    000072                  PTEM6=  +58.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2684
2685    000074                  PTEM7=  +60.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2686
2687    000076                  PTEM8=  +62.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2688
2689    000100                  PTEM9=  +64.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2690
2691    000102                  PTEM10= +66.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2692
2693    000104                  PTEM11= +68.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2694
2695    000106                  PTEM12= +70.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2696
2697    000110                  PTEM13= +72.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2698
2699    000112                  PTEM14= +74.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2700
2701    000114                  PTEM15= +76.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
2702
2703    000116                  PNBR=   +78.    ;NUMBER OF BYTES TO TRANSFER ON MOVE (NBR) - 1 WORD
2704
2705    000120                  PSRC=   +80.    ;DATA SOURCE ADDRESS ON MOVE (SRC) - 1 WORD
2706
2707    000122                  PDST=   +82.    ;DATA DESTINATION ADDRESS ON MOVE (DST) - 1 WORD
2708
2709    000124                  PSTKCT= +84.    ;# OF WORDS (X 2) SAVED OFF STACK - 1 WORD
2710
2711    000126                  PSTKSV= +86.    ;STACK WORDS STORAGE AREA - 30 WORDS
2712
2713    000222                  PSVREG= +146.   ;USER'S R0 THRU R5 REGISTERS STORAGE AREA - 6 WORDS
2714
2715    000236                  PUSRPC= +158.   ;USER'S CURRENT PROGRAM COUNTER - 1 WORD
2716
```

```
2718                                    ;FOLLOWING ENTRIES (PRDIOX THRU PUBMAP) ARE ONLY IN MEM MGMNT VERSION
2719
2720            000240                  PRDIOX= +160. ;18/22 BIT ABSOLUTE ADDRESS OF READ I/O AREA - 2 WORDS)
2721
2722            000244                  PRDIOV= +164. ;18 BIT VIRTUAL ADDRESS OF READ I/O AREA - 2 WORDS)
2723
2724            000250                  PWRIOX= +168. ;18/22 BIT ABSOLUTE ADDRESS OF WRITE I/O AREA - 2 WORDS)
2725
2726            000254                  PWRIOV= +172. ;18 BIT VIRTUAL ADDRESS OF WRITE I/O AREA - 2 WORDS)
2727
2728                                    ;(PUPARS= +176. ;STORAGE AREA FOR USER'S PAR'S 0 THRU 7 - 8 WORDS)
2729
2730                                    ;(PUPDRS= +192. ;STORAGE AREA FOR USER'S PDR'S 0 THRU 7 - 8 WORDS)
2731
2732                                    ;(PUBMAP= +208. ;1ST UNIBUS MAP REG # AND # OF REGS USED - 1 WORD)
2733
2734                                            ;END OF MEM MGMNT ONLY ENTRIES
2735
2736            000240                  PTSIZE= +160.   ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - NON MEM MGMNT
2737
2738                                    ;(PTSIZE= +210. ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - MEM MGMNT VERSION)
2739
2740            000242                  PTEND=  +162.   ;END OF PROGRAM TABLE - NON MEM MGMNT VERSION
2741
2742                                    ;(PTEND=  +212. ;END OF PROGRAM TABLE - MEM MGMNT VERSION)
```

```
                        ;       DEVICE ROUTINE TABLE
                        ;
000116                  DRTLTH= 78.     ;DEVICE ROUTINE TABLE LENGTH
                        ;
000000                  DEVRSZ= +0.     ;DEVICE ROUTINE SIZE IN BYTES - 1 WORD
