RK06L

CARTRIDGE FORMATTER

MD-11-DZR6L-B

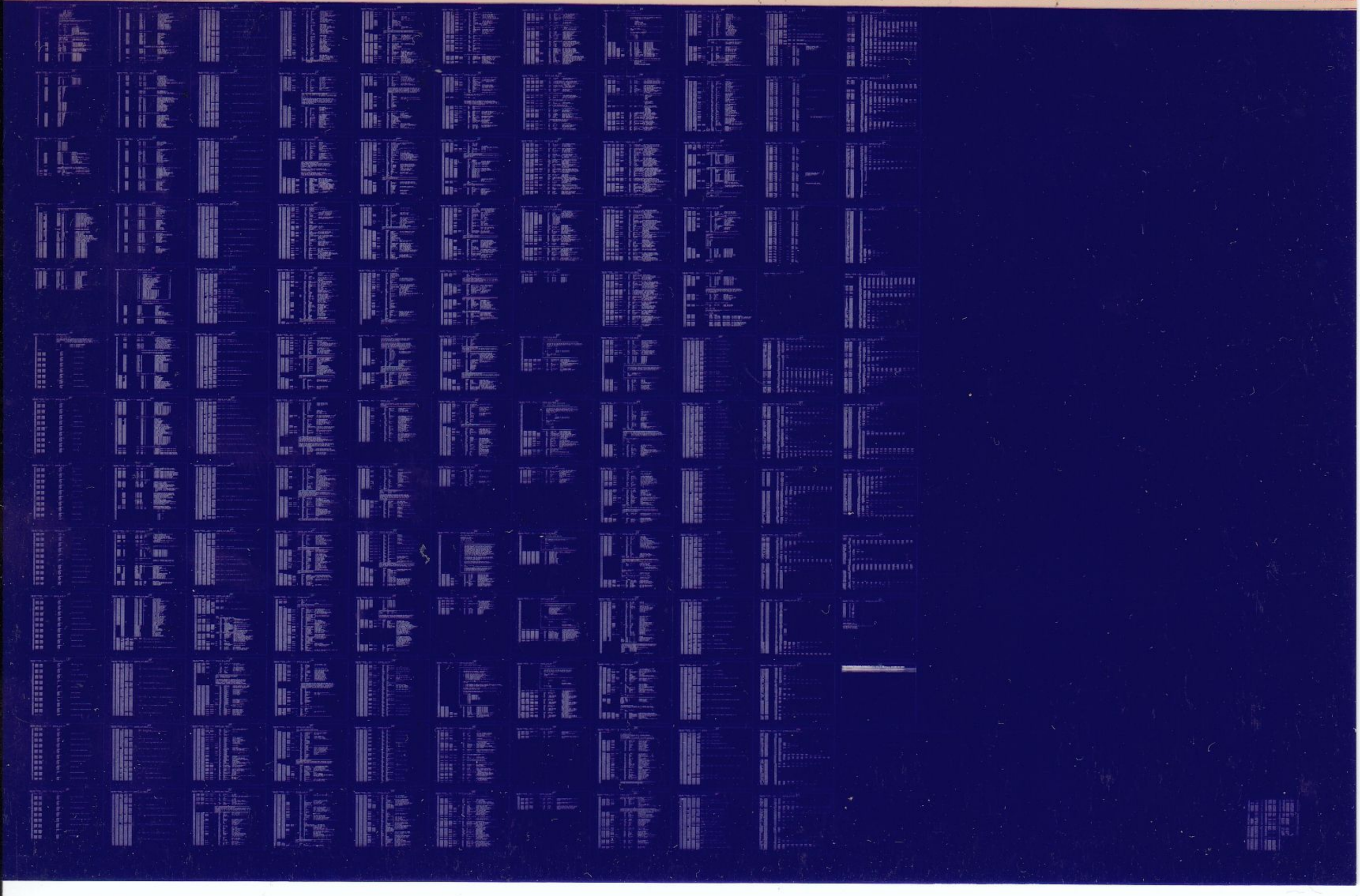
EP-DZR6L-B-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA



```
                                .NLIST  MC,MD,CND
                                .LIST   ME
                                .ENABL  ABS,AMA
163000                          $SWR=   163000
                        .TITLE  RK06 PACK FORMATTER
                        ;*COPYRIGHT (C) 1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY MARV TEGROTENHUIS
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
                        ;*
000001                  $TN=1
                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        10            BELL ON ERROR
                        ;*         9            LOOP ON ERROR
                        ;*         7            PRINT LIST OF BAD SECTORS
                        ;*         1            LIMIT DATA COMPARE ERROR REPORTS TO 10
                        ;*         0            REPEAT ENTIRE FORMAT OPERATION
                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                  STACK=  1100
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL


                        ;*MISCELLANEOUS DEFINITIONS
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE FOR LINE FEED
000015                  CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                  PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER


                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENEPAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
000006                  SP=     %6              ;;STACK POINTER
```

```
                        PC=     %7              ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
001000                  BIT09=  1000
000400                  BIT08=  400
000200                  BIT07=  200
000100                  BIT06=  100
000040                  BIT05=  40
000020                  BIT04=  20
000010                  BIT03=  10
000004                  BIT02=  4
000002                  BIT01=  2
```

```
                    000001                  BIT00=  1
                                            .EQUIV  BIT09,BIT9
                                            .EQUIV  BIT08,BIT8
                                            .EQUIV  BIT07,BIT7
                                            .EQUIV  BIT06,BIT6
                                            .EQUIV  BIT05,BIT5
                                            .EQUIV  BIT04,BIT4
                                            .EQUIV  BIT03,BIT3
                                            .EQUIV  BIT02,BIT2
                                            .EQUIV  BIT01,BIT1
                                            .EQUIV  BIT00,BIT0

                                            ;*BASIC "CPU" TRAP VECTOR ADDRESSES
                    000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
                    000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
                    000014                  TBITVEC=14              ;;"T" BIT
                    000014                  TRTVEC= 14              ;;TRACE TRAP
                    000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
                    000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
                    000024                  PWRVEC= 24              ;;POWER FAIL
                    000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
                    000034                  TRAPVEC=34              ;;"TRAP" TRAP
                    000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
                    000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
                    000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
                                            .SBTTL  TRAP CATCHER

                    000000                          .=0
                                            ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                            ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                            ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
                    000174                          .=174
000174  000000                             DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000                             SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                            .SBTTL  STARTING ADDRESS(ES)
000200  000137  016100                              JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
                    000204                          .=204
000204  000137  017776                              JMP     @#DRINIT
```

```
                                .SBTTL  COMMON TAGS

                                ;;**********************************************************************
                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                ;*USED IN THE PROGRAM.

                        001100          .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000          $PASS:  .WORD   0               ;;CONTAINS PASS COUNT
001102  000             $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103  000             $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114  000             $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115  001             $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134  000             $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135  000             $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154  000             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155  002             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156  012             $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157  000             $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001162  000000          $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001164  000000          $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001166  000000          $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001170  000000          $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001172  000000          $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001174  000000          $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001176  000000          $REG6:  .WORD   0               ;;CONTAINS (($REGAD)+14)
001200  000000          $REG7:  .WORD   0               ;;CONTAINS (($REGAD)+16)
001202  000000          $REG10: .WORD   0               ;;CONTAINS (($REGAD)+20)
001204  000000          $REG11: .WORD   0               ;;CONTAINS (($REGAD)+22)
001206  000000          $REG12: .WORD   0               ;;CONTAINS (($REGAD)+24)
001210  000000          $REG13: .WORD   0               ;;CONTAINS (($REGAD)+26)
001212  000000          $REG14: .WORD   0               ;;CONTAINS (($REGAD)+30)
001214  000000          $REG15: .WORD   0               ;;CONTAINS (($REGAD)+32)
001216  000000          $REG16: .WORD   0               ;;CONTAINS (($REGAD)+34)
001220  000000          $REG17: .WORD   0               ;;CONTAINS (($REGAD)+36)
001222  000000          $REG20: .WORD   0               ;;CONTAINS (($REGAD)+40)
```

F01


```
  207  001224  000000                  $REG21: .WORD   0               ;;CONTAINS (($REGAD)+42)
  208  001226  000000                  $REG22: .WORD   0               ;;CONTAINS (($REGAD)+44)
  209  001230  000000                  $REG23: .WORD   0               ;;CONTAINS (($REGAD)+46)
  210  001232  000000                  $REG24: .WORD   0               ;;CONTAINS (($REGAD)+50)
  211  001234  000000                  $REG25: .WORD   0               ;;CONTAINS (($REGAD)+52)
  212  001236  000000                  $TMP0:  .WORD   0               ;;USER DEFINED
  213  001240  000000                  $TMP1:  .WORD   0               ;;USER DEFINED
  214  001242  000000                  $TMP2:  .WORD   0               ;;USER DEFINED
  215  001244  000000                  $TMP3:  .WORD   0               ;;USER DEFINED
  216  001246  000000                  $TMP4:  .WORD   0               ;;USER DEFINED
  217  001250  000000                  $TMP5:  .WORD   0               ;;USER DEFINED
  218  001252  000000                  $TMP6:  .WORD   0               ;;USER DEFINED
  219  001254  000000                  $TMP7:  .WORD   0               ;;USER DEFINED
  220  001256  000000                  $TMP10: .WORD   0               ;;USER DEFINED
  221  001260  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
  222  001262  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
  223  001266     077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
  224  001267     015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
  225  001270  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
  226                                  ;;**************************************************************
```

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT


001272                          $ERRTB:

                                        ;ERROR 1                ;UNIBUS PARITY ERROR
001272  042462                         EM1
001274  046203                         DH100
001276  047502                         DT100
001300  047600                         DF01

                                        ;ERROR 2                ;NON-EXISTANT MEMORY
001302  042512                         EM2
001304  046203                         DH100
001306  047502                         DT100
001310  047600                         DF01

                                        ;ERROR 3                ;NON-EXISTANT DRIVE
001312  042550                         EM3
001314  046203                         DH100
001316  047502                         DT100
001320  047600                         DF01

                                        ;ERROR 4                ;UNIT FIELD ERROR
001322  042605                         EM4
001324  046203                         DH100
001326  047502                         DT100
001330  047600                         DF01

                                        ;ERROR 5                ;SUBSYSTEM TIMEOUT
001332  042632                         EM5
001334  046203                         DH100
001336  047502                         DT100
001340  047624                         DF02

                                        ;ERROR 6
001342  042660                         EM6                     ;SERCON PARITY ERROR
001344  046203                         DH100
001346  047502                         DT100
001350  047654                         DF03

                                        ;ERROR 7
001352  042713                         EM7                     ;DRIVE DETECTED PARITY ERROR
001354  046203                         DH100
001356  047502                         DT100
```

```
  283  001360  047654                        DF03
  284
  285                                        ;ERROR 10
  286  001362  042753                        EM10                ;AC LOW
  287  001364  046203                        DH100
  288  001366  047502                        DT100
  289  001370  047624                        DF02
  290
  291                                        ;ERROR 11
  292  001372  042766                        EM11                ;SPEED LOSS
  293  001374  046203                        DH100
  294  001376  047502                        DT100
  295  001400  047624                        DF02
  296
  297                                        ;ERROR 12
  298  001402  043005                        EM12                ;ILLEGAL FUNCTION
  299  001404  046203                        DH100
  300  001406  047502                        DT100
  301  001410  047624                        DF02
  302
  303                                        ;ERROR 13
  304  001412  043040                        EM13                ;PROGRAMMING ERROR
  305  001414  046203                        DH100
  306  001416  047502                        DT100
  307  001420  047600                        DF01
  308
  309                                        ;ERROR 14           ;NON-EXISTANT FUNCTION
  310  001422  043066                        EM14
  311  001424  046203                        DH100
  312  001426  047502                        DT100
  313  001430  047624                        DF02
  314
  315                                        ;ERROR 15
  316  001432  043126                        EM15                ;DRIVE TYPE ERROR
  317  001434  046203                        DH100
  318  001436  047502                        DT100
  319  001440  047624                        DF02
  320
  321                                        ;ERROR 16
  322  001442  043153                        EM16                ;FORMAT ERROR
  323  001444  046203                        DH100
  324  001446  047502                        DT100
  325  001450  047624                        DF02
  326
  327                                        ;ERROR 17
  328  001452  043174                        EM17                ;WRITE LOCK ERROR
  329  001454  046203                        DH100
  330  001456  047502                        DT100
  331  001460  047624                        DF02
  332
  333                                        ;ERROR 20
  334  001462  043221                        EM20                ;DRIVE UNSAFE
  335  001464  046203                        DH100
  336  001466  047502                        DT100
  337  001470  047624                        DF02
  338
```

```
339                                     ;ERROR 21
340  001472  043250                     EM21                ;SEEK INCOMPLETE
341  001474  046203                     DH100
342  001476  047502                     DT100
343  001500  047624                     DF02
344
345                                     ;ERROR 22
346  001502  043302                     EM22                ;CYLINDER OVERFLOW
347  001504  046203                     DH100
348  001506  047502                     DT100
349  001510  047624                     DF02
350
351                                     ;ERROR 23
352  001512  043336                     EM23                ;ILLEGAL CYLINDER
353  001514  046203                     DH100
354  001516  047502                     DT100
355  001520  047624                     DF02
356
357                                     ;ERROR 24
358  001522  043401                     EM24                ;DRIVE OFF TRACK
359  001524  046203                     DH100
360  001526  047502                     DT100
361  001530  047624                     DF02
362
363                                     ;ERROR 25
364  001532  043425                     EM25                ;DRIVE TIMING ERROR
365  001534  046203                     DH100
366  001536  047502                     DT100
367  001540  047624                     DF02
368
369                                     ;ERROR 26
370  001542  043454                     EM26                ;DATA LATE
371  001544  046203                     DH100
372  001546  047502                     DT100
373  001550  047624                     DF02
374
375                                     ;ERROR 27
376  001552  043500                     EM27                ;CONTROLLER TIMEOUT
377  001554  046203                     DH100
378  001556  047502                     DT100
379  001560  047624                     DF02
380
381                                     ;ERROR 30           ;OPERATION INCOMPLETE
382  001562  043535                     EM30
383  001564  046203                     DH100
384  001566  047502                     DT100
385  001570  047710                     DF05
386
387                                     ;ERROR 31
388  001572  043574                     EM31                ;HEADER CRC ERROR
389  001574  046203                     DH100
390  001576  047502                     DT100
391  001600  047740                     DF06
392
393                                     ;ERROR 32
394  001602  043621                     EM32                ;DATA CHECK ERROR
```

```
  395  001604  046203                    DH10C
  396  001606  047502                    DT10C
  397  001610  047770                    DF07
  398
  399                                    ;ERROR 33
  400  001612  043646                    EM33                ;WRITE CHECK ERROR
  401  001614  046203                    DH10C
  402  001616  047502                    DT10C
  403  001620  050020                    DF10
  404
  405                                    ;ERROR 34
  406  001622  043674                    EM34                ;DATA MISCOMPARE
  407  001624  046777                    DH604
  408  001626  047564                    DT601
  409  001630  050040                    DF11
  410
  411                                    ;ERROR 35
  412  001632  043720                    EM35                ;NO DRIVE RESPONSE-UFE & NED
  413  001634  046203                    DH10C
  414  001636  047502                    DT10C
  415  001640  047600                    DF01
  416
  417                                    ;ERROR 36
  418  001642  043760                    EM36                ;DRIVE ERROR WILL NOT CLEAR
  419  001644  000000                    0
  420  001646  000000                    0
  421  001650  000000                    0
  422
  423                                    ;ERROR 37           ;DRIVE STATUS CHANGE WILL NOT CLEAR
  424  001652  044017                    EM37
  425  001654  000000                    0
  426  001656  000000                    0
  427  001660  000000                    0
  428
  429                                    ;ERROR 40
  430  001662  044066                    EM40                ;ATTENTION BUT NO STATUS CHANGE OR FAULT
  431  001664  046203                    DH10C
  432  001666  047502                    DT10C
  433  001670  047624                    DF02
  434
  435                                    ;ERROR 41
  436  001672  044142                    EM41                ;ATTENTION BUT DRIVE NOT AVAILABLE
  437  001674  046203                    DH10C
  438  001676  047502                    DT10C
  439  001700  047624                    DF02
  440
  441                                    ;ERROR 42
  442  001702  044210                    EM42                ;ATTENTION WHEN NOT EXPECTED
  443  001704  046203                    DH10C
  444  001706  047502                    DT10C
  445  001710  047624                    DF02
  446
  447                                    ;ERROR 43           ;ERROR WHILE GATHERING DRIVE STATUS
  448  001712  044250                    EM43
  449  001714  046203                    DH10C
  450  001716  047502                    DT10C
```

```
  451  001720  050054                        DF12
  452
  453                                       ;ERROR 44
  454  001722  044576                        EM63            ;CLEAR CONTROLLER DID NOT CLEAR ERROR
  455  001724  046203                        DH100
  456  001726  047502                        DT100
  457  001730  050054                        DF12
  458
  459                                       ;ERROR 45
  460  001732  044647                        EM64            ;NO ATTENTION IN ATTENTION SUMMARY REG
  461  001734  046203                        DH100
  462  001736  047502                        DT100
  463  001740  050054                        DF12
  464
  465                                       ;ERROR 46
  466  001742  044726                        EM65            ;UNSOLICITED ATTENTION
  467  001744  046203                        DH100
  468  001746  047502                        DT100
  469  001750  050054                        DF12
  470
  471                                       ;ERROR 47
  472  001752  044760                        EM66            ;UNEXPECTED DATA TYPE ERROR
  473  001754  046203                        DH100
  474  001756  047502                        DT100
  475  001760  050054                        DF12
  476
  477                                       ;ERROR 50
  478  001762  045017                        EM67            ;ATTENTION DID NOT RESET WITH CLEAR
  479  001764  046203                        DH100
  480  001766  047502                        DT100
  481  001770  050054                        DF12
  482
  483                                       ;ERROR 51
  484  001772  045066                        EM70            ;SUBSYSTEM CLEAR DID NOT CLEAR DRIVE ATTENTION
  485  001774  046203                        DH100
  486  001776  047502                        DT100
  487  002000  050054                        DF12
  488
  489                                       ;ERROR 52
  490  002002  044317                        EM52            ;MULTIPLE DRIVE SELECT
  491  002004  046203                        DH100
  492  002006  047502                        DT100
  493  002010  050054                        DF12
  494
  495                                       ;ERROR 53
  496  002012  044351                        EM53            ;ABREVIATED HCE ERROR
  497  002014  046203                        DH100
  498  002016  047502                        DT100
  499  002020  050100                        DF13
  500
  501                                       ;ERROR 54
  502  002022  043535                        EM30            ;OPERATION INCOMPLETE ERROR
  503  002024  046203                        DH100
  504  002026  047502                        DT100
  505  002030  050124                        DF14
  506
```