000002                  DEVFWD= +2.     ;DEVICE ROUTINE FLAGWORD - 1 WORD
000004                  DEVIW1= +4.     ;DEVICE INTERFACE WORD # 1 - 1 WORD
000006                  DEVIW2= +6.     ;DEVICE INTERFACE WORD # 2 - 1 WORD
000010                  DEVIW3= +8.     ;DEVICE INTERFACE WORD # 3 - 1 WORD
000012                  DEVIW4= +10.    ;DEVICE INTERFACE WORD # 4 - 1 WORD
000014                  DEVIW5= +12.    ;DEVICE INTERFACE WORD # 5 - 1 WORD
000016                  DEVIW6= +14.    ;DEVICE INTERFACE WORD # 6 - 1 WORD
000020                  DEVIW7= +16.    ;DEVICE INTERFACE WORD # 7 - 1 WORD  (SIZE)
000022                  DEVIW8= +18.    ;DEVICE INTERFACE WORD # 8 - 1 WORD  (ERR)
000024                  DEVDRA= +20.    ;DEVICE REGISTERS ADDRESS - 1 WORD
000026                  DEVIVA= +22.    ;DEVICE INTERRUPT VECTOR ADDRESS - 1 WORD
000030                  DEVRPS= +24.    ;DEVICE READ PROCESSOR STATUS WORD (BUS REQ) - 1 WORD
000032                  DEVWPS= +26.    ;DEVICE WRITE PROC STATUS WORD (BUS REQ) - 1 WORD
000034                  DHKPAD= +28.    ;DEVICE ROUT HOUSEKEEPING ROUT REL ENTRY ADR - 1 WORD
000036                  DERPAD= +30.    ;DEVICE ROUT REPORT ROUT REL ENTRY ADR - 1 WORD
000040                  DKILAD= +32.    ;DEVICE ROUT KILL ROUTINE REL ENTRY ADR - 1 WORD
000042                  DECTAD= +34.    ;DEVICE ROUT ERROR COUNTER REL ADR - 1 WORD
000044                  DTOEAD= +36.    ;DEVICE ROUT TIMEOUT ERR ROUT REL ENTRY ADR - 1 WORD
000046                  DEVIOB= +38.    ;DEVICE I/O BUSY BRANCH ADDRESS (CIOBSY) - 1 WORD
000050                  DEVDER= +40.    ;DEVICE ERROR BRANCH ADDRESS (CUPGER) - 1 WORD
000052                  DVUPRT= +42.    ;USER MODE PRINT BRANCH ADDRESS (ULIST) - 1 WORD
000054                  DVCPRT= +44.    ;CMND MODE PRINT BRANCH ADDRESS (CLIST) - 1 WORD
000056                  DEVBTA= +46.    ;CONVERT BINARY TO ASCII BR ADR (BINASC) - 1 WORD
000060                  DVBTDA= +48.    ;CONVERT BINARY TO DECIMAL ASCII BR ADR (BTASLZ) - 1 WORD
```

```
2800            000062                  DVPDTA= +50.    ;CONVERT PACKED DECIMAL TO ASCII BR ADR (DECASC) - 1 WORD
2801
2802            000064                  DVSFWD= +52.    ;MPG SYSTEM FLAGWORD ADDRESS (CSYSFW) - 1 WORD
2803
2804            000066                  DVSVEC= +54.    ;SET INTERRUPT VECTOR BR ADR (SETVEC) - 1 WORD
2805
2806            000070                  DVCVEC= +56.    ;CLEAR INTERRUPT VECTOR BR ADR (CLRVEC) - 1 WORD
2807
2808            000072                  DVTVEC= +58.    ;TEST INTERRUPT VECTOR BR ADR (TSTVEC) - 1 WORD
2809
2810            000074                  DVRINT= +60.    ;RETURN FROM INTERRUPT BR ADR (RTNINT) - 1 WORD
2811
2812            000076                  DVGETB= +62.    ;GET DATA BYTE BR ADR (GETBYT) - 1 WORD
2813
2814            000100                  DVPUTB= +64.    ;PUT DATA BYTE BR ADR (PUTBYT) - 1 WORD
2815
2816            000102                  DEVSTP= +66.    ;DEVICE ROUT REL SYMBOL TABLE POINTER - 1 WORD
2817
2818            000104                  DEVETP= +68.    ;DEVICE ROUT REL ENTRY TABLE POINTER - 1 WORD
2819
2820            000106                  DVPTEP= +70.    ;PACK TABLE EXTEN. REL POINTER - 1 WORD
2821
2822            000110                  DVVTEP= +72.    ;VECTOR TABLE EXTEN. REL POINTER - 1 WORD
2823
2824            000112                  DVCTEP= +74.    ;COMPILER TBL EXTEN. REL POINTER - 1 WORD
2825