```
                                        ;ERROR 55
002032  043574                          EM31            ;ABREVIATED HCRC ERROR
002034  046203                          DH100
002036  047502                          DT100
002040  050100                          DF13

                                        ;ERROR 56
002042  044402                          EM56            ;2 TIMEOUT ERROR
002044  046203                          DH100
002046  047502                          DT100
002050  050150                          DF15

                                        ;ERROR 57       ;2ND LEVEL IN SUBSYSTEM TIMEOUT
002052  044402                          EM56
002054  046203                          DH100
002056  047502                          DT100
002060  050210                          DF16

                                        ;ERROR 60
002062  044430                          EM60            ;ERROR IN RECAL FOR RECOVERY
002064  000000                          0
002066  000000                          0
002070  000000                          0

                                        ;ERROR 61
002072  044476                          EM61            ;ABORT MESSAGE
002074  000000                          0
002076  000000                          0
002100  000000                          0

                                        ;ERROR 62
002102  044550                          EM62            ;HEADER MISCOMPARE
002104  046662                          DH601
002106  047564                          DT601
002110  050250                          DF17

                                        ;ERROR 63       ;DATA ERROR WORDS
002112  000000                          0
002114  000000                          0
002116  047572                          DT602
002120  050310                          DF25

                                        ;ERROR 64
002122  044576                          EM63            ;CLEAR CONTROLLER DID NOT CLEAR ERROR
002124  046203                          DH100
002126  047502                          DT100
002130  047624                          DF02

                                        ;ERROR 65
002132  044647                          EM64            ;NO ATTENTION IN ATTENTION SUMMARY REG
002134  046203                          DH100
002136  047502                          DT100
002140  047624                          DF02

                                        ;ERROR 66
002142  044726                          EM65            ;UNSOLICITED ATTENTION
```

```
563  002144  046203                    DH100
564  002146  047502                    DT100
565  002150  047624                    DF02
566
567                                     ;ERROR 67
568  002152  044760                    EM66            ;UNEXPECTED DATA TYPE ERROR
569  002154  046203                    DH100
570  002156  047502                    DT100
571  002160  047624                    DF02
572
573                                     ;ERROR 70
574  002162  045017                    EM67            ;ATTENTION DID NOT RESET WITH CLEAR
575  002164  046203                    DH100
576  002166  047502                    DT100
577  002170  047624                    DF02
578
579                                     ;ERROR 71
580  002172  045066                    EM70            ;SUBSYSTEM CLEAR DID NOT CLEAR ATT
581  002174  046203                    DH100
582  002176  047502                    DT100
583  002200  047624                    DF02
584
585                                     ;ERROR 72
586  002202  045150                    EM71            ;DATA LATE WHEN UNLOADING HEADER
587  002204  046203                    DH100
588  002206  047502                    DT100
589  002210  047624                    DF02
590
591                                     ;ERROR 73
592  002212  045214                    EM72            ;CONTROLLER ERROR DURING DRIVER SERVICE
593  002214  046203                    DH100
594  002216  047502                    DT100
595  002220  047624                    DF02
596
597                                     ;ERROR 74
598  002222  045267                    EM73            ;DRIVE DETECTED PARITY ERROR
599  002224  046203                    DH100
600  002226  047502                    DT100
601  002230  047624                    DF02
602
603                                     ;ERROR 75
604  002232  045327                    EM74            ;UNDEFINED ERROR
605  002234  046203                    DH100
606  002236  047502                    DT100
607  002240  047624                    DF02
608
609                                     ;ERROR 76
610  002242  045353                    EM75            ;MARKING SECTOR BAD MESSAGE
611  002244  000000                    0
612  002246  000000                    0
613  002250  000000                    0
614
615                                     ;ERROR 77
616  002252  045407                    EM76            ;BAD DATA VERIFICATION WITH READ
617  002254  047076                    DH605
618  002256  047564                    DT601
```

```
  619  002260  050270                        DF21
  620
  621                                        ;ERROR 100
  622  002262  045502                        EM77                ;RETRY SUCCESFUL MESSAGE
  623  002264  047431                        DH800
  624  002266  047564                        DT601
  625  002270  050300                        DF23
  626
  627                                        ;ERROR 101
  628  002272  045502                        EM77                ;ANOTHER RETRY SUCCESSFUL MESSAGE
  629  002274  047431                        DH800
  630  002276  047564                        DT601
  631  002300  050300                        DF23
  632
  633                                        ;ERROR 102
  634  002302  045527                        EM100               ;RETRY UNSUCCESSFUL MESSAGE
  635  002304  047431                        DH800
  636  002306  047564                        DT601
  637  002310  050300                        DF23
  638
  639                                        ;ERROR 103
  640  002312  045556                        EM101               ;NO VALID HEADERS IN TRACK JUST READ
  641  002314  047060                        D:6042
  642  002316  047564                        DT601
  643  002320  050304                        DF24
  644
  645                                        ;ERROR 104
  646  002322  045640                        EM102               ;BSE ERROR ON SECTOR NOT LISTED AS BAD
  647  002324  046203                        DH100
  648  002326  047502                        DT100
  649  002330  047624                        DF02
  650
  651                                        ;ERROR 105
  652  002332  045716                        EM103                       ;WORD COUNT NOT CORRECT TO CONTINUE
  653  002334  047454                        DH900
  654  002336  047572                        DT602
  655  002340  050310                        DF25
  656
  657                                        ;ERROR 106
  658  002342  045762                        EM104               ;DRIVE STATUS NOT VALID
  659  002344  000000                        0
  660  002346  000000                        0
  661  002350  000000                        0
  662
  663                                        ;ERROR 107
  664  002352  046047                        EM105                       ;BAD HEADER AREA
  665  002354  046777                        DH604
  666  002356  047564                        DT601
  667  002360  050314                        DF26
```

```
                                .SBTTL  RK06 CONTROLLER REGISTER DEFINITION
                        
000000                  RKCS1=  0               ;CONTROL AND STATUS REGISTER 1
000002                  RKWC=   2               ;WORD COUNT REGISTER
000004                  RKBA=   4               ;BUS ADDRESS REGISTER
000006                  RKDA=   6               ;DESIRED TRACK SECTOR REGISTER
000010                  RKCS2=  10              ;CONTROL AND STATUS REGISTER 2
000012                  RKDS=   12              ;DRIVE STATUS REGISTER
000014                  RKER=   14              ;ERROR REGISTER
000016                  RKASOF= 16              ;ATTENTION SUMMARY AND OFFSET REGISTER
000020                  RKDC=   20              ;DESIRED CYLINDER REGISTER
000020                  RKDCYL= 20              ;DESIRED CYLINDER REGISTER
000024                  RKDB=   24              ;DATA BUFFER
000022                  RKMR1=  22              ;MAINTENANCE REGISTER 1
000034                  RKMR2=  34              ;MAINTENANCE REGISTER 2
000036                  RKMR3=  36              ;MAINTENANCE REGISTER 3
000030                  RKPOS=  30              ;ECC POSITION INFORMATION
000030                  RKECPS= 30              ;ECC POSITION INFORMATION
000032                  RKPAT=  32              ;ECC PATTERN INFORMATION
000032                  RKECPT= 32              ;ECC PATTERN INFORMATION

                                .SBTTL  DRIVE COMMANDS

000101                  SELDRV= 101             ;SELECT DRIVE
000103                  PACK=   103             ;PACK ACKNOWLEDGE
000105                  CLEAR=  105             ;DRIVE CLEAR
000107                  UNLOAD= 107             ;UNLOAD
000111                  SRTSPL= 111             ;START SPINDLE
000113                  RECAL=  113             ;RECALIBRATE
000115                  OFFSET= 115             ;OFFSET
000117                  SEEK=   117             ;SEEK
000121                  RDDATA= 121             ;READ DATA
000123                  WRDATA= 123             ;WRITE DATA
000125                  RDHEAD= 125             ;READ HEADER
000127                  WRHEAD= 127             ;WRITE HEADER AND DATA
000131                  WRTCHK= 131             ;WRITE CHECK

                        ;       THE FOLLOWING ARE NOT DRIVE COMMANDS BUT ARE USED BY THE DRIVER
                        ;       TO SIMULATE A SPECIFIC DESIRED OPERATION

000140                  RELEAS= 140             ;RELEASE DRIVE
000141                  RDSTAT= 141             ;GET ALL STATUS FROM DRIVE
000164                  RDALHD= 164             ;READ ALL HEADERS
000176                  CONCLR= 176             ;CONTROLLER CLEAR (BIT 15 OF CS1)
000177                  SUBCLR= 177             ;SUBSYSTEM CLEAR (BIT 5 OF CS2)
000300                  INTR=   300             ;GENERATE INTERRUPT TO CPU

                        ;       DRIVER ISSUED SERVICE COMMANDS

000001                  DR.SEL= 001             ;DRIVE SELECT
000005                  DR.CLR= 005             ;DRIVE CLEAR

                                .SBTTL  CONTROL AND STATUS REGISTER 1 BITS

000001                  GO=     BIT0            ;GO BIT
```

```
000100                  IE=     BIT6                    ;INTERRUPT ENABLE
000200                  RDY=    BIT7                    ;CONTROLLER READY
000400                  BA16=   BIT8                    ;BUS ADDRESS BIT 16
001000                  BA17=   BIT9                    ;BUS ADDRESS BIT 17
002000                  CDT=    BIT10                   ;CONTROLLER DRIVE TYPE (0=RK06)
004000                  CTO=    BIT11                   ;CONTROLLER TIMED OUT WAITING FOR
                                                        ;  DRIVE RESPONSE
010000                  CFMT=   BIT12                   ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
020000                  SPAR=   BIT13                   ;DRIVE BUS PARITY ERROR DETECTED BY CONTROLLER
040000                  DI=     BIT14                   ;DRIVE INTERRUPT
100000                  CERR=   BIT15                   ;CONTROLLER ERROR
100000                  CCLR=   BIT15                   ;CONTROLLER CLEAR

                        ;       THESE BIT DEFINITIONS ARE USED FOR ADDRESS
                        ;       THE HIGH BYTE OF RKCS1

000001                  B.BA16= BIT0                    ;BUS ADDRESS BIT 16
000002                  B.BA17= BIT1                    ;BUS ADDRESS BIT 17
000004                  B.CDT=  BIT2                    ;CONTROLLER DRIVE TYPE (0=RK06)
000020                  B.CFMT= BIT4                    ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)

                        .SBTTL  CONTROL AND STATUS REGISTER 2 BITS

000007                  DRVMSK= 7                       ;MASK FOR DRIVE SELECTION CODE
000010                  DESL=   BIT3                    ;DESELECT OR RELEASE DRIVE IN BITS 0-2
000010                  RLS=    BIT3                    ;DESELECT OR RELEASE DRIVE IN BITS 0-2
000020                  BAI=    BIT4                    ;BUS ADDRESS INCREMENT INHIBIT
000040                  CLR=    BIT5                    ;CLEAR CONTROLLER AND ALL DRIVES
000040                  SCLR=   BIT5                    ;CLEAR CONTROLLER AND ALL DRIVES
000100                  IR=     BIT6                    ;INPUT READY
000200                  OR=     BIT7                    ;OUTPUT READY
000400                  UFE=    BIT8                    ;UNIT FIELD ERROR
001000                  MDS=    BIT9                    ;MULTIPLE DRIVE SELECT
002000                  PGE=    BIT10                   ;PROGRAMMING ERROR
004000                  NEM=    BIT11                   ;NON-EXISTENT MEMORY
010000                  NED=    BIT12                   ;NON-EXISTENT DRIVE
020000                  UPE=    BIT13                   ;UNIBUS PARITY ERROR
040000                  WCE=    BIT14                   ;WRITE CHECK ERROR
100000                  DLT=    BIT15                   ;DATA LATE ERROR

                        .SBTTL  ERROR REGISTER BIT DEFINITION

000001                  ILC=    BIT0                    ;ILLEGAL FUNCTION CODE
                        .*ILF=  BIT0                    ;ILLEGAL FUNCTION CODE
000002                  SKI=    BIT1                    ;SEEK INCOMPLETE
000004                  ILF=    BIT2                    ;ILLEGAL DRIVE FUNCTION
000004                  NXF=    BIT2                    ;ILLEGAL DRIVE FUNCTION
000010                  DRPAR=  BIT3                    ;DRIVE DETECTED DRIVE BUS PARITY ERROR
000020                  FMTE=   BIT4                    ;FORMAT ERROR
000040                  DTYE=   BIT5                    ;DRIVE TYPE ERROR
000100                  ECH=    BIT6                    ;ECC HARD
000200                  BSE=    BIT7                    ;BAD SECTOR ERROR
000400                  HCRC=   BIT8                    ;HEADER CRC ERRROR
000400                  HVRC=   BIT8                    ;HEADER VRC ERRROR
001000                  COE=    BIT9                    ;CYLINDER ADDRESS OVERFLOW ERROR
002000                  IDAE=   BIT10                   ;INVALID DISK ADDRESS ERROR
```

```
                        WLE=    BIT11                   ;WRITE LOCK ERROR
                        DTE=    BIT12                   ;DRIVE TIMING ERROR
                        OPI=    BIT13                   ;OPERATION (SEARCH) INCOMPLETE
                        UNS=    BIT14                   ;DRIVE UNSAFE
                        DCK=    BIT15                   ;DATA CHECK

                        .SBTTL  STATUS REGISTER BIT DEFINITION

                        DRA=    BIT0                    ;DRIVE AVAILABLE (CONTROLLER IS SET IF
                                                        ;  THIS BIT IS RESET)
                        OFST=   BIT2                    ;DRIVE OFFSET
                        ACLO=   BIT3                    ;AC LOW
                        SPDLSS= BIT4                    ;SPEED LOSS
                        DCLO=   BIT4                    ;DC LOW
                        DROT=   BIT5                    ;DRIVE OFF TRACK
                        VV=     BIT6                    ;VOLUME VALID
                        DRY=    BIT7                    ;DRIVE READY
                        DRDY=   BIT7                    ;DRIVE READY
                        DDT=    BIT8                    ;DRIVE TYPE (0=RK06)
                        WRL=    BIT11                   ;WRITE LOCK
                        PIP=    BIT13                   ;POSITIONING IN PROGRESS
                        DSC=    BIT14                   ;DRIVE STATUS CHANGE
                        SVAL=   BIT15                   ;STATUS VALID

                        .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION

                        MESMSK= 17                      ;MESSAGE MASK

                        PAT=    BIT4                    ;FORCE EVEN PARITY ON DRIVE BUS MESSAGE LINES
                        DMD=    BIT5                    ;DIAGNOSTIC MODE
                        MSP=    BIT6                    ;MAINTENANCE SECTOR PULSE
                        MIND=   BIT7                    ;MAINTENANCE INDEX
                        MCLK=   BIT8                    ;MAINTENANCE CLOCK
                        MERD=   BIT9                    ;MAINTENANCE ENCODED READ DATA
                        MEWD=   BIT10                   ;MAINTENACNE ENCODED WRITE DATA
                        PCA=    BIT11                   ;PRECOMPENSATION ADVANCE
                        PCD=    BIT12                   ;PRECOMPENSATION DELAY
                        ECCW=   BIT13                   ;ECC WORD IS BEING READ OR WRITTEN
                        WRTGAT= BIT14                   ;WRITE GATE
                        RDGATE= BIT15                   ;READ GATE

                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A

                        S.DRA=  BIT5                    ;DRIVE AVAILIABLE
                        S.VV=   BIT6                    ;VOLUME VALID
                        S.DRY=  BIT7                    ;DRIVE READY
                        S.TYPE= BIT8                    ;DRIVE TYPE
                        S.FORM= BIT9                    ;DRIVE FORMAT
                        S.OFF=  BIT10                   ;OFFSET
                        S.WRL=  BIT11                   ;WRITE LOCK
                        S.SPIN= BIT12                   ;SPINDLE ON
                        S.PIP=  BIT13                   ;POSITIONING IN PROGRESS
                        S.DSC=  BIT14                   ;DRIVE STATUS CHANGE

                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B
```

```
                        DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B

000040                  S.ICYL= BIT5            ;ILLEGAL CYLINDER ADDRESS
000100                  S.ACLO= BIT6            ;AC LOW
000200                  S.FLT=  BIT7            ;DRIVE FAULT
000400                  S.ILF=  BIT8            ;ILLEGAL FUNCTION
001000                  S.PAR=  BIT9            ;DRIVE DETECTED DRIVE BUS PARITY ERROR
002000                  S.SKI=  BIT10           ;SEEK INCOMPLETE
004000                  S.WLE=  BIT11           ;WRITE LOCK ERROR
010000                  S.SPLS= BIT12           ;SPEED LOSS
010000                  S.DCLO= BIT12           ;DC LOW
020000                  S.DROT= BIT13           ;DRIVE OFF TRACK
040000                  S.UNS=  BIT14           ;DRIVE UNSAFE

                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A

000020                  S.XDOK= BIT4            ;TRANSDUCER OK
000040                  S.HDHM= BIT5            ;HEADS HOME
000100                  S.BRHM= BIT6            ;BRUSHES HOME
000200                  S.DOOR= BIT7            ;DOOR INTERLOCKED
000400                  S.CART= BIT8            ;CARTRAGE INTERLOCK
001000                  S.SPOK= BIT9            ;SPEED OK
002000                  S.FWD=  BIT10           ;FORWARD
004000                  S.REV=  BIT11           ;REVERSE
010000                  S.LOAD= BIT12           ;HEADS LOADING
020000                  S.RTZ=  BIT13           ;RETURN TO ZERO
040000                  S.UNLD= BIT14           ;HEADS UNLOADING

                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B

000020                  S.SECT= BIT4            ;SECTOR ERROR
000040                  S.WCLK= BIT5            ;WRITE CLOCK AND NO WRITE GATE
000100                  S.WGAT= BIT6            ;WRITE GATE AND NO TRANSISTIONS
000200                  S.HDFL= BIT7            ;HEAD FAULT
000400                  S.MHD=  BIT8            ;MULTIPLE HEAD SELECT
001000                  S.XERR= BIT9            ;INDEX ERROR
002000                  S.DIB=  BIT10           ;DIBIT ERROR
004000                  S.PLO=  BIT11           ;PLO ERROR
010000                  S.NMOV= BIT12           ;SEEK AND NO MOTION
020000                  S.LIMD= BIT13           ;LIMIT DETECT ON SEEK
040000                  S.BRKE= BIT14           ;SERVO-BRAKE

                        .SBTTL  COMMON MASKS

000007                  M.DRV=  7               ;DRIVE CODE
100000                  M.PAR=  BIT15           ;PARITY
000003                  M.ID=   3               ;BYTE ID
017760                  M.CDIF= 17760           ;CYLINDER DIFFERENCE/OFFSET
017760                  M.CADD= 17760           ;CYLINDER ADDRESS
077770                  M.SER=  77770           ;DRIVE SERIAL NUMBER
000760                  M.SECT= 760             ;SECTOR COUNT
007000                  M.HEAD= 7000            ;HEAD DECODE
```

```
                        .SBTTL  PARAMETER BLOCK ALLOCATION

                        ;*      ************************************************
                        ;*  1   ! COMMAND                 ! DRIVE NO.          !  0
                        ;*  3   ! CYLINDER ADDRESS                             !  2
                        ;*  5   ! TRACK                   ! SECTOR             !  4
                        ;*  7   !BA16-17,FORMAT,DRV TYPE! OFFSET               !  6
                        ;* 11   ! BUS ADDRESS (LOW 16 BITS)                    ! 10
                        ;* 13   ! WORD COUNT (2'S COMPLEMENT)                  ! 12
                        ;* 15   ! PROGRAM DRIVE STATUS INFORMATION             ! 14
                        ;* 17   ! COMMAND AND STATUS REGISTER 1                ! 16
                        ;* 21   ! COMMAND AND STATUS REGISTER 2                ! 20
                        ;* 23   ! WORD COUNT REGISTER                          ! 22
                        ;* 25   ! BUS ADDRESS REGISTER                         ! 24
                        ;* 27   ! DESIRED TRACK AND SECTOR                     ! 26
                        ;* 31   ! DESIRED CYLINDER                             ! 30
                        ;* 33   ! ATTENTION SUMMARY AND DRIVE OFFSET           ! 32
                        ;* 35   ! ERROR REGISTER                               ! 34
                        ;* 37   ! STATUS REGISTER                              ! 36
                        ;* 41   ! MESSAGE LINE A STATUS BYTE 00                ! 40
                        ;* 43   ! MESSAGE LINE B STATUS BYTE 00                ! 42
                        ;* 45   ! MESSAGE LINE A STATUS BYTE 01                ! 44
                        ;* 47   ! MESSAGE LINE B STATUS BYTE 01                ! 46
                        ;* 51   ! MESSAGE LINE A STATUS BYTE 10                ! 50
                        ;* 53   ! MESSAGE LINE B STATUS BYTE 10                ! 52
                        ;* 55   ! MESSAGE LINE A STATUS BYTE 11                ! 54
                        ;* 57   ! MESSAGE LINE B STATUS BYTE 11                ! 56
                        ;* 61   ! ECC POSITION INFORMATION                     ! 60
                        ;* 63   ! ECC PATTERN INFORMATION                      ! 62
                        ;*      ************************************************

                        .SBTTL  PARAMETERS PASSED TO THE DRIVER

                        ;       THE FOLLOWING DEFINITIONS ARE USED TO PASS PARAMETERS
                        ;         TO THE RK06 DRIVER

000000                  P.DRVN= 0                       ;DRIVE NUMBER
000001                  P.CMND= 1                       ;COMMAND
000002                  P.CYLN= 2                       ;CYLINDER ADDRESS
000004                  P.SECT= 4                       ;SECTOR
000005                  P.TRCK= 5                       ;TRACK
000006                  P.OFST= 6                       ;OFFSET
000007                  P.CS1H= 7                       ;RKCS1 BITS 8-15
000007                  P.BAHI= 7                       ;BUS ADDRESS (BITS 16 AND 17)
000010                  P.BALO= 10                      ;BUS ADDRESS (BITS 0-15)
000012                  P.WC=   12                      ;WORD COUNT (2'S COMPLEMENT)
000014                  P.PRST= 14                      ;PROGRAM DRIVE STATUS INFORMATION

                        .SBTTL  PROGRAM DEVICE STATUS REGISTER DEFINITION

000001                  DRVUSE= BIT0                    ;DRIVE IN USE
000002                  DRVPOS= BIT1                    ;DRIVE POSITIONING
000004                  DRVPDT= BIT2                    ;DRIVE POSITIONED FOR DATA TRANSFER
000010                  UEXATT= BIT3                    ;UNEXPECTED ATTENTION
000020                  DRVHRD= BIT4                    ;DRIVE HAS HARD ERROR
000040                  DRVDSC= BIT5                    ;DRIVE STATUS CHANGE DID NOT CLEAR
```

```
                000100                  CMDTO=  BIT6                    ;NO TERMINATION TO COMMAND FOR AT
                                                                        ;  LEAST 1 SECOND
                000200                  W.WCK=  BIT7                    ;WRITE FOR WRITE WRITE CHECK
                000400                  NOCHK=  BIT8                    ;NO CHECK, DO NOT SET INTERRUPT ENABLE
                001000                  PBSVAL= BIT9                    ;PARAMETER STATUS WORDS VALID
                                                                        ;  (SET WHEN ERROR TERMINATION OR
                                                                        ;  READ STATUS COMMAND)
                002000                  DRPDRV= BIT10                   ;DROP DRIVE FROM TEST SEQUENCE
                004000                  NOOSC=  BIT11                   ;ATTENTION SET BUT DCS AND FAULT RESET
                010000                  DRVSZD= BIT12                   ;DRIVE SEIZED BY OTHER PORT
                020000                  E.UNLD= BIT13                   ;DRIVE UNLOADED DUE TO ERROR
                040000                  Q.INIT= BIT14                   ;PARAMETER BLOCK ENQUEUED IN INITIATION QUEUE
                100000                  DTBAII= BIT15                   ;INHIBIT BUS ADDRESS INCREMENT

                                        .SBTTL  PARAMETERS PASSED FROM DRIVER TO PROGRAM

                                        ;       THE FOLLOWING DEFINITIONS ARE USED FOR REGISTER RETURNS
                                        ;         FROM THE DRIVER TO THE CALLING PROGRAM

                000016                  P.CS1=  16                      ;COMMAND AND STATUS REGISTER 1
                000020                  P.CS2=  20                      ;COMMAND AND STATUS REGISTER 2
                000022                  P.WCR=  22                      ;WORD COUNT REGISTER
                000024                  P.BAR=  24                      ;BUS ADDRESS REGISTER
                000026                  P.DTS=  26                      ;DESIRED TRACK SECTOR REGISTER
                000030                  P.DCYL= 30                      ;DESIRED CYLINDER REGISTER
                000032                  P.ASOF= 32                      ;ATTENTION SUMMARY/OFFSET REGISTER
                000034                  P.ER=   34                      ;ERROR REGISTER
                000036                  P.DS=   36                      ;STATUS REGISTER
                000040                  P.A00=  40                      ;MESSAGE A STATUS BYTE 00
                000042                  P.B00=  42                      ;MESSAGE B STATUS BYTE 00
                000044                  P.A01=  44                      ;MESSAGE A STATUS BYTE 01
                000046                  P.B01=  46                      ;MESSAGE B STATUS BYTE 01
                000050                  P.A10=  50                      ;MESSGGE A STATUS BYTE 10
                000052                  P.B10=  52                      ;MESSAGE B STATUS BYTE 10
                000054                  P.A11=  54                      ;MESSAGE A STATUS BYTE 11
                000056                  P.B11=  56                      ;MESSAGE B STATUS BYTE 11
                000060                  P.EPOS= 60                      ;ECC POSITION INFORMATION
                000062                  P.EPAT= 62                      ;ECC PATTERN INFORMATION

                                        .SBTTL  PARAMETER BLOCK 0 FOR DRIVE

002362      000         PARM0:  .BYTE   0                               ;DRIVE NUMBER
002363      000                 .BYTE   0                               ;COMMAND
002364   000000                 .WORD   0                               ;CYLINDER ADDRESS
002366      000                 .BYTE   0                               ;SECTOR ADDRESS
002367      000                 .BYTE   0                               ;TRACK ADDRESS
002370      000                 .BYTE   0                               ;OFFSET VALUE
002371      000                 .BYTE   0                               ;BUS ADDRESS (BITS 16 AND 17)
002372   000000                 .WORD   0                               ;BUS ADDRESS (BITS 0 - 15)
002374   000000                 .WORD   0                               ;WORD COUNT (2'S COMPLEMENT)
002376   000000                 .WORD   0                               ;PROGRAM DRIVE STATUS INFORMATION
002400   000000                 .WORD   0                               ;COMMAND AND STATUS REGISTER 1
002402   000000                 .WORD   0                               ;COMMAND AND STATUS REGISTER 2
002404   000000                 .WORD   0                               ;WORD COUNT REGISTER
002406   000000                 .WORD   0                               ;BUS ADDRESS REGISTER
002410   000000                 .WORD   0                               ;DESIRED TRACK AND SECTOR REGISTER
```

```
  998  002412  000000                    .WORD   0               ;DESIRED CYLINDER REGISTER
  999  002414  000000                    .WORD   0               ;ATTENTION SUMMARY/OFFSET REGISTER
 1000  002416  000000                    .WORD   0               ;ERROR REGISTER
 1001  002420  000000                    .WORD   0               ;STATUS REGISTER
 1002  002422  000000                    .WORD   0               ;MESSAGE LINE A STATUS BYTE 00
 1003  002424  000000                    .WORD   0               ;MESSAGE LINE B STATUS BYTE 00
 1004  002426  000000                    .WORD   0               ;MESSAGE LINE A STATUS BYTE 01
 1005  002430  000000                    .WORD   0               ;MESSAGE LINE B STATUS BYTE 01
 1006  002432  000000                    .WORD   0               ;MESSAGE LINE A STATUS BYTE 10
 1007  002434  000000                    .WORD   0               ;MESSAGE LINE B STATUS BYTE 10
 1008  002436  000000                    .WORD   0               ;MESSAGE LINE A STATUS BYTE 11
 1009  002440  000000                    .WORD   0               ;MESSAGE LINE B STATUS BYTE 11
 1010  002442  000000                    .WORD   0               ;ECC POSITION INFORMATION
 1011  002444  000000                    .WORD   0               ;ECC PATTERN INFORMATION
 1012
 1013                          .SBTTL  PARAMETER BLOCK 1 FOR DRIVE
 1014
 1015  002446      000         PARM1:  .BYTE   0               ;DRIVE NUMBER
 1016  002447      000                 .BYTE   0               ;COMMAND
 1017  002450  000000                  .WORD   0               ;CYLINDER ADDRESS
 1018  002452      000                 .BYTE   0               ;SECTOR ADDRESS
 1019  002453      000                 .BYTE   0               ;TRACK ADDRESS
 1020  002454      000                 .BYTE   0               ;OFFSET VALUE
 1021  002455      000                 .BYTE   0               ;BUS ADDRESS (BITS 16 AND 17)
 1022  002456  000000                  .WORD   0               ;BUS ADDRESS (BITS 0 - 15)
 1023  002460  000000                  .WORD   0               ;WORD COUNT (2'S COMPLEMENT)
 1024  002462  000000                  .WORD   0               ;PROGRAM DRIVE STATUS INFORMATION
 1025  002464  000000                  .WORD   0               ;COMMAND AND STATUS REGISTER 1
 1026  002466  000000                  .WORD   0               ;COMMAND AND STATUS REGISTER 2
 1027  002470  000000                  .WORD   0               ;WORD COUNT REGISTER
 1028  002472  000000                  .WORD   0               ;BUS ADDRESS REGISTER
 1029  002474  000000                  .WORD   0               ;DESIRED TRACK AND SECTOR REGISTER
 1030  002476  000000                  .WORD   0               ;DESIRED CYLINDER REGISTER
 1031  002500  000000                  .WORD   0               ;ATTENTION SUMMARY/OFFSET REGISTER
 1032  002502  000000                  .WORD   0               ;ERROR REGISTER
 1033  002504  000000                  .WORD   0               ;STATUS REGISTER
 1034  002506  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 00
 1035  002510  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 00
 1036  002512  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 01
 1037  002514  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 01
 1038  002516  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 10
 1039  002520  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 10
 1040  002522  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 11
 1041  002524  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 11
 1042  002526  000000                  .WORD   0               ;ECC POSITION INFORMATION
 1043  002530  000000                  .WORD   0               ;ECC PATTERN INFORMATION
 1044
 1045                          .SBTTL  TEMPORARY CONTROLLER REGISTER STORAGE
 1046
 1047  002532  000000         T.CS1:  .WORD   0               ;TEMPORARY STORAGE FOR COMMAND AND STATUS
 1048                                                         ;  REGISTER 1
 1049  002534  000000         T.CS2:  .WORD   0               ;TEMPORARY STORAGE FOR COMMAND AND STATUS
 1050                                                         ;  REGISTER 2
 1051  002536  000000         T.WCR:  .WORD   0               ;TEMPORARY STORAGE FOR WORD COUNT REGISTER
 1052  002540  000000         T.BA:   .WORD   0               ;TEMPORARY STORAGE FOR BUS ADDRESS REGISTER
 1053  002542  000000         T.DA:   .WORD   0               ;TEMPORARY STORAGE FOR DISK TRACK AND SECTOR
```

```
1054  002544  000000                  T.DC:   .WORD   0               ;TEMPORARY STORAGE FOR DRIVE CYLINDER
1055  002546  000000                  T.ASOF: .WORD   0               ;TEMPORARY STORAGE FOR ATTENTION SUMMARY
1056                                                                  ;  AND OFFSET
1057  002550  000000                  T.ER:   .WORD   0               ;TEMPORARY STORAGE FOR ERROR REGISTER
1058  002552  000000                  T.DS:   .WORD   0               ;TEMPORARY STORAGE FOR DRIVE STATUS REGISTER
1059  002554  000000                  T.MR1:  .WORD   0               ;TEMPORARY STORAGE FOR MAINTENANCE REGISTER 1
1060  002556  000000                  T.MR2:  .WORD   0               ;TEMPORARY STORAGE FOR MAINTENANCE REGISTER 2
1061  002560  000000                  T.MR3:  .WORD   0               ;TEMPORARY STORAGE FOR MAINTENANCE REGISTER 3
1062  002562  000000                  T.POS:  .WORD   0               ;TEMPORARY STORAGE FOR ECC POSITION
1063  002564  000000                  T.PAT:  .WORD   0               ;TEMPORARY STORAGE FOR ECC PATTERN
1064  002566  000000                  T.DB:   .WORD   0               ;TEMPORARY STORAGE FOR DATA BUFFER REGISTER
1065
1066                                  .SBTTL  DRIVER PARAMERTERS
1067
1068  002570  177440                  RKBAS:  .WORD   177440          ;ADDRESS OF RK611 UNIBUS ADDRESS BLOCK
1069  002572  000210                  RKVEC:  .WORD   210             ;ADDRESS OF R611 VECTOR
1070  002574  000240                  RKPRI:  .WORD   PR5             ;RK611 INTERRUPT PRIORITY
1071  002576  025302                  A.NORM: ERRFRE                  ;ADDRESS OF NORMAL RETURN FROM DRIVER
1072  002600  026240                  A.ABNL: ERRHDL                  ;ADDRESS OF ABNORMAL RETURN FROM DRIVER
1073  002602  025616                  A.CONT: CONERR                  ;ADDRESS OF CONTROLLER ERROR RETURN
1074  002604  000000                  E.CONT: .WORD   0               ;CONTROLLER ERROR STATUS
1075                                                                  ;  THIS LOCATION IS CLEARED WHEN EVERY COMMAND
1076                                                                  ;  IS INITIATED.  IF A CONTROLLER ERROR
1077                                                                  ;  OCCURS THE FOLLOWING BIT ASSIGNMENT IS
1078                                                                  ;  USED:
1079                                                                  ;
1080          000001                  E.CCLR= BIT0                    ;CLEAR CONTROLLER DID NOT CLEAR ERROR
1081          000002                  E.NOAT= BIT1                    ;NO ATTENTION IN ATTENTION SUMMARY REG
1082          000004                  E.UATT= BIT2                    ;UNSOLICATED ATTENTION (SEQUENTIAL ONLY)
1083          000010                  E.UDAT= BIT3                    ;UNEXPECTED DATA TYPE ERROR
1084          000020                  E.CLAT= BIT4                    ;ATTENTION DID NOT RESET WITH CLEAR
1085          000040                  E.SCLR= BIT5                    ;SUBSYSTEM CLEAR DID NOT CLEAR DRIVE
1086                                                                  ;  ATTENTION
1087          000100                  E.ILLD= BIT6                    ;ILLEGAL DRIVER COMMAND
1088          000400                  E.DLT=  BIT8                    ;DATA LATE WHEN UNLOADING HEADER
1089          001000                  E.CERR= BIT9                    ;CONTROLLER ERROR DURING DRIVER SERVICING
1090          002000                  E.DPAR= BIT10                   ;DRIVE DETECTED PARITY ERROR
1091          040000                  E.CMTO= BIT14                   ;CONTROLLER COMMAND TIME OUT (QUEUED ONLY)
1092          100000                  E.MDS=  BIT15                   ;MULTIPLE DRIVE SELECT
1093
1094  002606  000000                  O.WAIT: .WORD   0               ;PARAMETER BLOCK OF THE DRIVE
1095                                                                  ;WAITING FOR COMMAND COMPLETION
1096  002610  000400                  W.MTIM: .WORD   400             ;LOOP COUNTER FOR MILISECOND SCAN OF DRIVE
1097  002612  000400                  W.MILI: .WORD   400             ;16 MILISECOND TIME FOR PROGRAM
1098                                                                  ;
1099                                                                  ;       CPU     VALUE
1100                                                                  ;       ---     -----
1101                                                                  ;
1102                                                                  ;       11/05   100
1103                                                                  ;       11/10
1104                                                                  ;       11/20
1105                                                                  ;       11/34
1106                                                                  ;       11/40
1107                                                                  ;       11/45   400
1108                                                                  ;       11/50
1109                                                                  ;       11/70
```