2826            000114                  DVIWSP= +76.    ;DEVICE INTERFACE WORD SYMBOL TBL REL POINTER - 1 WORD
2827
2828            000116                  DRTEND= +78.    ;END OF DEVICE ROUTINE TABLE
2829
2830
2831            000001                          .END
```

```
ACLO   = 000010          CECCER  001740R 002  CTRLCL  013206R 002  DOCORR  010452R 002  DWN2    003412R 002
ACQERR  004334R 002      CFMT   = 010000      CUPGER  000050R 002  DOENCX= 000400        DWN2    003422R 002
ACQHSK  004214R 002      CIOBSY  000046R 002  CURADR  011744R 002  DOIOT  = 000040        DWMM    003362R 002
ACQRTY  004216R 002      CKCNT   001730R 002  CURCHD  011772R 002  DOTERM= 000002        EBSBAS  011442R 002
ACTIVE= 100000           CKCORR  010172R 002  CURCNT  002000R 002  DRA    = 000001        EBSTAT  011444R 002
AFTIO   014374R 002      CKDBSY  010670R 002  CURFLG  011770R 002  DRDY   = 000200        ECH    = 000100
ANYIOI= 000200           CKRTRY  007730R 002  CURMSG  013303R 002  DREGAD  000024R 002  EOBS    015517R 002
ATATBL  004364R 002      CKRI    010124R 002  CURPBC  002002R 002  DRESET  003702R 002  EOBSX  = 015544R 002
ATIMSG  013267R 002      CKSPIN  002612R 002  CURPSW  002012R 002  DRIB   = 000001        ERCDTB  011720R 002
AURPEP= 000010           CLIST   000054R 002  CURRTY  002006R 002  DROT   = 000040        ERMBAS  011410R 002
AUTORP= 000020           CLRCOD  006706R 002  CYCOVL= 002000        DRPAR  = 000010        ERRADR  001764R 002
BADSEC  005564R 002      CLRCI   006720R 002  CYCPNG= 040000        DRTEND= 000116        ERRCNT  001736R 002
BAD1    005660R 002      CLRVEC  000070R 002  CYL     000004R 002  DRTLTH= 000116        ERRCOM  011250R 002
BAD11   005734R 002      CMDCCK  013601R 002  DATAER  001742R 002  DRVCLR  013250R 002  ERRCS   011210R 002
BAD12   005716R 002      CMDCMS  013634R 002  DCCODE= 000005        DRVDLY  013162R 002  ERRCS1  011216R 002
BAD13   005746R 002      CMDCOM  005116R 002  DCK    = 100000        DRV1    013170R 002  ERREX   011716R 002
BAD14   006000R 002      CMDCRD  013553R 002  DCKCNT  001754R 002  DRV2    013202R 002  ERRI    000022R 002
BAD141  006032R 002      CMDCSK  013617R 002  DECASC  000062R 002  DSC    = 040000        ERRIS   011234R 002
BAD142  006040R 002      CMDCWR  013566R 002  DECIAD= 000042        DSL    = 000020        ERRSNM  011636R 002
BAD2    006160R 002      CMDISU= 000100       DERPAD= 000036        DTE    = 010000        ERSTAD  011630R 002
BAD3    006232R 002      CMDI    005122R 002  DEVBTA= 000056        DTECNT  001746R 002  ERSTOP= 000004
BAD32   006350R 002      CNTADR  001766R 002  DEVDER= 000050        DTOEAD= 000044        EVEN    003464R 002
BAD32X  006376R 002      CNTCEC  013703R 002  DEVDRA= 000024        DTYE   = 000040        FERCNT  001752R 002
BAD32Y  007436R 002      CNTDCK  014036R 002  DEVETP= 000104        DVBTDA= 000060        FINCNT  002004R 002
BAD32Z  007500R 002      CNTDER  013724R 002  DEVFWD= 000002        DVCMDS  000256R 002  FMTE   = 000020
BAD321  006326R 002      CNTDLT  013752R 002  DEVID   012440R 002  DVCPRT= 000054        FMT20   003544R 002
BAD322  007566R 002      CNTDTE  013766R 002  DEVIIV  012476R 002  DVCPTE  001210R 002  FMT22   003534R 002
BAD33   007600R 002      CNTEND= 001764R 002  DEVIML  012524R 002  DVCTEP= 000112        GETBYT  000076R 002
BAD331  007634R 002      CNTERR  013662R 002  DEVIOB= 000046        DVCVEC= 000070        GO     = 000001