```
1110  002614  000100                  W.SEC:  .WORD   100             ;SECOND COUNT COUNT FOR ALL COMMANDS
1111                                                                  ;  EXCEPT START SPINDLE
1112  002616  001000                  W.8SEC: .WORD   1000            ;8 SECOND FOR DRIVE CYCLE DOWN
1113  002620  010000                  W.MIN:  .WORD   10000           ;MINUTE TIME FOR START SPINDLE
1114  002622  000000                  HDR.AD: .WORD   0               ;ADDRESS USED FOR READ ALL HEADERS
1115  002624  000000                  HDR.CT: .WORD   0               ;NUMBER OF HEADERS LEFT TO READ FOR READ
1116                                                                  ;  ALL HEADERS
1117  002626     000                  I.ISRL: .BYTE   0               ;INTERRUPT OR RELEASED COMMAND ISSUED
1118  002627     002     004     010  H.HEAD: .BYTE   2,4,10          ;HEAD DECODES
1119  002632     000                  W.TIME: .BYTE   0               ;DRIVES BEING WATCH-DOG TIMED
1120
1121                                  .SBTTL  INTERRUPT MASKS
1122
1123  002633     000                  INTMSK: .BYTE   0               ;INTERRUPT MASKS FOR DRIVE IN PARAMETER BLOCK
1124
1125                                  ;       INTERRUPT MASK TABLE
1126
1127  002634     001                  I.DRV:  .BYTE   1               ;INTERRUPT MASK FOR DRIVE 0
1128  002635     002                          .BYTE   2               ;INTERRUPT MASK FOR DRIVE 1
1129  002636     004                          .BYTE   4               ;INTERRUPT MASK FOR DRIVE 2
1130  002637     010                          .BYTE   10              ;INTERRUPT MASK FOR DRIVE 3
1131  002640     020                          .BYTE   20              ;INTERRUPT MASK FOR DRIVE 4
1132  002641     040                          .BYTE   40              ;INTERRUPT MASK FOR DRIVE 5
1133  002642     100                          .BYTE   100             ;INTERRUPT MASK FOR DRIVE 6
1134  002643     200                          .BYTE   200             ;INTERRUPT MASK FOR DRIVE 7
1135
1136                                  .SBTTL  PARAMETER BLOCK TABLE
1137
1138  002644  002362                  PBLKT:  PARM0                   ;ADDRESS OF PARAMETER BLOCK GIVEN WITH
1139                                                                  ;DRIVE CALL.  MUST BE LOADED INTO PBLKT
1140
1141
1142                                  .SBTTL  TIME FOR WATCH-DOG TIMER
1143
1144  002646  000000                  W.DRV:  .WORD   0               ;TIME FOR INSTRUCTION IN PARAMETER BLOCK
1145                                  .SBTTL  PROGRAM SPECIFIC RESERVED LOCATIONS
1146  002650     000                  DERCNT: .BYTE   0               ;DATA ERROR COUNT
1147  002651     000                  OPCOMP: .BYTE   0               ;OPERATION COMPLETE FLAG
1148  002652     000                  DONE:   .BYTE   0               ;DONE SWITCH
1149  002653     000                  TYPFMT: .BYTE   0               ;DRIVE TYPE & FORMAT CONTROL
1150  002654     000                  ERRCNT: .BYTE   0               ;ERROR COUNT
1151  002655     004                  ERRLMT: .BYTE   4               ;ERROR LIMIT
1152  002656     000                  OPCONT: .BYTE   0               ;OPERATION CONTROL SWITCHES
1153             000001               WHDSW=BIT0                      ;WRITE HEADER & DATA SWITCH
1154             000002               VFHDSW=BIT1                     ;VERIFY HEADERS SWITCH
1155             000004               WCDASW=BIT2                     ;WRITE CHECK DATA SWITCH
1156             000010               RCDASW=BIT3                     ;READ CHECK DATA SWITCH
1157             000020               OREQSW=BIT4                     ;OFFSET REQUIRED SWITCH
1158             000040               PSDBSF= BIT5            ;SAVE SOFTWARE DETECTED BAD SECTOR FILES SWITCH
1159             002660                       .EVEN
1160
1161  002660  000400                  RDBUF:  .BLKW   ^D256           ;READ BUFFER
1162  003660  000400                  BSSOFT: .BLKW   ^D256           ;RECORD OF BAD SECTORS FROM SOFTWARE
1163  004660  000400                  BSFACT: .BLKW   ^D256           ;RECORD OF BAD SECTORS FROM FACTORY
1164  005660  000006                  COMSTR: .BLKW   6               ;COMMAND STORAGE
1165  005674  000102                  BUFFO:  .BLKW   ^D66            ;OUTPUT BUFFER 1
```

```
1166  006100  000102                  BUFF1:  .BLKW   ↑D66            ;OUTPUT BUFFER 2
1167  006304  000000                  BUFPRT: .WORD   0               ;BUFFER POINTER
1168  006306  000000                  TKWDCT: .WORD   0               ;FULL TRACK WORD COUNT STORAGE
1169  006310  000006                  INITTP: .WORD   6               ;DEFAULT DRIVE TYPE
1170  006312  000000                  INITDR: .WORD   0               ;DEFAULT DRIVE
1171  006314  000026                  INITSE: .WORD   26              ;DEFAULT SECTOR/TRACK
1172  006316  000002                  INITMD: .WORD   2               ;DEFAULT MODE
1173  006320  135143                  INITWE: .WORD   135143          ;DEFAULT DATA FOR EVEN CYL
1174  006322  072307                  INITWO: .WORD   072307          ;DEFAULT DATA FOR ODD CYL
1175  006324  000000                  INITST: .WORD   0               ;DEFAULT STARTING TRACK
1176  006326  000002                  INITET: .WORD   2               ;DEFAULT ENDING TRACK
1177  006330  000000                  INITSC: .WORD   0               ;DEFAULT STARTING CYLINDER
1178  006332  000632                  INITEC: .WORD   632             ;DEFAULT END CYLINDER
1179  006334  000000                  INITOF: .WORD   0               ;DEFAULT OFFSET (+0)
1180  006336  000000                  TPINUS: .WORD   0               ;DRIVE TYPE IN USE
1181  006340  000000                  DRINUS: .WORD   0               ;DRIVE IN USE
1182  006342  000000                  SEINUS: .WORD   0               ;SECTOR/TRACK IN USE
1183  006344  000000                  MDINUS: .WORD   0               ;MODE IN USE
1184  006346  000000                  WEINUS: .WORD   0               ;DATA IN USE FOR EVEN CYL
1185  006350  000000                  WOINUS: .WORD   0               ;DATA IN USE FOR ODD CYL
1186  006352  000000                  STINUS: .WORD   0               ;STARTING TRACK IN USE
1187  006354  000000                  ETINUS: .WORD   0               ;ENDING TRACK IN USE
1188  006356  000000                  SCINUS: .WORD   0               ;STARTING CYLINDER IN USE
1189  006360  000000                  ECINUS: .WORD   0               ;END CYLINDER IN USE
1190  006362  000000                  OFINUS: .WORD   0               ;OFFSET VALUE IN USE
1191  006364  000000                  RECODE: .WORD   0               ;RECOVERY CODE WORD
1192  006366  000000                  CTINUS: .WORD   0               ;CURRENT TRACK IN USE
1193  006370  000000                  CCINUS: .WORD   0               ;CURRENT CYLINDER IN USE
1194  006372  000000                  ERRCOM: .WORD   0               ;ERROR COMMAND
1195          000002                  BSERR=BIT1                      ;BSE ERROR
1196          000004                  HVRCER=BIT2                     ;HVRC ERROR
1197          000010                  OPIERR=BIT3                     ;OPI ERROR
1198          000020                  DCKERR=BIT4                     ;DATA CHECK ERROR
1199          000040                  ECCNC=BIT5                      ;ECC NON-CORRECTABLE
1200          000100                  WCERR=BIT6                      ;WRITE CHECK ERROR
1201          000200                  ABORT=BIT7                      ;ABORT
1202          000400                  LEV2ER=BIT8                     ;LEVEL TWO ERROR
1203          001000                  BADSEC=BIT9                     ;BAD SECTOR FLAG
1204          002000                  TWOTOS=BIT10                    ;TWO TIME OUTS
1205          004000                  RCLREQ=BIT11                    ;RECALIBRATE REQUIRED
1206          100000                  ANYDER=BIT15                    ;ANY ERROR DETECTED FLAG
1207
1208  006374  037477     000          QUES:   .ASCIZ  /??/
1209  006377     015  044012  046105  HELPFL: .ASCII  <15><12>/HELP FILE/<15><12><12>
1210  006404  020120  044506  042514
1211  006412  005015     012
1212  006415     115  052517  052116          .ASCII  /MOUNT AN RK06K DISK CARTRIDGE/<15><12>
1213  006422  040440  020116  045522
1214  006430  033060  020113  044504
1215  006436  045523  041440  051101
1216  006444  051124  042111  042507
1217  006452  005015
1218  006454  042522  042523  020124          .ASCII  /RESET WRITE LOCK IF WRITE HEADERS OR DATA OPERATION/<15><12>
1219  006462  051127  052111  020105
1220  006470  047514  045503  044440
1221  006476  020106  051127  052111
```

```
1222  006504  020105  042510  042101
1223  006512  051105  020123  051117
1224  006520  042040  052101  020101
1225  006526  050117  051105  052101
1226  006534  047511  006516     012
1227  006541     122  051505  047520          .ASCII  /RESPOND TO THE FOLLOWING PARAMETER REQUESTS BY:/
1228  006546  042116  052040  020117
1229  006554  044124  020105  047506
1230  006562  046114  053517  047111
1231  006570  020107  040520  040522
1232  006576  042515  042524  020122
1233  006604  042522  052521  051505
1234  006612  051524  041040  035131
1235  006620  005015  047503  052116          .ASCII  <15><12>/CONTROL Z (↑Z) (CR) TO ALLOW ALL OR /
1236  006626  047522  020114  020132
1237  006634  057050  024532  024040
1238  006642  051103  020051  047524
1239  006650  040440  046114  053517
1240  006656  040440  046114  047440
1241  006664  020122
1242  006666  042522  040515  047111          .ASCII  /REMAINING PARAMETERS TO DEFAULT/<15><12>
1243  006674  047111  020107  040520
1244  006702  040522  042515  042524
1245  006710  051522  052040  020117
1246  006716  042504  040506  046125
1247  006724  006524     012
1248  006727     103  051101  044522          .ASCII  /CARRIAGE RETURN TO DEFAULT THAT SPECIFIC PATAMETER/<15><12>
1249  006734  043501  020105  042522
1250  006742  052524  047122  052040
1251  006750  020117  042504  040506
1252  006756  046125  020124  044124
1253  006764  052101  051440  042520
1254  006772  044503  044506  020103
1255  007000  040520  040524  042515
1256  007006  042524  006522     012
1257  007013     103  047117  051124          .ASCII  /CONTROL C (↑C) TO RETURN TO FIRST PARAMETER REQUEST/<15><12>
1258  007020  046117  041440  024040
1259  007026  041536  020051  047524
1260  007034  051040  052105  051125
1261  007042  020116  047524  043040
1262  007050  051111  052123  050040
1263  007056  051101  046501  052105
1264  007064  051105  051040  050505
1265  007072  042525  052123  005015
1266  007100  047503  052116  047522          .ASCII  /CONTROL C (↑C) ALSO ABORTS THE PROGRAM ONCE /
1267  007106  020114  020103  057050
1268  007114  024503  040440  051514
1269  007122  020117  041101  051117
1270  007130  051524  052040  042510
1271  007136  050040  047522  051107
1272  007144  046501  047440  041516
1273  007152  020105
1274  007154  047506  046522  052101          .ASCII  /FORMATTING HAS STARTED/<15><12><12>
1275  007162  044524  043516  044040
1276  007170  051501  051440  040524
1277  007176  052122  042105  005015
```

```
1278  007204    012
1279  007205    117  006522  005012          .ASCII  /OR/<15><12><12>
1280  007212  047105  042524  020122          .ASCII  /ENTER THE DESIRED PARAMETER OPTION SELECTED /
1281  007220  044124  020105  042504
1282  007226  044523  042522  020104
1283  007234  040520  040522  042515
1284  007242  042524  020122  050117
1285  007250  044524  047117  051440
1286  007256  046105  041505  042524
1287  007264  020104
1288  007266  051106  046517  052040          .ASCII  /FROM THE LIST BELOW./<15><12>
1289  007274  042510  046040  051511
1290  007302  020124  042502  047514
1291  007310  027127  005015
1292  007314  046101  020114  040526          .ASCII  /ALL VALUES TO BE ENTERED ARE OCTAL/<15><12><12>
1293  007322  052514  051505  052040
1294  007330  020117  042502  042440
1295  007336  052116  051105  042105
1296  007344  040440  042522  047440
1297  007352  052103  046101  005015
1298  007360    012
1299  007361    104  044522  042526          .ASCII  /DRIVE TYPE=(0 - 7) DEFAULTS TO 6 FOR RK06/<15><12>
1300  007366  052040  050131  036505
1301  007374  030050  026440  033440
1302  007402  020051  042504  040506
1303  007410  046125  051524  052040
1304  007416  020117  020066  047506
1305  007424  020122  045522  033060
1306  007432  005015
1307  007434  051104  053111  020105          .ASCII  /DRIVE NUM=(0-7) DEFAULTS TO 0/<15><12><12>
1308  007442  052516  036515  030050
1309  007450  033455  020051  042504
1310  007456  040506  046125  051524
1311  007464  052040  020117  006460
1312  007472  005012
1313  007474  042523  052103  051117          .ASCII  &SECTORS/TRACK=(24,26) DEFAULTS TO 26&<15><12><12>
1314  007502  027523  051124  041501
1315  007510  036513  031050  026064
1316  007516  033062  020051  042504
1317  007524  040506  046125  051524
1318  007532  052040  020117  033062
1319  007540  005015    012
1320  007543    115  042117  036505          .ASCII  /MODE=(0-4) DEFAULTS TO 2/<15><12>
1321  007550  030050  032055  020051
1322  007556  042504  040506  046125
1323  007564  051524  052040  020117
1324  007572  006462    012
1325  007575    040  020040  030040          .ASCII  /    0=WRITE HEADERS & DATA/<15><12>
1326  007602  053475  044522  042524
1327  007610  044040  040505  042504
1328  007616  051522  023040  042040
1329  007624  052101  006501    012
1330  007631    040  020040  030440          .ASCII  /    1=WRITE HEADERS & DATA; VERIFY HEADERS/<15><12>
1331  007636  053475  044522  042524
1332  007644  044040  040505  042504
1333  007652  051522  023040  042040
```

```
1334  007660  052101  035501  053040
1335  007666  051105  043111  020131
1336  007674  042510  042101  051105
1337  007702  006523     012
1338  007705     040  020040  031040          .ASCII  /    2=WRITE & VERIFY HEADERS & DATA/<15><12>
1339  007712  053475  044522  042524
1340  007720  023040  053040  051105
1341  007726  043111  020131  042510
1342  007734  042101  051105  020123
1343  007742  020046  040504  040524
1344  007750  005015
1345  007752  020040  020040  036463          .ASCII  /    3=VERIFY HEADERS/<15><12>
1346  007760  042526  044522  054506
1347  007766  044040  040505  042504
1348  007774  051522  005015
1349  010000  020040  020040  036464          .ASCII  /    4=VERIFY HEADERS AND VERIFY DATA/<15><12><12>
1350  010006  042526  044522  054506
1351  010014  044040  040505  042504
1352  010022  051522  040440  042116
1353  010030  053040  051105  043111
1354  010036  020131  040504  040524
1355  010044  005015     012
1356  010047     105  042526  020116          .ASCII  /EVEN CYLINDER PATTERN=(DATA WORD) DEFAULTS TO 135143/<15><12>
1357  010054  054503  044514  042116
1358  010062  051105  050040  052101
1359  010070  042524  047122  024075
1360  010076  040504  040524  053440
1361  010104  051117  024504  042040
1362  010112  043105  052501  052114
1363  010120  020123  047524  030440
1364  010126  032463  032061  006463
1365  010134     012
1366  010135     117  042104  041440          .ASCII  /ODD CYLINDER PATTERN=(DATA WORD) DEFAULTS TO 072307/<15><12>
1367  010142  046131  047111  042504
1368  010150  020122  040520  052124
1369  010156  051105  036516  042050
1370  010164  052101  020101  047527
1371  010172  042122  020051  042504
1372  010200  040506  046125  051524
1373  010206  052040  020117  033460
1374  010214  031462  033460  005015
1375  010222  051124  041501  020113          .ASCII  /TRACK LIMITS = (0-2,0-2) DEFAULTS TO 0,2/<15><12>
1376  010230  044514  044515  051524
1377  010236  036440  024040  026460
1378  010244  026062  026460  024462
1379  010252  042040  043105  052501
1380  010260  052114  020123  047524
1381  010266  030040  031054  005015
1382  010274  054503  044514  042116          .ASCII  /CYLINDER LIMITS = (0-632,0-632) DEFAULTS TO 0,632/<15><12><12>
1383  010302  051105  046040  046511
1384  010310  052111  020123  020075
1385  010316  030050  033055  031063
1386  010324  030054  033055  031063
1387  010332  020051  042504  040506
1388  010340  046125  051524  052040
1389  010346  020117  026060  031466
```

```
1390  010354  036462  005012
1391  010360  043117  051506  052105          .ASCII  /OFFSET=(OCTAL OFFSET VALUE) DEFAULTS TO C (0 OFFSET)/<15><12>
1392  010366  024075  041517  040524
1393  010374  020114  043117  051506
1394  010402  052105  053040  046101
1395  010410  042525  020051  042504
1396  010416  040506  046125  051524
1397  010424  052040  020117  020060
1398  010432  030050  047440  043106
1399  010440  042523  024524  005015
1400  010446  020040  020040  020040          .ASCII  /      000 AND 100 = BOTH ARE ZERO OFFSET/<15><12>
1401  010454  030060  020060  047101
1402  010462  020104  030061  020060
1403  010470  020075  047502  044124
1404  010476  040440  042522  055040
1405  010504  051105  020117  043117
1406  010512  051506  052105  005015
1407  010520  020040  020040  020040          .ASCII  /      160 = -1200 MICRO INCHES/<15><12>
1408  010526  033061  020060  020075
1409  010534  030455  030062  020060
1410  010542  044515  051103  020117
1411  010550  047111  044103  051505
1412  010556  005015
1413  010560  020040  020040  020040          .ASCII  /      060 = +1200 MICRO INCHES/<15><12>
1414  010566  033060  020060  020075
1415  010574  030453  030062  020060
1416  010602  044515  051103  020117
1417  010610  047111  044103  051505
1418  010616  005015
1419  010620  020040  020040  020040          .ASCII  /      EACH COUNT IS AN OFFSET INCREMENT OF 25 MICRO INCHES/<15><12><12>
1420  010626  040505  044103  041440
1421  010634  052517  052116  044440
1422  010642  020123  047101  047440
1423  010650  043106  042523  020124
1424  010656  047111  051103  046505
1425  010664  047105  020124  043117
1426  010672  031040  020065  044515
1427  010700  051103  020117  047111
1428  010706  044103  051505  005015
1429  010714     012
1430  010715     101  054516  051440          .ASCII  /ANY SECTORS TO BE FLAGED BAD? (TYPE Y OR N)/<15><12>
1431  010722  041505  047524  051522
1432  010730  052040  020117  042502
1433  010736  043040  040514  042507
1434  010744  020104  040502  037504
1435  010752  024040  054524  042520
1436  010760  054440  047440  020122
1437  010766  024516  005015
1438  010772  020040  020040  020040          .ASCII  /      IF N IS TYPED, THE FORMAT OPERATION IS INITIALIZED/<15><12>
1439  011000  043111  047040  044440
1440  011006  020123  054524  042520
1441  011014  026104  052040  042510
1442  011022  043040  051117  040515
1443  011030  020124  050117  051105
1444  011036  052101  047511  020116
1445  011044  051511  044440  044516
```