BAD4    006656R 002      CNTFER  014022R 002  DEVIPR  012516R 002  DVGETB= 000076        GTNXTD= 001000
BAD41   007666R 002      CNTHCR  014003R 002  DEVIVA= 000026        DVIWSP= 000114        HARDER  010150R 002
BAI    = 000020          CNTINT  014132R 002  DEVIW1= 000004        DVIWST  001522R 002  HCRCNT  001750R 002
BAIOFF  003504R 002      CNTNUM= 000026       DEVIW2= 000006        DVMVTE  001024R 002  HEAD    000006R 002
BAION   003474R 002      CNTRTY  014104R 002  DEVIW3= 000010        DVPDTA= 000062        HSKEEP  002076R 002
BCMCK   013520R 002      CNTSEN= 014140R 002  DEVIW4= 000012        DVPKTE  000454R 002  HSKPEN= 002014R 002
BCMRD   013452R 002      CNTSMG  013436R 002  DEVIW5= 000014        DVPTEP= 000106        HSKPEP= 000004
BCMWR   013474R 002      CNTWCE  014055R 002  DEVIW6= 000016        DVPUTB= 000100        HSKPST= 001610R 002
BEFIO   014344R 002      COOFLD  014242R 002  DEVIW7= 000020        DVROT1  013374R 002  HVRC   = 000400
BINASC  000056R 002      COE    = 001000       DEVIWB= 000022        DVROT2  013412R 002  ICS1    001610R 002
BIT11  = 004000          CORFLG= 002000       DEVRPS= 000030        DVROT3  013430R 002  ICS2    001620R 002
BIT12  = 010000          COROFF  003524R 002  DEVRSZ= 000000        DVRSIZE= 000256R 002  IDAE   = 002000
BIT6   = 000100          CORON   003514R 002  DEVSTP= 000102        DVREGS  000116R 002  IE     = 000100
BSE    = 000200          COUNTS  001710R 002  DEVWPS= 000032        DVREND= 015544R 002  IFCYL   014145R 002
BSH     015354R 002      CPU70  = 000010       DFLGWD  000002R 002  DVREX   002436R 002  IFHEAD  014162R 002
BSHEAD  015430R 002      CRESET  003554R 002  DHKPAD= 000034        DVRGMG  013366R 002  IFSECT  014177R 002
BSHEX  = 015454R 002     CREI    003576R 002  DI     = 040000        DVRINT= 000074        ILF    = 000001
BSHX   = 015404R 002     CRLF    013510R 002  DIAFLG  002022R 002  DVSFWD= 000064        INFOMG  014140R 002
BTASLZ  000060R 002      CRLT    014720R 002  DISCNT  002336R 002  DVSVEC= 000066        INTCNT  001762R 002
BUSMAP= 170200           CRTO    014423R 002  DISPST  012530R 002  DVTVEC= 000072        INTEAD  011102R 002
BYCK    001720R 002      CSTAT   001650R 002  DKEMSG  014224R 002  DVUPRT= 000052        INVDVN  015172R 002
BYRD    001710R 002      CSYSFW  000064R 002  DKILAD= 000040        DVVTEP= 000110        INVPAT  014647R 002
BYWR    001714R 002      CTO    = 004000       DLT    = 100000        DWNSUB  003356R 002  INVPOS  014673R 002
CCLR   = 100000          CTPRIO= 000020       DLTCNT  001744R 002  DWN1    003374R 002  IOERR  = 000001
```

```
IOTERM  014521R  002   OCPRES= 000100          PSVREG= 000222          RHP24   007140R  002   R2    =%000002
IOTO    014472R  002   OOO     003454R  002   PSWD    000030R  002   RHP3    007410R  002   R3    =%000003
IRKAS   001626R  002   OFFSET  005000R  002   PTEMO = 000056          RHP31   007456R  002   R4    =%000004
IRKBA   001614R  002   ONARA   015131R  002   PTEM1 = 000060          RHP311  007472R  002   R5    =%000005
IRKDB   001634R  002   OPI   = 020000          PTEM10= 000102          RHP312  007516R  002   SAVREG  012272R  002
IRKDS   001622R  002   OR    = 000200          PTEM11= 000104          RHP313  007416R  002   SCLR  = 000040
IRKER   001624R  002   OTHATA  004406R  002   PTEM12= 000106          RHP314  007532R  002   SCODE = 000017
IRKPA   001642R  002   PACODE= 000003          PTEM13= 000110          RHP315  007442R  002   SD      013142R  002