```
1446  011052  044524  046101  055111
1447  011060  042105  005015
1448  011064  020040  020040  020040          .ASCII  /      AND ONLY THOSE SECTORS FOUND BAD BY THE FORMAT/<15><12>
1449  011072  047101  020104  047117
1450  011100  054514  052040  047510
1451  011106  042523  051440  041505
1452  011114  047524  051522  043040
1453  011122  052517  042116  041040
1454  011130  042101  041040  020131
1455  011136  044124  020105  047506
1456  011144  046522  052101  005015
1457  011152  020040  020040  020040          .ASCII  /      PROGRAM AND THOSE LISTED IN THE FACTORY TABLE/<15><12>
1458  011160  051120  043517  040522
1459  011166  020115  047101  020104
1460  011174  044124  051517  020105
1461  011202  044514  052123  042105
1462  011210  044440  020116  044124
1463  011216  020105  040506  052103
1464  011224  051117  020131  040524
1465  011232  046102  006505     012
1466  011237     040  020040  020040          .ASCII  /      ARE FLAGED BAD. IF Y IS TYPED, THE USER HAS THE/<15><12>
1467  011244  040440  042522  043040
1468  011252  040514  042507  020104
1469  011260  040502  027104  044440
1470  011266  020106  020131  051511
1471  011274  052040  050131  042105
1472  011302  020054  044124  020105
1473  011310  051525  051105  044040
1474  011316  051501  052040  042510
1475  011324  005015
1476  011326  020040  020040  020040          .ASCII  /      CAPABILITY OF INSTRUCTING THE FORMATTER TO FLAG/<15><12>
1477  011334  040503  040520  044502
1478  011342  044514  054524  047440
1479  011350  020106  047111  052123
1480  011356  052522  052103  047111
1481  011364  020107  044124  020105
1482  011372  047506  046522  052101
1483  011400  042524  020122  047524
1484  011406  043040  040514  006507
1485  011414     012
1486  011415     040  020040  020040          .ASCII  /      SPECIFIED SECTORS AS BAD. THE PROGRAM WILL/<15><12>
1487  011422  051440  042520  044503
1488  011430  044506  042105  051440
1489  011436  041505  047524  051522
1490  011444  040440  020123  040502
1491  011452  027104  052040  042510
1492  011460  050040  047522  051107
1493  011466  046501  053440  046111
1494  011474  006514     012
1495  011477     040  020040  020040          .ASCII  /      DIRECT THE USER IN HOW THIS IS DONE./<15><12><12>
1496  011504  042040  051111  041505
1497  011512  020124  044124  020105
1498  011520  051525  051105  044440
1499  011526  020116  047510  020127
1500  011534  044124  051511  044440
1501  011542  020123  047504  042516
```

```
1502  011550  006456  005012
1503  011554  051120  051505  051105          .ASCII  /PRESERVE SOFTWARE BAD SECTOR FILE?(TYPE Y OR N)(CR)/<15><12>
1504  011562  042526  051440  043117
1505  011570  053524  051101  020105
1506  011576  040502  020104  042523
1507  011604  052103  051117  043040
1508  011612  046111  037505  052050
1509  011620  050131  020105  020131
1510  011626  051117  047040  024051
1511  011634  051103  006451     012
1512  011641     011  043111  054440          .ASCII  /       IF Y IS TYPED, THE FORMATTER WILL PRESERVE THE/<15><12>
1513  011646  044440  020123  054524
1514  011654  042520  026104  052040
1515  011662  042510  043040  051117
1516  011670  040515  052124  051105
1517  011676  053440  046111  020114
1518  011704  051120  051505  051105
1519  011712  042526  052040  042510
1520  011720  005015
1521  011722  040411  050120  044514          .ASCII  /       APPLICABLE BAD SECTOR FILE, MARKING THE SECTORS/<15><12>
1522  011730  040503  046102  020105
1523  011736  040502  020104  042523
1524  011744  052103  051117  043040
1525  011752  046111  026105  046440
1526  011760  051101  044513  043516
1527  011766  052040  042510  051440
1528  011774  041505  047524  051522
1529  012002  005015
1530  012004  041011  042101  052040          .ASCII  /       BAD THAT ARE LISTED IN THE FILE. ANY ADDITIONAL/<15><12>
1531  012012  040510  020124  051101
1532  012020  020105  044514  052123
1533  012026  042105  044440  020116
1534  012034  044124  020105  044506
1535  012042  042514  020056  047101
1536  012050  020131  042101  044504
1537  012056  044524  047117  046101
1538  012064  005015
1539  012066  051411  041505  047524          .ASCII  /       SECTORS FOUND BAD BY THE PROGRAM ARE APPENDED/<15><12>
1540  012074  051522  043040  052517
1541  012102  042116  041040  042101
1542  012110  041040  020131  044124
1543  012116  020105  051120  043517
1544  012124  040522  020115  051101
1545  012132  020105  050101  042520
1546  012140  042116  042105  005015
1547  012146  052011  020117  044124          .ASCII  /       TO THE FILE. IF N IS TYPED THE BAD SECTOR FILE/<15><12>
1548  012154  020105  044506  042514
1549  012162  020056  043111  047040
1550  012170  044440  020123  054524
1551  012176  042520  020104  044124
1552  012204  020105  040502  020104
1553  012212  042523  052103  051117
1554  012220  043040  046111  006505
1555  012226     012
1556  012227     011  051511  041440          .ASCII  /       IS CLEARED BEFORE THE PROGRAM BEGINS./<15><12><12>
1557  012234  042514  051101  042105
```

```
1558  012242  041040  043105  051117
1559  012250  020105  044124  020105
1560  012256  051120  043517  040522
1561  012264  020115  042502  044507
1562  012272  051516  006456  005012
1563  012300  042523  020124  053523          .ASCII  /SET SWITCH 7 TO PRINT A LIST OF ALL BAD SECTORS./<15><12><12>
1564  012306  052111  044103  033440
1565  012314  052040  020117  051120
1566  012322  047111  020124  020101
1567  012330  044514  052123  047440
1568  012336  020106  046101  020114
1569  012344  040502  020104  042523
1570  012352  052103  051117  027123
1571  012360  005015     012
1572  012363     123  052105  051440          .ASCII  /SET SWITCH 1 TO FORCE REPORTING OF ALL DATA IN ERROR/<15><12><12>
1573  012370  044527  041524  020110
1574  012376  020061  047524  043040
1575  012404  051117  042503  051040
1576  012412  050105  051117  044524
1577  012420  043516  047440  020106
1578  012426  046101  020114  040504
1579  012434  040524  044440  020116
1580  012442  051105  047522  006522
1581  012450  005012
1582  012452  042523  020124  053523          .ASCII  /SET SWITCH 0 TO REPEAT THE ENTIRE FORMAT OPERATION/<15><12><12>
1583  012460  052111  044103  030040
1584  012466  052040  020117  042522
1585  012474  042520  052101  052040
1586  012502  042510  042440  052116
1587  012510  051111  020105  047506
1588  012516  046522  052101  047440
1589  012524  042520  040522  044524
1590  012532  047117  005015     012
1591  012537     111  020106  044124          .ASCII  /IF THE SOFTWARE SWITCH REGISTER IS IN USE, /<15><12>
1592  012544  020105  047523  052106
1593  012552  040527  042522  051440
1594  012560  044527  041524  020110
1595  012566  042522  044507  052123
1596  012574  051105  044440  020123
1597  012602  047111  052440  042523
1598  012610  020054  005015
1599  012614  054524  044520  043516          .ASCII  /TYPING CONTROL G (↑G) CAUSES THE PROGRAM/<15><12>
1600  012622  041440  047117  051124
1601  012630  046117  043440  024040
1602  012636  043536  020051  040503
1603  012644  051525  051505  052040
1604  012652  042510  050040  047522
1605  012660  051107  046501  005015
1606  012666  047524  051040  050505          .ASCII  /TO REQUEST NEW SWR SETTING/<15><12><12>
1607  012674  042525  052123  047040
1608  012702  053505  051440  051127
1609  012710  051440  052105  044524
1610  012716  043516  005015     012
1611  012723     105  042116  044040          .ASCIZ  /END HELP FILE/<15><12><12>
1612  012730  046105  020120  044506
1613  012736  042514  005015  000012
```

```
1614  012744  020040     000          SPACE2: .WSCIZ  /  /
1615
1616  012747     015  025012  020052  PROGID: .ASCII  <15><12>/** RK06K CARTRIDGE FORMATER **/<15><12>
1617  012754  045522  033060  020113
1618  012762  040503  052122  044522
1619  012770  043504  020105  047506
1620  012776  046522  052101  051105
1621  013004  025040  006452     012
1622  013011     115  044501  042116          .ASCIZ  /MAINDEC-11-DZR6L-B/<15><12>
1623  013016  041505  030455  026461
1624  013024  055104  033122  026514
1625  013032  006502  000012
1626  013036  054524  042520  044040  HELPQ:  .ASCIZ  /TYPE HELP FOR OPERATING INFO, ELSE CR/<15><12>
1627  013044  046105  020120  047506
1628  013052  020122  050117  051105
1629  013060  052101  047111  020107
1630  013066  047111  047506  020054
1631  013074  046105  042523  041440
1632  013102  006522  000012
1633  013106  051104  053111  020105  TPQ:     .ASCIZ  /DRIVE TYPE=/
1634  013114  054524  042520  000075
1635  013122  051104  053111  020105  DRVQ:    .ASCIZ  /DRIVE NUM=/
1636  013130  052516  036515     000
1637  013135     123  041505  047524  SECTQ:   .ASCIZ  &SECTOR/TRACK=&
1638  013142  027522  051124  041501
1639  013150  036513     000
1640  013153     115  042117  036505  MODEQ:   .ASCIZ  /MODE=/
1641  013160     000
1642  013161     105  042526  020116  WDEQ:    .ASCIZ  /EVEN CYLINDER PATTERN=/
1643  013166  054503  044514  042116
1644  013174  051105  050040  052101
1645  013202  042524  047122  000075
1646  013210  042117  020104  054503  WDOD:    .ASCIZ  /ODD CYLINDER PATTERN=/
1647  013216  044514  042116  051105
1648  013224  050040  052101  042524
1649  013232  047122  000075
1650  013236  051124  041501  020113  TRKLIM:  .ASCIZ  /TRACK LIMITS=/
1651  013244  044514  044515  051524
1652  013252  000075
1653  013254  054503  044514  042116  CYLLIM:  .ASCIZ  /CYLINDER LIMITS=/
1654  013262  051105  046040  046511
1655  013270  052111  036523     000
1656  013275     117  043106  042523  OFSETQ:  .ASCIZ  /OFFSET=/
1657  013302  036524     000
1658  013305     120  042522  042523  PSDQ:    .ASCIZ  /PRESERVE SOFTWARE BAD SECTOR FILES?(TYPE Y OR N)(CR)/
1659  013312  053122  020105  047523
1660  013320  052106  040527  042522
1661  013326  041040  042101  051440
1662  013334  041505  047524  020122
1663  013342  044506  042514  037523
1664  013350  052050  050131  020105
1665  013356  020131  051117  047040
1666  013364  024051  051103  000051
1667  013372  047523  052106  040527  SDBSFP:  .ASCIZ  /SOFTWARE DETECTED BAD SECTOR FILES /
1668  013400  042522  042040  052105
1669  013406  041505  042524  020104
```

```
1670  013414  040502  020104  042523
1671  013422  052103  051117  043040
1672  013430  046111  051505  000040
1673  013436  051120  051505  051105  PRSVD:  .ASCIZ  /PRESERVED/<15><12>
1674  013444  042526  006504  000012
1675  013452  047516  020124  051120  NPRSVD: .ASCIZ  /NOT PRESERVED/<15><12>
1676  013460  051505  051105  042526
1677  013466  006504  000012
1678  013472  047101  020131  042523  UBSQ:   .ASCIZ  /ANY SECTORS TO BE FLAGED BAD? (TYPE Y OR N)(CR)/
1679  013500  052103  051117  020123
1680  013506  047524  041040  020105
1681  013514  046106  043501  042105
1682  013522  041040  042101  020077
1683  013530  052050  050131  020105
1684  013536  020131  051117  047040
1685  013544  024051  051103  000051
1686  013552  047105  042524  020122  UBSFMT: .ASCIZ  /ENTER OCTAL ADDRESS TO BE FLAGED BAD IN FORMAT CCC,T,SS/<15><12>
1687  013560  041517  040524  020114
1688  013566  042101  051104  051505
1689  013574  020123  047524  041040
1690  013602  020105  046106  043501
1691  013610  042105  041040  042101
1692  013616  044440  020116  047506
1693  013624  046522  052101  041440
1694  013632  041503  052054  051454
1695  013640  006523  000012
1696  013644  047105  042524  020122  NEXTQ:  .ASCIZ  /ENTER ADDRESS OR (CR) TO TERMINATE/<15><12>
1697  013652  042101  051104  051505
1698  013660  020123  051117  024040
1699  013666  051103  020051  047524
1700  013674  052040  051105  044515
1701  013702  040516  042524  005015
1702  013710     000
1703  013711     111  053116  046101  BADENT: .ASCIZ  /INVALID ENTRY/<15><12>
1704  013716  042111  042440  052116
1705  013724  054522  005015     000
1706  013731     111  053116  020104  ILLSEC: .ASCIZ  /INVD SECTOR NUMBER/<15><12>
1707  013736  042523  052103  051117
1708  013744  047040  046525  042502
1709  013752  006522  000012
1710  013756  047111  042126  051440  INVDSE: .ASCIZ  /INVD STARTING OR ENDING PACK ADDRESS/<15><12>
1711  013764  040524  052122  047111
1712  013772  020107  051117  042440
1713  014000  042116  047111  020107
1714  014006  040520  045503  040440
1715  014014  042104  042522  051523
1716  014022  005015     000
1717  014025     111  053116  020104  INVDCY: .ASCIZ  /INVD CYLINDER NUMBER /<15><12>
1718  014032  054503  044514  042116
1719  014040  051105  047040  046525
1720  014046  042502  020122  005015
1721  014054     000
1722  014055     111  053116  020104  INVDTK: .ASCIZ  /INVD TRACK NUMBER /<15><12>
1723  014062  051124  041501  020113
1724  014070  052516  041115  051105
1725  014076  006440  000012
```

```
1726  014102  047111  042126  042040  INVDDR: .ASCIZ  /INVD DRIVE NUMBER/<15><12>
1727  014110  044522  042526  047040
1728  014116  046525  042502  006522
1729  014124  000012
1730  014126  047111  042126  042040  INVDTP: .ASCIZ  /INVD DRIVE TYPE/<15><12>
1731  014134  044522  042526  052040
1732  014142  050131  006505  000012
1733  014150  047111  042126  046440  INVDMD: .ASCIZ  /INVD MODE SELECTION (MAX OF 4)/<15><12>
1734  014156  042117  020105  042523
1735  014164  042514  052103  047511
1736  014172  020116  046450  054101
1737  014200  047440  020106  024464
1738  014206  005015     000
1739  014211     111  053116  020104  INVDOF: .ASCIZ  /INVD OFFSET VALUE/<15><12>
1740  014216  043117  051506  052105
1741  014224  053040  046101  042525
1742  014232  005015     000
1743  014235     120  042514  051501  SWLREQ: .ASCIZ  /PLEASE SET WRITE LOCK FOR VERIFY OPERATIONS/<15><12>
1744  014242  020105  042523  020124
1745  014250  051127  052111  020105
1746  014256  047514  045503  043040
1747  014264  051117  053040  051105
1748  014272  043111  020131  050117
1749  014300  051105  052101  047511
1750  014306  051516  005015     000
1751  014313     120  042514  051501  RWLREQ: .ASCIZ  /PLEASE RESET WRITE LOCK FOR FORMAT OPERATIONS/<15><12>
1752  014320  020105  042522  042523
1753  014326  020124  051127  052111
1754  014334  020105  047514  045503
1755  014342  043040  051117  043040
1756  014350  051117  040515  020124
1757  014356  050117  051105  052101
1758  014364  047511  051516  005015
1759  014372     000
1760  014373     120  042522  051523  CONREQ: .ASCIZ  /PRESS CONTINUE WHEN READY/<15><12>
1761  014400  041440  047117  044524
1762  014406  052516  020105  044127
1763  014414  047105  051040  040505
1764  014422  054504  005015     000
1765  014427     103  047101  047516  BAD632: .ASCIZ  /CANNOT READ BAD SECTOR TRACK/<15><12>
1766  014434  020124  042522  042101
1767  014442  041040  042101  051440
1768  014450  041505  047524  020122
1769  014456  051124  041501  006513
1770  014464  000012
1771  014466  047523  052106  040527  TDMSOF: .ASCIZ  /SOFTWARE BAD SECTOR FILE NOT FORMATTED PROPERLY OR/<15><12>
1772  014474  042522  041040  042101
1773  014502  051440  041505  047524
1774  014510  020122  044506  042514
1775  014516  047040  052117  043040
1776  014524  051117  040515  052124
1777  014532  042105  050040  047522
1778  014540  042520  046122  020131
1779  014546  051117  005015     000
1780  014553     106  046111  020105  TDMSUF: .ASCIZ  /FILE IS TOO LARGE FOR THE PROGRAM ( )100 ENTRIES)/<15><12>
1781  014560  051511  052040  047517
```

```
1782  014566  046040  051101  042507
1783  014574  043040  051117  052040
1784  014602  042510  050040  047522
1785  014610  051107  046501  024040
1786  014616  037040  030061  020060
1787  014624  047105  051124  042511
1788  014632  024523  005015     000
1789  014637     106  041501  047524  TDMFAC: .ASCIZ  /FACTORY BAD SECTOR FILE NOT FORMATTED PROPERLY OR/<15><12>
1790  014644  054522  041040  042101
1791  014652  051440  041505  047524
1792  014660  020122  044506  042514
1793  014666  047040  052117  043040
1794  014674  051117  040515  052124
1795  014702  042105  050040  047522
1796  014710  042520  046122  020131
1797  014716  051117  005015     000
1798  014723     103  051101  051124  SERNUM: .ASCIZ  /CARTRIDGE SERIAL NUMBER *****  /
1799  014730  042111  042507  051440
1800  014736  051105  040511  020114
1801  014744  052516  041115  051105
1802  014752  025040  025052  025052
1803  014760  020040     000
1804  014763     103  051101  051124  ALNPAC: .ASCII  /CARTRIDGE MOUNTED IS AN ALIGNMENT CARTRIDGE./<15><12>
1805  014770  042111  042507  046440
1806  014776  052517  052116  042105
1807  015004  044440  020123  047101
1808  015012  040440  044514  047107
1809  015020  042515  052116  041440
1810  015026  051101  051124  042111
1811  015034  042507  006456     012
1812  015041     106  051117  040515          .ASCIZ  /FORMAT IS ABORTED/<15><12><12>
1813  015046  020124  051511  040440
1814  015054  047502  052122  042105
1815  015062  005015  000012
1816  015066  047516  042040  052101  IMPER2: .ASCIZ  /NO DATA CHECK  ON SECTOR THAT WILL NOT PASS WRITE CHECK/<15><12>
1817  015074  020101  044103  041505
1818  015102  020113  047440  020116
1819  015110  042523  052103  051117
1820  015116  052040  040510  020124
1821  015124  044527  046114  047040
1822  015132  052117  050040  051501
1823  015140  020123  051127  052111
1824  015146  020105  044103  041505
1825  015154  006513  000012
1826  015160  040504  040524  041440  IMPERR: .ASCIZ  /DATA CHECK ERROR IN WRITE. IMPOSSIBLE!!!/<15><12>
1827  015166  042510  045503  042440
1828  015174  051122  051117  044440
1829  015202  020116  051127  052111
1830  015210  027105  044440  050115
1831  015216  051517  044523  046102
1832  015224  020505  020441  005015
1833  015232     000
1834  015233     105  041503  041440  CORABL: .ASCIZ  /ECC CORRECTABLE/<15><12>
1835  015240  051117  042522  052103
1836  015246  041101  042514  005015
1837  015254     000
```

```
1838  015255     012  025052  020040  PRGRS1: .ASCIZ  <12>/**  WRITE OPERATION COMPLETE  **/<15><12><12>
1839  015262  051127  052111  020105
1840  015270  050117  051105  052101
1841  015276  047511  020116  047503
1842  015304  050115  042514  042524
1843  015312  020040  025052  005015
1844  015320  000012
1845  015322  025012  020052  053040  PRGRS2: .ASCIZ  <12>/**  VERIFY OPERATION COMPLETE  **/<15><12><12>
1846  015330  051105  043111  020131
1847  015336  050117  051105  052101
1848  015344  047511  020116  047503
1849  015352  050115  042514  042524
1850  015360  020040  025052  005015
1851  015366  000012
1852  015370  020040  025052  025052  STARS:  .ASCIZ  /  *****/<15><12>
1853  015376  006452  000012
1854  015402  042412  051122  051117  BSWERR: .ASCIZ  <12>/ERROR WRITING SOFTWARE DETECTED BAD SECTOR TABLES./<15><12>
1855  015410  053440  044522  044524
1856  015416  043516  051440  043117
1857  015424  053524  051101  020105
1858  015432  042504  042524  052103
1859  015440  042105  041040  042101
1860  015446  051440  041505  047524
1861  015454  020122  040524  046102
1862  015462  051505  006456  000012
1863  015470  042523  052103  051117  SECINE: .ASCIZ  /SECTOR IN ERROR =/
1864  015476  044440  020116  051105
1865  015504  047522  020122  000075
1866  015512  041012  042101  051440  BSWTFI: .ASCIZ  <12>/BAD SECTOR TABLE WRITTEN./<15><12>
1867  015520  041505  047524  020122
1868  015526  040524  046102  020105
1869  015534  051127  052111  042524
1870  015542  027116  005015     000
1871  015547     012  044124  020105  BSHEAD: .ASCIZ  <12>/THE BAD SECTORS FOR THIS CARTRIDGE IN /
1872  015554  040502  020104  042523
1873  015562  052103  051117  020123
1874  015570  047506  020122  044124
1875  015576  051511  041440  051101
1876  015604  051124  042111  042507
1877  015612  044440  020116     000
1878  015617     040  042523  052103  BSTAIL: .ASCIZ  / SECTOR FORMAT ARE:/<15><12>
1879  015624  051117  043040  051117
1880  015632  040515  020124  051101
1881  015640  035105  005015     000
1882  015645     012  040502  020104  FBSLAB: .ASCIZ  <12>/BAD SECTORS IN FACTORY TABLE:/<15><12>
1883  015652  042523  052103  051117
1884  015660  020123  047111  043040
1885  015666  041501  047524  054522
1886  015674  052040  041101  042514
1887  015702  006472  000012
1888  015706  005015  040502  020104  SBSLAB: .ASCIZ  <15><12>/BAD SECTORS IN SOFTWARE TABLE:/<15><12>
1889  015714  042523  052103  051117
1890  015722  020123  047111  051440
1891  015730  043117  053524  051101
1892  015736  020105  040524  046102
1893  015744  035105  005015     000
```

```
1894  015751  116     047117  006505  NONE:   .ASCIZ  /NONE/<15><12>
1895  015756  000012
1896  015760  030062  000             TWENTY: .ASCIZ  /20/
1897  015763  062     000062          TWENT2: .ASCIZ  /22/
1898  015766  020040  020040  054503  COLHD:  .ASCIZ  /    CYLNDR  TRACK   SECTOR/<15><12>
1899  015774  047114  051104  020040
1900  016002  051124  041501  020113
1901  016010  020040  042523  052103
1902  016016  051117  005015  000
1903  016023  040     020040  000040  SPACE4: .ASCIZ  /    /
1904  016030  020040  000040          SPACE3: .ASCIZ  /   /
1905  016034  005015  036440  052040  TOTMES: .ASCIZ  <15><12>/ = TOTAL NUMBER OF BAD SECTORS/<15><12>
1906  016042  052117  046101  047040
1907  016050  046525  042502  020122
1908  016056  043117  041040  042101
1909  016064  051440  041505  047524
1910  016072  051522  005015  000
1911
1912          016100                  .EVEN
1913  016100                  START:
1914                          .SBTTL  INITIALIZE THE COMMON TAGS
1915                          ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1916  016100  012706  001100          MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1917  016104  005026                  CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1918  016106  022706  001140          CMP     #SWR,R6 ;;DONE?
1919  016112  001374                  BNE     .-6             ;;LOOP BACK IF NO
1920  016114  012706  001100          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
1921                          ;;INITIALIZE A FEW VECTORS
1922  016120  012737  040350  000030  MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1923  016126  012737  000340  000032  MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1924  016134  012737  042400  000034  MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1925  016142  012737  000340  000036  MOV     #340,@#TRAPVEC+2;LEVEL 7
1926  016150  012737  042030  000024  MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1927  016156  012737  000340  000026  MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1928  016164  005037  001260          CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1929  016170  112737  000001  001115  MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1930                          ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1931                          ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1932  016176  013746  000004          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1933  016202  012737  016236  000004  MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1934  016210  012737  177570  001140  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1935  016216  012737  177570  001142  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1936  016224  022777  177777  162706  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1937  016232  001012                  BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1938                                                          ;;AND  THE HARDWARE SWR IS NOT = -1
1939  016234  000403                  BR      65$             ;;BRANCH IF NO TIMEOUT
1940  016236  012716  016244  64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1941  016242  000002                  RTI
1942  016244  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1943  016252  012737  000174  001142          MOV     #DISPREG,DISPLAY
1944  016260  012637  000004  66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1945
1946  016264  004737  040520          JSR     PC,$TKINT       ;INIT KEYBOARD
1947  016270  013701  002572          MOV     RKVEC,R1        ;GET ADDRESS OF VECTOR STORAGE
1948  016274  012721  032232          MOV     #I.INTR,(R1)+   ;SET IT TO INTERRUPT HNDLR
1949  016300  012711  000340          MOV     #PR7,(R1)       ;SET INTERRUPT HANDLER PR7
```

```
1950  016304  013746  000000                  MOV     0,-(SP) ;;PUT NEW PS ON STACK
1951  016310  012746  016316                  MOV     #67$,-(SP)      ;;PUT NEW PC ON  STACK
1952  016314  000002                          RTI                     ;;POP NEW PC AND PS
1953  016316                          67$:
1954  016316  104401  012747                  TYPE    ,PROGID         ;TYPE IDENTIFICATION
1955  016322  104401  013036                  TYPE    ,HELPQ          ;TYPE HELP QUESTION
1956  016326  104410                          RDLIN                   ;GET ANSWER TO HELP Q.
1957  016330  012601                          MOV     (SP)+,R1        ;GET ADDRESS OF ANSWER
1958  016332  105711                          TSTB    (R1)            ;TEST FIRST CHAR
1959  016334  001402                          BEQ     STPARM
1960  016336  104401  006377                  TYPE    ,HELPFL         ;TYPE HELP FILE
1961
1962                                  ;;******************************************************************
1963                                  .SBTTL  PARAMETER REQUEST & CHECK ROUTINE
1964                                  ;*ENTRY:  AUTOMATIC AT START OF PROGRAM.
1965                                  ;*EXIT:  GO TO MODE DECODE.
1966                                  ;*
1967                                  ;*THIS ROUTINE REQUESTS EACH PARAMETER IN TURN AND CALLS
1968                                  ;*THE GETANS SUBROUTINE TO HANDLE THE ANSWERS AND STORE
1969                                  ;*THEM.  IT THEN CHECKS THE PARAMETERS FOR LEGALITY AND
1970                                  ;*VALIDITY AND, IF AN ERROR IS FOUND, REQUESTS THE
1971                                  ;*PARAMETERS AGAIN, STARTING WITH THE PARAMETER IN ERROR.
1972                                  ;;******************************************************************
1973
1974  016342  012702  006310          STPARM: MOV     #INITTP,R2      ;STARTING ADDRESS OF DEF. PARAM
1975  016346  012703  006336                  MOV     #TPINUS,R3      ;STARTING ADD OF ACTIVE PARAMS
1976  016352  104401  013106                  TYPE    ,TPQ            ;TYPE DRIVE TYPE QUESTION
1977  016356  004437  017610                  JSR     R4,GETANS
1978  016362  104401  013122                  TYPE    ,DRVQ           ;TYPE DRIVE QUESTION
1979  016366  004437  017610                  JSR     R4,GETANS       ;GO GET PARAM
1980  016372  104401  013135                  TYPE    ,SECTQ          ;TYPE SECTOR/TRACK QUESTION
1981  016376  004437  017610                  JSR     R4,GETANS       ;GO GET PARAM
1982  016402  104401  013153          PARAM3: TYPE    ,MODEQ          ;TYPE MODE QUES
1983  016406  004437  017610                  JSR     R4,GETANS       ;GO GET PARAM
1984  016412  104401  013161                  TYPE    ,WDEQ
1985  016416  004437  017610                  JSR     R4,GETANS
1986  016422  104401  013210                  TYPE    ,WDOD
1987  016426  004437  017610                  JSR     R4,GETANS
1988  016432  104401  013236          PARAM5: TYPE    ,TRKLIM         ;TYPE TRACK LIMITS QUESTION
1989  016436  004437  017610                  JSR     R4,GETANS       ;GO GET PARAM
1990  016442  104401  013254                  TYPE    ,CYLLIM         ;TYPE CYLINDER LIMITS QUESTION
1991  016446  004437  017610                  JSR     R4,GETANS       ;GO GET PARAM
1992  016452  104401  013275          PARM6:  TYPE    ,OFSETQ         ;TYPE OFFSET QUESTION
1993  016456  004437  017610                  JSR     R4,GETANS
1994  016462  023727  006336  000007  DONEPR: CMP     TPINUS,#7       ;TEST IF DRIVE TYPE IS
1995  016470  001405                          BEQ     4$              ;6 OR 7
1996  016472  023727  006336  000006          CMP     TPINUS,#6       ;IF NOT TYPE ERROR
1997  016500  001401                          BEQ     4$
1998  016502  000473                          BR      IVDTP
1999  016504  023737  006360  006356  4$:     CMP     ECINUS,SCINUS   ;COMP START & END CYL
2000  016512  002447                          BLT     RSTADD          ;NG-GET PARAM AGAIN
2001  016514  023737  006354  006352          CMP     ETINUS,STINUS   ;COMP START & END TRK
2002  016522  002443                          BLT     RSTADD          ;NG-GET PARAM AGAIN
2003  016524  023727  006360  000632          CMP     ECINUS,#632     ;TEST IF VALID CYL
2004  016532  003042                          BGT     IVDCYL          ;TOO BIG-BRANCH
2005  016534  105037  002653                  CLRB    TYPFMT          ;CLEAR TYPE-FORMAT SWITCH
```

```
2006  016540  023727  006336  000006          CMP     TPINUS,#6       ;TEST IF TRACK PARAMETERS ARE
2007  016546  001410                          BEQ     2$              ;VALID FOR THE DRIVE TYPE
2008  016550  152737  000004  002653          BISB    #B.CDT,TYPFMT
2009  016556  023727  006354  000004          CMP     ETINUS,#4       ;SPECIFIED.
2010  016564  003030                          BGT     IVDTRK
2011  016566  000404                          BR      3$
2012  016570  023727  006354  000002  2$:     CMP     ETINUS,#2
2013  016576  003023                          BGT     IVDTRK
2014  016600                          3$:
2015  016600  023727  006340  000007          CMP     DRINUS,#7       ;TEST IF VALID DRIVE
2016  016606  003026                          BGT     IVDDRV          ;TOO BIG-BRANCH
2017  016610  023727  006344  000004          CMP     MDINUS,#4       ;TEST IF VALID MODE
2018  016615  003030                          BGT     IVDMOD          ;ERROR IN MODE SELECTION
2019  016620  023727  006362  000160          CMP     OFINUS,#160     ;TEST IF VALID OFFSET
2020  016626  003033                          BGT     IVDOF           ;TO BIG, ERROR
2021  016630  000441                          BR      GETOUT          ;YES-BR TO EXIT
2022  016632  104401  013756          RSTADD: TYPE    ,INVDSE         ;TYPE ERROR MESSAGE
2023  016636  000405                          BR      SETRS
2024  016640  104401  014025          IVDCYL: TYPE    ,INVDCY         ;TYPE ERROR
2025  016644  000402                          BR      SETRS
2026  016646  104401  014055          IVDTRK: TYPE    ,INVDTK         ;TYPE ERROR
2027  016652  012702  006324          SETRS:  MOV     #INITST,R2      ;RESET REGISTERS TO REENTER ALL
2028  016656  012703  006352                  MOV     #STINUS,R3      ;START & END ADDRESSES
2029  016662  000663                          BR      PARAM5          ;GO GET NEW START & END ADD
2030  016664  104401  014102          IVDDRV: TYPE    ,INVDDR
2031  016670  000624                          BR      STPARM          ;GO GET ALL PARAM AGAIN
2032  016672  104401  014126          IVDTP:  TYPE    ,INVDTP
2033  016676  000621                          BR      STPARM
2034  016700  104401  014150          IVDMOD: TYPE    ,INVDMD
2035  016704  012702  006316                  MOV     #INITMD,R2      ;RESET REGISTERS FOR RE-REQUEST
2036  016710  012703  006344                  MOV     #MDINUS,R3      ;OF MODE
2037  016714  000632                          BR      PARAM3          ;RETURN TO MODE REQUEST
2038  016716  104401  014211          IVDOF:  TYPE    ,INVDOF         ;TYPE INVALID OFFSET MESSAGE
2039  016722  012702  006334                  MOV     #INITOF,R2      ;SET UP R2 AND R3 TO REDO OFFSET
2040  016726  012703  006362                  MOV     #OFINUS,R3      ;PARAMETER
2041  016732  000647                          BR      PARM6           ;GO GET OFFSET
2042  016734  105037  002656          GETOUT: CLRB    OPCONT          ;CLEAR OPERATION CONTROL SWITCHES
2043  016740  023727  006342  000026          CMP     SEINUS,#26      ;TEST IF 26 SECTOR FORMAT
2044  016746  001412                          BEQ     3$              ;YES-SKIP
2045  016750  012737  000024  006342          MOV     #24,SEINUS      ;FORCE TO 24 SECTORS/TRACK
2046  016756  152737  000020  002653          BISB    #B.CFMT,TYPFMT  ;SET BIT FOR FORMAT &
2047  016764  012737  012000  006306          MOV     #12000,TKWDCT   ;SET WORD COUNT
2048  016772  000403                          BR      4$
2049  016774  012737  013000  006306  3$:     MOV     #13000,TKWDCT   ;ELSE SET FOR 22 SECTOR WORD COUNT
2050  017002  005437  006306          4$:     NEG     TKWDCT          ;MAKE IT NEG FOR RK611
2051  017006  023737  006362  006334          CMP     OFINUS,INITOF   ;TEST IF OFFSET REQUIRED
2052  017014  001403                          BEQ     5$              ;NO-SKIP
2053  017016  152737  000020  002656          BISB    #OREQSW,OPCONT  ;ELSE SET SWITCH
2054  017024  023727  006344  000004  5$:     CMP     MDINUS,#4       ;IF MODE 4,
2055  017032  001004                          BNE     6$              ;NO-SKIP
2056  017034  152737  000012  002656          BISB    #RCDASW!VFHDSW,OPCONT ;ELSE SET READ CHECK & HEADER VERIFY
2057  017042  000433                          BR      10$
2058  017044  023727  006344  000003  6$:     CMP     MDINUS,#3       ;IF MODE 3
2059  017052  001004                          BNE     7$              ;NO-SKIP
2060  017054  152737  000002  002656          BISB    #VFHDSW,OPCONT  ;SET VERIFY HEADER
2061  017062  000423                          BR      10$
```