IRKTA   001640R  002   PAHBA   015031R  002   PTEM14= 000112          RHP4    007562R  002   SDCODE= 000001
IRKWC   001612R  002   PAKACK  003724R  002   PTEM15= 000114          RHP41   007612R  002   SDMAX = 000003
ISDWC   015205R  002   PASCIN= 000006          PTEM2 = 000062          RHP42   007722R  002   SECT    000010R  002
ISTAT = 001610R  002   PAT   = 000020          PTEM3 = 000064          RHP6    007202R  002   SEEK    004756R  002
IVINT = 000250          PATCH   002026R  002   PTEM4 = 000066          RHP61   007210R  002   SELDRI  005024R  002
IVKIRQ  000254R  002   PBSW  = 000040          PTEM5 = 000070          RHP611  007356R  002   SEL1    005104R  002
JSETER  010154R  002   PC    =%000007          PTEM6 = 000072          RHP612  007244R  002   SETDED= 000040
KILL    002770R  002   PCUROV= 000035          PTEM7 = 000074          RHP613  007330R  002   SETVEC  000066R  002
KILLEX  003020R  002   PDNUMS= 000036          PTEM8 = 000076          RHP614  007374R  002   SIZE    000020R  002
KPAR4 = 172350          PDPNTR= 000034          PTEM9 = 000100          RHP62   007400R  002   SKCNT   001732R  002
KPDR4 = 172310          PDRCON= 077406          PTEND = 000242          RINTEX  011156R  002   SKCOM   004766R  002
LOCZ    000000R  002   POST  = 000122          PTLGTH= 000242          RINTV   011134R  002   SKI   = 000002
LUPCNT  002024R  002   PFBBOV= 000002          PTOCNT= 000030          RIOTS   015245R  002   SP    =%000006
MOS   = 001000          PFLGWD= 000000          PTSIZE= 000240          RKERBT= 067357          SPAR  = 020000
MISCNT  001734R  002   PFWADR= 000004          PUSRPC= 000236          RKNU  = 000022          SPIN    003716R  002
MMVER = 000001          PGE   = 002000          PUTBYT  000100R  002   RLS   = 000010          SPINFL= 000400
MSFMT1  001570R  002   PLNGTH= 000026          PWRIOA= 000020          RPAS  = 000016          SPOPER= 000200
MSFMT2  001571R  002   PMDLCD= 000032          PWRIOV= 000254          RPBA  = 000004          SRESET  003624R  002
MSFMT3  001576R  002   PNAME = 000010          PWRIOX= 000250          RPCS1 = 000000          SRES1   003654R  002
MSFMT5  001605R  002   PNTR  = 000116          P4CONS= 100000          RPCS1V  002014R  002   SRES2   003700R  002
MSGA    000014R  002   PNUMSG  013262R  002   RCCODE= 000013          RPCS2 = 000010          SRTO    014441R  002
MSGB    000016R  002   PNUM    015404R  002   RCODE = 000021          RPCS2V  002016R  002   SSCODE= 000011
MYATA   004404R  002   PNUMX = 015430R  002   RDCNT   001724R  002   RPDA  = 000006          STEIV   003130R  002
NCC   = 000760          PNUM1   015412R  002   RDCOM   004420R  002   RPDB  = 000024          STEPDN  003174R  002
NDC   = 000004          PNUM2   015420R  002   RDHD    004702R  002   RPDC  = 000020          STEPUP  003022R  002
NED   = 010000          POBJST= 000024          RDY   = 000200          RPDS  = 000012          STE1    003112R  002
NEE   = 000002          POPSW = 000002          READ    004410R  002   RPEC1 = 000030          STE2    003126R  002
NEM   = 004000          PRDIOA= 000016          RECAL   005012R  002   RPEC2 = 000032          STMNMG  014206R  002
NOATA   014547R  002   PRDIOV= 000244          REGNUM= 000020          RPER  = 000014          STMNUM  014216R  002
NOCOMP= 000001          PRDIOX= 000240          REL     003732R  002   RPER1 = 000014          STONER= 100000
NODI    014567R  002   PRINT   012764R  002   REL1    003756R  002   RPMR1 = 000026          STPCEX  003172R  002