```
2062  017064  023727  006344  000002  7$:     CMP     MDINUS,#2       ;IF MODE 2
2063  017072  001004                          BNE     8$              ;NO-SKIP
2064  017074  152737  000007  002656          BISB    #WHDSW!VFHDSW!WCDASW,OPCONT ;SET WRITE HEADER & DATA VERIFY HEADER,
2065                                                                  ;AND WRITE CHECK DATA SWITCHES
2066  017102  000413                          BR      10$
2067  017104  023727  006344  000001  8$:     CMP     MDINUS,#1       ;IF MODE 1
2068  017112  001004                          BNE     9$              ;NO-SKIP
2069  017114  152737  000003  002656          BISB    #WHDSW!VFHDSW,OPCONT ;SET VERIFY HEADER &
2070                                                                  ;SET WRITE HEADER & DATA SWITCH
2071  017122  000403                          BR      10$             ;NO-SKIP
2072  017124  152737  000001  002656  9$:     BISB    #WHDSW,OPCONT   ;SET WRITE HDR & DATA SWITCH
2073  017132  000137  017136          10$:    JMP     GETUBS          ;GET USER BAD SECTORS
2074
2075                                  ;;*****************************************************************
2076                                  .SBTTL  USER BAD SECTORS INPUT ROUTINE
2077                                  ;*THIS ROUTINE PROVIDES THE CAPABILITY TO INDICATE SPECIFIC PACK
2078                                  ;*ADDRESSES THAT ARE TO BE FLAGED AS SOFTWARE BAD SECTORS. THE USER
2079                                  ;*IS DIRECTED IN HOW TO ENTER THE ADDRESSES AND HOW TO END THE
2080                                  ;*OPERATION.
2081                                  ;;*****************************************************************
2082  017136  132737  000001  002656  GETUBS: BITB    #WHDSW,OPCONT   ;TEST IF WRITING IS TO BE DONE
2083  017144  001566                          BEQ     14$             ;NO - SKIP OUT
2084  017146  012700  000400                  MOV     #400,R0         ;SET UP TO CLEAR BSSOFT FILE
2085  017152  012701  003660                  MOV     #BSSOFT,R1      ;TO ALL ONES
2086  017156  012721  177777          1$:     MOV     #177777,(R1)+
2087  017162  005300                          DEC     R0
2088  017164  001374                          BNE     1$
2089  017166  012701  002660                  MOV     #RDBUF,R1       ;GET ADDRESS OF READ BUFFER
2090  017172  012700  000400                  MOV     #400,R0         ;SET COUNT FOR CLEAR
2091  017176  012721  177777          21$:    MOV     #177777,(R1)+   ;CLEAR TO ALL ONES
2092  017202  005300                          DEC     R0              ;DECREMENT COUNT
2093  017204  001374                          BNE     21$             ;LOOP UNTIL 0
2094  017206  104401  013472                  TYPE    ,UBSQ           ;ASK USER BAD SECTOR QUESTION
2095  017212  104410                          RDLIN                   ;GET ANSWER
2096  017214  012601                          MOV     (SP)+,R1
2097  017216  121127  000032                  CMPB    (R1),#032       ;TEST IF ^Z
2098  017222  001002                          BNE     3$              ;NO - SKIP
2099  017224  000137  017776                  JMP     DRINIT          ;ELSE GO START PROGRAM
2100  017230  105711                  3$:     TSTB    (R1)            ;TEST IF CR (NULL)
2101  017232  001531                          BEQ     15$             ;YES - EXIT TEST
2102  017234  121127  000116                  CMPB    (R1),#'N        ;TEST IF "NO"
2103  017240  001526                          BEQ     15$             ;YES - EXIT TEST
2104  017242  012700  002660                  MOV     #RDBUF,R0       ;SET POINTER TO FIRST WORD
2105  017246  104401  013552          4$:     TYPE    ,UBSFMT         ;TYPE USER BAD SECTOR FORMAT
2106  017252  104401  013644                  TYPE    ,NEXTQ          ;TYPE NEXT ENTRY REQUEST
2107  017256  104410                  12$:    RDLIN                   ;GET BAD SECTOR ENTRY
2108  017260  012601                          MOV     (SP)+,R1
2109  017262  105711                          TSTB    (R1)            ;TEST IF CR (NULL)
2110  017264  001512                          BEQ     13$             ;YES - EXIT TEST
2111  017266  010146                          MOV     R1,-(SP)        ;SET TO CONVERT TO BINARY
2112  017270  004737  037426                  JSR     PC,OCTBIN
2113  017274  017474                          30$                     ;ERROR RETURN
2114  017276  021637  006360                  CMP     (SP),ECINUS     ;TEST IF CYLINDER VALID
2115  017302  003003                          BGT     16$             ;YES - SKIP
2116  017304  021637  006356                  CMP     (SP),SCINUS     ;TEST IF CYLINDER VALID
2117  017310  002004                          BGE     5$              ;YES - SKIP
```

```
2118  017312  104401  014025          16$:    TYPE    ,INVDCY         ;INVALID CYLINDER MESSAGE
2119  017316  005726                          TST     (SP)+           ;CLEAN STACK
2120  017320  000752                          BR      4$              ;GO TYPE FORMAT MESSAGE
2121  017322  012620                  5$:     MOV     (SP)+,(R0)+     ;STORE CYLINDER
2122  017324  121127  000054          6$:     CMPB    (R1),#',        ;LOOP TO BUMP R1 PAST CYLINDER ENTRY
2123  017330  001406                          BEQ     7$              ;LOOK FOR COMMA BUT RESTART
2124  017332  105721                          TSTB    (R1)+           ;THIS ENTRY IF CR (NULL) IS
2125  017334  001373                          BNE     6$              ;IS FOUND
2126  017336  005740                          TST     -(R0)           ;RESET POINTER
2127  017340  104401  013711                  TYPE    ,BADENT         ;TYPE BAD ENTRY MESSAGE
2128  017344  000740                          BR      4$              ;GO TYPE ENTRY FORMAT MESSAGE
2129  017346  105721                  7$:     TSTB    (R1)+           ;BUMP PAST COMMA
2130  017350  010146                          MOV     R1,-(SP)        ;GO CONVERT TRACK TO BINARY
2131  017352  004737  037426                  JSR     PC,OCTBIN
2132  017356  017474                          30$                     ;ERROR RETURN
2133  017360  012603                          MOV     (SP)+,R3        ;STORE TRACK
2134  017362  121127  000054          8$:     CMPB    (R1),#',        ;LOOP TO BUMP R1 PAST
2135  017366  001406                          BEQ     9$              ;TRACK ENTRY. LOOK FOR COMMA
2136  017370  105721                          TSTB    (R1)+           ;BUT RESTART THIS ENTRY IF CR
2137  017372  001373                          BNE     8$              ;(NULL) IS FOUND
2138  017374  005740                          TST     -(R0)           ;RESET POINTER
2139  017376  104401  013711                  TYPE    ,BADENT         ;TYPE BAD ENTRY MESSAGE
2140  017402  000721                          BR      4$              ; AND TYPE FORMAT MESSAGE
2141  017404  105721                  9$:     TSTB    (R1)+           ;BUMP PAST COMMA
2142  017406  010146                          MOV     R1,-(SP)        ;GO CONVERT SECTOR TO BINARY
2143  017410  004737  037426                  JSR     PC,OCTBIN
2144  017414  017474                          30$                     ;ERROR RETURN
2145  017416  012604                          MOV     (SP)+,R4        ;STORE SECTOR
2146  017420  020337  006354                  CMP     R3,ETINUS               ;TEST IF TRACK VALID
2147  017424  003003                          BGT     17$             ;YES - SKIP
2148  017426  020337  006352                  CMP     R3,STINUS       ;TEST IF TRACK VALID
2149  017432  002004                          BGE     10$             ;YES - SKIP
2150  017434  104401  014055          17$:    TYPE    ,INVDTK         ;TYPE INVALID TRACK MESSAGE
2151  017440  005740                          TST     -(R0)           ;RESET POINTER
2152  017442  000701                          BR      4$              ;RESTART ENTRY
2153  017444  020437  006342          10$:    CMP     R4,SEINUS       ;TEST IF SECTOR VALID
2154  017450  002404                          BLT     11$             ;YES - SKIP
2155  017452  104401  013731                  TYPE    ,ILLSEC         ;TYPE ILLEGAL SECTOR MESSAGE
2156  017456  005740                          TST     -(R0)           ;RESET POINTER
2157  017460  000672                          BR      4$              ;GO RESTART ENTRY
2158  017462  110420                  11$:    MOVB    R4,(R0)+        ;INSERT SECTOR NUMBER
2159  017464  110320                          MOVB    R3,(R0)+        ;INSERT TRACK NUMBER
2160  017466  104401  013644                  TYPE    ,NEXTQ          ;TYPE NEXT QUESTION
2161  017472  000671                          BR      12$             ;GO GET NEXT ENTRY
2162  017474  104401  006374          30$:    TYPE    ,QUES           ;TYPE BAD ENTRY LINE
2163  017500  104401  041674                  TYPE    ,$TTYIN
2164  017504  104401  001267                  TYPE    ,$CRLF
2165  017510  000656                          BR      4$              ;GET LINE AGAIN
2166  017512  012710  177777          13$:    MOV     #177777,(R0)    ;MAKE SURE THE WORD AFTER THE LAST
2167                                                                  ;VALID ENTRY IS ALL ONES.
2168  017516  000137  017526          15$:    JMP     CKPRES          ;GO CHECK IF PRESERVE SDBSF
2169  017522  000137  017776          14$:    JMP     DRINIT          ;GO INITIALIZE DRIVE
2170                                  ;;*********************** ************************************
2171                                  .SBTTL  PRESERVE SDBSF QUESTION ROUTINE
2172                                  ;*      THIS ROUTINE DETERMINES IF SOFTWARE BAD SECTOR FILES ARE TO
2173                                  ;*      BE PRESERVED
```

```
2174
2175  017526  104401  013305          C.PRES: TYPE    ,PSDQ           ;TYPE PRESERVE SDBSF QUESTION
2176  017532  104410                          RDLIN                   ;GET ANSWER
2177  017534  012601                          MOV     (SP)+,R1        ;GET POINTER TO ANSWER
2178  017536  104401  013372                  TYPE    ,SDBSFP         ;TYPE START OF PRESERVE MESSAGE
2179  017542  121127  000032                  CMPB    (R1),#032       ;TEST IF ↑Z
2180  017546  001001                          BNE     2$              ;NO - SKIP
2181  017550  000405                          BR      1$              ;ELSE SET BIT AND GO FORMAT
2182  017552  105711                  2$:     TSTB    (R1)            ;TEST IF NULL
2183  017554  001403                          BEQ     1$              ;YES - EXIT
2184  017556  121127  000116                  CMPB    (R1),#'N        ;TEST IF "N"
2185  017562  001406                          BEQ     3$              ;YES    EXIT
2186  017564  152737  000040  002656  1$:     BISB    #PSDBSF,OPCONT  ;SET FLAG TO PRESERVE SDBSF
2187  017572  104401  013436                  TYPE    ,PRSVD          ;REPORT PRESERVED
2188  017576  000402                          BR      4$              ;SKIP
2189  017600  104401  013452          3$:     TYPE    ,NPRSVD         ;REPORT NOT PRESERVED
2190  017604  000137  017776          4$:     JMP     DRINIT
2191
2192                                  ;;*****************************************************************
2193                                  .SBTTL  GET ANSWER (PARAMETER ENTRY) SUBROUTINE
2194                                  ;*ENTRY  JSR     R4,GETANS  WITH R2 POINTING TO THE DEFAULT PARAM
2195                                  ;*                          R3 POINTING TO THE ACTIVE PARAM
2196                                  ;*RETURN  RTS   R4
2197                                  ;*
2198                                  ;*THIS ROUTINE RECEIVES THE PARAMETER ENTRY AND ALTERS THE
2199                                  ;*APPROPRIATE ACTIVE PARAMETER.  IF THE ENTRY IS NULL THE DEFAULT
2200                                  ;*VALUE IS PLACED IN THE ACTIVE ENTRY.  THE CONTROL C & Z
2201                                  ;*(↑C & ↑Z) ARE DECODED AND THE APPROPRIATE ACTION TAKEN.  SOME
2202                                  ;*PARAMETER CHECKING IS DONE.  ALL NUMERICS ARE TESTED TO
2203                                  ;*INSURE THEY ARE OCTAL.
2204                                  ;;*****************************************************************
2205
2206  017610                          GETANS:
2207  017610  104410                          RDLIN                   ;READ ANSWER
2208  017612  012601                          MOV     (SP)+,R1        ;GET START ADD OF MESSAGE
2209  017614  121127  000032                  CMPB    (R1),#032       ;TEST FOR CONTROL Z
2210  017620  001434                          BEQ     11$             ;YES-BRANCH
2211  017622  105711                          TSTB    (R1)            ;TEST FOR NULL
2212  017624  001002                          BNE     4$
2213  017626  011213                          MOV     (R2),(R3)       ;GET DEFAULT FOR THIS PARAM
2214  017630  000405                          BR      5$
2215  017632  010146                  4$:     MOV     R1,-(SP)        ;PUT ADDRESS OF PARAM IN CN STACK
2216  017634  004737  037426                  JSR     PC,OCTBIN       ;CALL CONVERSION
2217  017640  017730                          30$                     ;ERROR RETURN
2218  017642  012613                          MOV     (SP)+,(R3)      ;STORE RESULTS
2219  017644  022703  006352          5$:     CMP     #STINUS,R3      ;START TRACK PARAM
2220  017650  001403                          BEQ     6$              ;YES-BRANCH
2221  017652  022703  006356                  CMP     #SCINUS,R3      ;STARTING CYL PARAM
2222  017656  001010                          BNE     9$              ;NO-GO TO EXIT
2223  017660  022223                  6$:     CMP     (R2)+,(R3)+     ;BUMP R2 & R3
2224  017662  105711                  7$:     TSTB    (R1)            ;CHECK IF NULL
2225  017664  001404                          BEQ     8$              ;YES-GO STORE DEFAULT & EXIT
2226  017666  122127  000054                  CMPB    (R1)+,#',       ;TEST IF COMMA
2227  017672  001373                          BNE     7$              ;NO-LOOP
2228  017674  000756                          BR      4$              ;YES-GO BACK TO CONVERT PARAM
2229  017676  011213                  8$:     MOV     (R2),(R3)       ;STORE DEFAULT
```

```
2230  017700  022223                  9$:     CMP     (R2)+,(R3)+     ;BUMP R2 & R3
2231  017702  000204                          RTS     R4              ;RETURN
2232  017704  012704  016342          10$:    MOV     #STPARM,R4      ;DUMMY R4 TO RESTART PARAM
2233  017710  000204                          RTS     R4              ;RETURN
2234  017712  012223                  11$:    MOV     (R2)+,(R3)+     ;MOV IN DEFAULT
2235  017714  020327  006364                  CMP     R3,#OPINUS+2    ;LAST PARAM SET?
2236  017720  001374                          BNE     11$             ;NO-LOOP
2237  017722  012704  016462                  MOV     #DONEPR,R4      ;DUMMY RETURN FOR PARAM COMPLETE
2238  017726  000204                          RTS     R4              ;RETURN
2239  017730  104401  006374          30$:    TYPE    ,QUES           ;TYPE QUESTION MARK
2240  017734  104401  041674                  TYPE    ,$TTYIN         ;TYPE BAD LINE
2241  017740  104401  001267                  TYPE    ,$CRLF
2242  017744  020327  006354                  CMP     R3,#ETINUS      ;IS THIS ENDING TRACK?
2243  017750  001403                          BEQ     31$             ;YES-BRANCH
2244  017752  020327  006360                  CMP     R3,#ECINUS      ;IS THIS ENDING CYLINDER?
2245  017756  001004                          BNE     32$             ;NO-BRANCH
2246  017760  162702  000002          31$:    SUB     #2,R2           ;BACK UP REGISTERS TO REDO
2247  017764  162703  000002                  SUB     #2,R3           ;PARAM REQUEST
2248  017770  162704  000010          32$:    SUB     #10,R4          ;BACK UP TO REREQUEST PARAM
2249  017774  000204                          RTS     R4              ;RETURN
2250
2251                                  ;;*****************************************************************
2252                                  .SBTTL  DRIVE INITIALIZE ROUTINE
2253                                  ;*BEFORE FORMATTING BEGINS, THIS ROUTINE INITIALIZES
2254                                  ;*THE SUBSYSTEM, RECALIBRATES THE DRIVE, AND SETS VOLUME
2255                                  ;*VALID.  IT THEN CHECKS DRIVE STATUS TO INSURE IT IS
2256                                  ;*AVAILABLE, READY, AND VALID.  WRITE LOCK IS CHECKED TO
2257                                  ;*VERIFY IT IS RESET.
2258                                  ;*
2259                                  ;*REGISTER 5 IS LOADED WITH THE ADDRESS OF THE
2260                                  ;*PARAMETER BLOCK.  R5 WILL CONTAIN THIS ADDRESS FOR
2261                                  ;*THE ENTIRE PROGRAM.
2262                                  ;*
2263                                  ;*THE DRIVE NUMBER PARAMETER IN THE PARAMETER
2264                                  ;*BLOCK IS SET TO THE SELECTED DRIVE.  THIS ALSO REMAINS
2265                                  ;*VALID FOR THE DURATION OF THE PROGRAM.
2266                                  ;;*****************************************************************
2267
2268  017776                          DRINIT:
2269  017776  012706  001100                  MOV     #STACK,SP       ;RESET THE STACK
2270  020002  012737  017776  001110          MOV     #DRINIT,$LPERR  ;SET UP TO LOOP ON ERROR
2271  020010  012737  026240  002600          MOV     #ERRHDL,A.ABNL  ;SET UP FOR ERROR HANDLING
2272  020016  012737  025302  002576          MOV     #ERRFRE,A.NORM  ;SET UP FOR NO ERROR DRIVER RETURNS
2273  020024  105037  002654                  CLRB    ERRCNT          ;CLEAR ERROR COUNT
2274  020030  005037  006364                  CLR     RECODE          ;CLEAR RECOVERY CODE
2275  020034  012703  002446                  MOV     #PARM1,R3       ;SET UP PARAMETER BLOCK 1 FOR
2276  020040  113763  006340  000000          MOVB    DRINUS,P.DRVN(R3) ;ADDITIONAL INFO RECOVERY IF
2277  020046  153763  002653  000007          BISB    TYPFMT,P.CS1(R3) ;NEEDED IN ERROR PROCESSING
2278  020054  013702  002570                  MOV     RKBAS,R2        ;SET R2 FOR RKBASE ADDRESS
2279  020060  012705  002362                  MOV     #PARM0,R5       ;GET PARAMETER BLOCK ADDRESS
2280  020064  005065  000014                  CLR     P.PRST(R5)              ;CLEAR PROGRAM STATUS REG
2281  020070  112765  000177  000001          MOVB    #SUBCLR,P.CMND(R5) ;DO SUBSYSTEM CLEAR
2282  020076  004737  025220                  JSR     PC,DRVCAL
2283  020102  113765  006340  000000          MOVB    DRINUS,P.DRVN(R5)       ;LOAD DRIVE NUMBER
2284  020110  112765  000113  000001          MOVB    #RECAL,P.CMND(R5) ;DO RECALIBRATE
2285  020116  004737  025220                  JSR     PC,DRVCAL
```