NOICOM  004004R  002   PRINTX  013100R  002   REL2    004002R  002   RPMRIV  002020R  002   STPCOM  003132R  002
NOITER  014511R  002   PRIX1   013130R  002   RELOMG  013315R  002   RPMR2 = 000034          STPWRT  003064R  002
NOP1    004014R  002   PRIX2   013132R  002   REPORT  002160R  002   RPMR3 = 000036          STP1    003306R  002
NONE    015454R  002   PROCEX  011122R  002   REPTBL  002446R  002   RPTBAS  002402R  002   STP2    003204R  002
NONEX = 015463R  002   PROCTM  011006R  002   RESREG  012306R  002   RPTEND  002426R  002   STP3    003246R  002
NOWAIT  003444R  002   PROGNM  013264R  002   RETRYS  001760R  002   RPTLP   002364R  002   STP31   003300R  002
NPBYTE= 001000          PRONER= 020000          RHCODE= 000025          RPWC  = 000002          STSLUP= 002412
NRC   = 000010          PRTEX   013076R  002   RHPINT  006730R  002   RTNINT  000074R  002   STSTAT  012344R  002
NRCYL = 000633          PRTIWD  012714R  002   RHP1    007024R  002   RTO     014457R  002   SUIORG  005472R  002
NPHEAD= 000003          PS    = 177776          RHP11   007006R  002   RTRY    000012R  002   SUPTAD  012324R  002
NRWORD= 000400          PSRC  = 000120          RHP2    007050R  002   RTRYIP  002010R  002   SVAL  = 100000
NSVAL   015000R  002   PSRCST= 000022          RHP21   007172R  002   RTYEXH  014625R  002   SWOIER= 000020
NXF   = 000004          PSTKCT= 000124          RHP211  007164R  002   R0    =%000000          SWOVER= 000010
OCODE = 000015          PSTKSV= 000126          RHP23   007122R  002   R1    =%000001          SWOVTO= 000040
```

DTR6AA.P11 SYMBOL TABLE

```
TIAAP   015463R     002   UNIMAP= 000040           UP3     003354R     002   WAITMD= 100000           WRCOM   004452R     002
TOUTER  002522R     002   UNITMG  013334R     002 UP4     003314R     002   WCC.JE= 000031           WRHD    004714R     002
TSTVEC  000072R     002   UNKERR  014750R     002 URSTOP= 000002           WCE   = 040000           WRITE   004442R     002
TVECT   011160R     002   UNLOAD  003710R     002 USEUBM= 000200           WCECNT  001756R     002 WRL   = 004000
TVECTX  011206R     002   UNS   = 040000           US4TPS= 000002           WCODE = 000023           WR1     004562R     002
UCODE = 000007           UPE   = 020000           UXPATA  014606R     002 WHCODE= 000027           WR2     004650R     002
UFE   = 000400           UPSUB   003310R     002 VV    = 000100           WLE   = 004000           WR3     004534R     002
ULIST   000052R     002   UP1     003326R     002 VVFLG = 001000           WRCK    004726R     002 WT4IOT= 000010
UNASCI  013360R     002   UP2     003344R     002 WAIT    003424R     002   WRCNT   001726R     002 XXXX  = 000000
.     = 015544R     002

. ABS.  000000      000
        000000      001
RJP11   015544      002


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 *,DTR6AA/NL:TOC/DOC=DTR6AA.P11
 RUN-TIME: 8 18 1 SECONDS
 RUN-TIME RATIO: 57/28=1.9
 CORE USED:  6K  (11 PAGES)

 DOCUMENT PAGES:  64
```

NOS

Spooler cycling 10 Seconds, [illegible] [illegible] disk reads, 2 disk writes, 64 pages

```
[illegible] 11111111111111111111111111111110
[illegible] 11111111122222222222233312
[illegible] 345678901234567890 12 *
[illegible] 11111111111111111111111111111110
[illegible] 1111111111222222222223333333333444444444455555555556666666666777777777788888888889999999999000000000011111111112222222222233312
```