```
 2286  020122  112765  000103  000001          MOVB    #PACK,P.CMND(R5) ;DO PACK ACK
 2287  020130  004737  025220                  JSR     PC,DRVCAL
 2288  020134  112765  000141  000001          MOVB    #RDSTAT,P.CMND(R5) ;DO RD STAT
 2289  020142  004737  025220                  JSR     PC,DRVCAL
 2290  020146  023727  0C6344  000003          CMP     MDINUS,#3               ;TEST IF ANY WRITE TO BE DONE
 2291  020154  002415                          BLT     1$              ;NO-GO TEST WRITE LOCK ON
 2292  020156  032765  004000  000040          BIT     #S.WRL,P.ADD(R5) ;ELSE TEST IF WRITE LOCK OFF
 2293  020164  001026                          BNE     2$              ;IF WRONG - TYPE MESSAGE
 2294  020166  104401  014235                  TYPE    ,SWLREQ         ;AND WAIT FOR RESPONSE
 2295  020172  104401  014373                  TYPE    ,CONREQ
 2296  020176  042762  000100  000000          BIC     #IE,RKCS1(R2)   ;PREVENT INTERRUPT WHEN WRITE LOCK CHANGED
 2297  020204  000000                          HALT                    ;WRITE LOCK IS TESTED AGAIN
 2298  020206  000673                          BR      DRINIT          ;FOR CORRECT SETTING
 2299  020210  032765  004000  000040  1$:     BIT     #S.WRL,P.ADD(R5)
 2300  020216  001411                          BEQ     2$
 2301  020220  104401  014313                  TYPE    ,RWLREQ
 2302  020224  104401  014373                  TYPE    ,CONREQ
 2303  020230  042762  000100  000000          BIC     #IE,RKCS1(R2)   ;PREVENT INTERRUPT WHEN WRITE LOCK CHANGE
 2304  020236  000000                          HALT
 2305  020240  000656                          BR      DRINIT
 2306  020242  112765  000117  000001  2$:     MOVB    #SEEK,P.CMND(R5) ;DO SEEK
 2307  020250  012765  000632  000002          MOV     #632,P.CYLN(R5) ;TO CYL 410
 2308  020256  112765  000002  000005          MOVB    #2,P.TRCK(R5)   ;TRACK 2
 2309  020264  004737  025220                  JSR     PC,DRVCAL
 2310  020270  012765  004660  000010          MOV     #BSFACT,P.BALO(R5) ;SET BUFFER ADDRESS FOR BAD SECTORS
 2311  020276  005000                          CLR     R0
 2312  020300  012703  000011                  MOV     #11,R3          ;SET UP MAX SECTOR VALUE
 2313  020304  112765  000121  000001  8$:     MOVB    #RDDATA,P.CMND(R5) ;DO READ DATA FROM 0 THRU ?
 2314  020312  012765  000400  000012          MOV     #400,P.WC(R5)   ;READ ENTIRE SECTOR
 2315  020320  005465  000012                  NEG     P.WC(R5)
 2316  020324  142765  000020  000007          BICB    #B.CFMT,P.CS1H(R5) ;SET TO 22 SECTOR FORMAT
 2317  020332  132737  000020  0C3653          BITB    #B.CFMT,TYPFMT  ;TEST IF PACK TO BE FORMATTED 20 SECTOR
 2318  020340  001401                          BEQ     3$              ;NO-BR. LEAVE R0=0
 2319  020342  005200                          INC     R0              ;SET R0=1
 2320  020344  110065  000004          3$:     MOVB    R0,P.SECT(R5)   ;PUT IN BLOCK
 2321  020350  004737  025220                  JSR     PC,DRVCAL
 2322  020354  032737  100000  006364          BIT     #ANYDER,RECODE  ;TEST IF ERROR
 2323  020362  001410                          BEQ     4$              ;NO-GET A GOOD SECTOR READ
 2324  020364  062700  000002                  ADD     #2,R0           ;BUMP TO NEXT SECTOR
 2325  020370  020003                          CMP     R0,R3           ;CHECK IF STILL IN
 2326  020372  003764                          BLE     3$              ;BAD SECTOR. IF YES, DO IT AGAIN
 2327  020374  104401  014427                  TYPE    ,BAD632         ;TYPE MESSAGE
 2328  020400  000000                          HALT
 2329  020402  000776                          BR      .-2
 2330  020404  132737  000001  002656  4$:     BITB    #WHDSW,OPCONT   ;WILL THIS PASS WRITE?
 2331  020412  001404                          BEQ     17$             ;NO - SKIP
 2332  020414  132737  000040  002656          BITB    #PSDBSF,OPCONT  ;TEST IF PRESEPVE FILE FLAG SET
 2333  020422  001413                          BEQ     7$              ;NO SKIP
 2334  020424  020027  000011          17$:    CMP     R0,#11          ;TEST IF R0 GREATER THAN 11
 2335  020430  003010                          BGT     7$              ;IF YES, WE ALREADY HAVE A GOOD READ
 2336  020432  012765  003660  000010          MOV     #BSSOFT,P.BALO(R5)  ;ELSE SET UP TO READ SOFTWARE FILE
 2337  020440  012703  000025                  MOV     #25,R3          ;SET UP MAX SECTOR VALUE
 2338  020444  012700  000012                  MOV     #12,R0          ;SET UP STARTING SECTOR
 2339  020450  000715                          BR      8$              ;GO READ FILE
 2340  020452  104401  014723          7$:     TYPE    ,SERNUM         ;TYPE OUT SERIAL NUMBER
 2341  020456  012746  004660                  MOV     #BSFACT,-(SP)   ;OF CARTIDGE TO BE FORMATTED
```

```
2342  020462  004737  040002              JSR     PC,@#$DB20      ;CONVERT SERIAL NUMBER TO ASCII
2343  020466  012637  020474              MOV     (SP)+,6$        ;GET RESULT
2344  020472  104401                      TYPE                    ;PRINT IT
2345  020474  000000              6$:     .WORD                   ;ADDRESS OF RESULT GOES HERE
2346  020476  104401  015370              TYPE    ,STARS
2347  020502  005737  004666              TST     BSFACT+6        ;TEST IF 3RD WORD ALL ZEROS
2348  020506  001404                      BEQ     10$             ;IF YES - SKIP
2349  020510  104401  014763              TYPE    ,ALNPAC         ;ELSE TYPE ALIGNMENT PACK MESSAGE
2350                                                              ;AND ABORT
2351  020514  000137  016100              JMP     START           ;JUMP TO START IF CONTINUE
2352  020520  153765  002653  000007  10$:     BISB    TYPFMT,P.CSIH(R5) ;SET TYPE & FORMAT AGAIN
2353  020526  132737  000001  002656          BITB    #WHDSW,OPCONT   ;TEST IF WE WILL BE WRITING
2354  020534  001441                      BEQ     19$             ;NO - SKIP BSSOFT INIT
2355  020536  012700  003670              MOV     #BSSOFT+10,R0   ;ELSE SET POINTER TO BAD SECTOR DATA
2356  020542  012701  000200              MOV     #200,R1         ;SET A COUNT
2357  020546  132737  000040  002656          BITB    #PSDBSF,OPCONT  ;TEST IF PRESERVE FLAG SET
2358  020554  001413                      BEQ     25$             ;NO - SKIP
2359  020556  005720              20$:    TST     (R0)+           ;TEST IF ALL ONES
2360  020560  100410                      BMI     21$             ;YES - SKIP
2361  020562  005301                      DEC     R1              ;DECREMENT COUNT TO CHECK IT DOES NOT
2362  020564  001374                      BNE     20$             ;OVERFLOW THE BUFFER
2363  020566  104401  014466              TYPE    ,TDMSOF         ;MESSAGE CAUSED BY OVERFLOW
2364  020572  104401  014553              TYPE    ,TDMSUF
2365  020576  000137  016100              JMP     START           ;RESTART THE PROGRAM
2366  020602  005740              21$:    TST     -(R0)           ;REPOSITION POINTER
2367  020604  012703  002660      25$:    MOV     #RDBUF,R3       ;SET POINTER TO SECTORSENTERED AS BAD
2368  020610  012320              22$:    MOV     (R3)+,(R0)+     ;MOV IN THE ADDRESS
2369  020612  100412                      BMI     19$             ;IF NEGATIVE, MOVE IS DONE
2370  020614  012320                      MOV     (R3)+,(R0)+     ;ELSE GET SECOND WORD
2371  020616  024141                      CMP     -(R1),-(R1)     ;DEC COUNT BY 2
2372  020620  005701                      TST     R1              ;CHECK IT DIDN'T OVERFLOW
2373  020622  100372                      BPL     22$             ;NOT YET - LOOP
2374  020624  104401  014466              TYPE    ,TDMSOF         ;ELSE TYPE MESSAGE
2375  020630  104401  014553              TYPE    ,TDMSUF
2376  020634  000137  016100              JMP     START           ;RESTART TEST
2377  020640  012700  003660      19$:    MOV     #BSSOFT,R0      ;GET ADDRESS OF BSSOFT
2378  020644  013720  004660              MOV     BSFACT,(R0)+    ;GET FIRST WORD
2379  020650  013720  004662              MOV     BSFACT+2,(R0)+  ;GET SECOND WORD
2380  020654  005020                      CLR     (R0)+           ;CLEAR NEXT TWO WORDS
2381  020656  005020                      CLR     (R0)+           ;AS REQUIRED
2382  020660  012701  000200              MOV     #200,R1         ;SET COUNT
2383  020664  012700  004670              MOV     #BSFACT+10,R0   ;GET ADDR OF START OF BAD SECTORS
2384  020670  005720              24$:    TST     (R0)+           ;TEST IF NEGATIVE
2385  020672  100410                      BMI     23$             ;YES - SKIP
2386  020674  005301                      DEC     R1              ;ELSE DEC COUNT
2387  020676  100374                      BPL     24$             ;IF NOT YET ZERO - LOOP
2388  020700  104401  014637              TYPE    ,TDMFAC         ;MESSAGE REPORT
2389  020704  104401  014553              TYPE    ,TDMSUF
2390  020710  000137  016100              JMP     START           ;GO RESTART PROGRAM
2391  020714  000137  020720      23$:    JMP     DSPTCH          ;GO TO DISPATCHER
2392                                      ;;*****************************************************************
2393                                      .SBTTL  DISPATCH ROUTINE
2394                                      ;;*****************************************************************
2395  020720                      DSPTCH:
2396  020720  132737  000001  002656          BITB    #WHDSW,OPCONT   ;TEST IF HDRS & DATA TO BE WRITTEN
2397  020726  001404                      BEQ     1$
```

```
2398  020730  004737  021642                  JSR     PC,WRTOP        ;GO TO WRITE HEADERS & DATA
2399  020734  104401  015255                  TYPE    ,PRGRS1         ;TYPE PROGRESS 1 MESSAGE
2400
2401  020740  132737  000016  002656  1$:     BITB    #VFHDSW!WCDASW!RCDASW,OPCONT ;ANY OTHER OPERATION?
2402  020746  001030                          BNE     3$              ;YES-SKIP
2403  020750  104401  015322          4$:     TYPE    ,PRGRS2         ;TYPE DONE MESSAGE
2404  020754  132737  000001  002656          BITB    #WHDSW,OPCONT   ;TEST IF HEADERS AND DATA WERE WRITTEN
2405  020762  001402                          BEQ     5$              ;NO - DON'T WRITE BAD SECTOR FILES
2406  020764  004737  031452                  JSR     PC,BDSTWT       ;GO WRITE BAD SECTOR TABLES
2407  020770  032777  000200  160142  5$:     BIT     #SW7,@SWR       ;TEST IF BAD SECTOR REPORT REQUESTED
2408  020776  001402                          BEQ     2$
2409  021000  004737  031652                  JSR     PC,BSREPT       ;GO REPORT BAD SECTORS
2410  021004  012706  001100          2$:     MOV     #STACK,SP       ;RESET STACK
2411  021010  032777  000001  160122          BIT     #SW0,@SWR       ;TEST IF LOOP ON PROGRAM
2412  021016  001402                          BEQ     6$              ;NO - GO GET NEXT PARAMETERS
2413  021020  000137  017776                  JMP     DRINIT          ;ELSE RESTART OPERATION
2414  021024  000137  016342          6$:     JMP     STPARM          ;GO GET NEXT PARAM SET
2415  021030  004737  022642          3$:     JSR     PC,VEROP        ;GO TO VERIFY HEADERS AND/OR DATA
2416  021034  000745                          BR      4$
2417                                  ;;*****************************************************************
2418                                  .SBTTL  TRACK INCREMENT ROUTINE
2419                                  ;;*****************************************************************
2420  021036  032777  040000  160074  TRKINC: BIT     #SW14,@SWR      ;TEST IF LOOP ON OPERATION
2421  021044  001011                          BNE     2$              ;YES - BYPASS INCREMENT
2422  021046  020027  000632                  CMP     R0,#632         ;CHECK IF NEXT INCREMENT WOULD
2423  021052  001007                          BNE     1$              ;SELECT LAST TRACK ON THE
2424  021054  020127  000001                  CMP     R1,#1           ;PACK. IF SO-DO NOT
2425  021060  001004                          BNE     1$              ;INCREMENT
2426  021062  112737  000377  002651  3$:     MOVB    #377,OPCOMP     ;SET DONE
2427  021070  000207                  2$:     RTS     PC              ;& EXIT
2428  021072  005201                  1$:     INC     R1              ;ELSE INCREMENT TRACK
2429  021074  020137  006354                  CMP     R1,ETINUS       ;IF THIS BUMP DOES NOT
2430  021100  003773                          BLE     2$              ;EXCEED THE SELECTED PACK
2431  021102  013701  006352                  MOV     STINUS,R1       ;ADDRESS FORMAT-EXIT
2432  021106  005200                          INC     R0              ;IF IT DOES, SET THE TRACK TO
2433  021110  020037  006360                  CMP     R0,ECINUS       ;THE STARTING TRACK VALUES AND
2434  021114  003765                          BLE     2$              ;BUMP THE CYLINDER TEST THAT
2435  021116  000761                          BR      3$              ;THE NEW CYLINDER DOES NOT EXCEED
2436                                                                  ;THE DESIRED LIMIT. IF NOT SET DONE
2437                                                                  ;ELSE RETURN.
2438                                  ;;*****************************************************************
2439                                  .SBTTL  BUILD HEADERS ROUTINE
2440                                  ;;*****************************************************************
2441  021120                          BLDHDR:
2442  021120  104411                          SAVREG
2443  021122  012703  000026                  MOV     #26,R3          ;PRESET FOR 22 SECTOR FORMAT
2444  021126  012702  005674                  MOV     #BUFF0,R2       ;GET BUFFER ADDRESS
2445  021132  006301                          ASL     R1              ;SHIFT TO ALIGN FOR HEADER
2446  021134  006301                          ASL     R1
2447  021136  006301                          ASL     R1
2448  021140  006301                          ASL     R1
2449  021142  006301                          ASL     R1
2450  021144  052701  140000                  BIS     #140000,R1      ;SET GOOD SECTOR BITS
2451  021150  132737  000020  002653          BITB    #B.CFMT,TYPFMT  ;TEST IF 20 SET FORMAT
2452  021156  001404                          BEQ     1$              ;NO-SKIP
2453  021160  052701  001000                  BIS     #BIT9,R1        ;ELSE SET FORMAT BIT
```

```
2454  021164  012703  000024            MOV     #24,R3          ;ADJUST SECTOR COUNT
2455  021170  010022                1$: MOV     R0,(R2)+        ;INSERT CYLINDER WORD
2456  021172  010122                    MOV     R1,(R2)+        ;INSERT TRK/SECT WORD
2457  021174  010004                    MOV     R0,R4           ;COMPUTE VRC
2458  021176  010105                    MOV     R1,R5
2459  021200  040005                    BIC     R0,R5
2460  021202  040104                    BIC     R1,R4
2461  021204  050504                    BIS     R5,R4
2462  021206  010422                    MOV     R4,(R2)+        ;INSERT VRC
2463  021210  005201                    INC     R1              ;BUMP SECTOR COUNT
2464  021212  005303                    DEC     R3              ;DEC COUNT
2465  021214  001365                    BNE     1$              ;EXIT BUILD IF DONE
2466  021216  104412                    RESREG
2467  021220  012737  100000  001250    MOV     #BIT15,$TMP5    ;SET BIT TO BE BAD SECTOR INDICATOR
2468  021226  012737  004660  021240    MOV     #BSFACT,2$      ;SET TABLE TO BE SEARCHED
2469  021234  004437  021334        3$: JSR     R4,SRHTBS       ;GO SEARCH TABLE FOR BAD SECTORS
2470  021240  000000                2$: .WORD
2471  021242  012603                    MOV     (SP)+,R3        ;GET BAD SECTOR COUNT
2472  021244  001417                    BEQ     4$              ;IF NO BAD SECTORS GO TEST BSSOFT
2473  021246  011602                6$: MOV     (SP),R2         ;GET BAD SECTOR #
2474  021250  006302                    ASL     R2              ;MULTIPLY BAD SECTOR # BY 6
2475  021252  006302                    ASL     R2              ;AND ADD 2 TO GET INDEX
2476  021254  061602                    ADD     (SP),R2         ;TO TRK/SEC WORD OF HEADER OF
2477  021256  062602                    ADD     (SP)+,R2        ;BAD SECTOR
2478  021260  062702  000002            ADD     #2,R2
2479  021264  043762  001250  005674    BIC     $TMP5,BUFF0(R2) ;CLEAR BIT TO INDICATE BAD SECTOR
2480  021272  043762  001250  005676    BIC     $TMP5,BUFF0+2(R2) ;CLEAR BIT IN VRC FOR BAD SECTOR
2481  021300  005303                    DEC     R3              ;DEC # OF BAD SECTOR COUNTER
2482  021302  001361                    BNE     6$              ;IF 0
2483  021304  023727  021240  003660  4$: CMP     2$,#BSSOFT      ;TEST IF BAD SECTOR SOFTWARE
2484  021312  001407                    BEQ     5$              ;HAS BEEN TESTED. IF NOT, GO
2485  021314  012737  040000  001250    MOV     #BIT14,$TMP5
2486  021322  012737  003660  021240    MOV     #BSSOFT,2$      ;SET TABLE TO BE SEARCHED
2487  021330  000741                    BR      3$
2488  021332                        5$:                         ;ELSE EXIT
2489  021332  000207                    RTS     PC
2490                                ;;*****************************************************************
2491                                .SBTTL  SEARCH BAD SECTOR TABLES ROUTINE
2492                                ;*THIS ROUTINE RETURNS A LIST OF BAD SECTORS FOUND IN
2493                                ;*THE SPECIFIC TABLE (BAD SECTOR SOFTWARE OR BAD SECTOR FACTORY)
2494                                ;*FOR THE SPECIFIC CYLINDER AND TRACK DESIRED. R0 AND R1 MUST
2495                                ;*CONTAIN THE CYLINDER AND TRACK, RESPECTIVELY.
2496                                ;*
2497                                ;*THE LIST OF BAD SECTORS IS FOUND ON THE STACK WHEN THE ROUTINE
2498                                ;*RETURNS TO THE CALLER. THE FIRST ENTRY ON THE STACK WILL BE THE
2499                                ;*NUMBER OF BAD SECTORS FOUND FOR THIS CYLINDER AND TRACK.
2500                                ;;*****************************************************************
2501  021334  010237  001242    SRHTBS: MOV     R2,$TMP2
2502  021340  010337  001244            MOV     R3,$TMP3
2503  021344  012637  001246            MOV     (SP)+,$TMP4     ;STORE RETURN CONTENTS OF R4
2504  021350  011402                    MOV     (R4),R2 ;GET ADDRESS OF BAD SECTOR TABLE TO SEARCH
2505  021352  012437  001240            MOV     (R4)+,$TMP1     ;SETUP FOR LENGTH OF BAD SECTOR TABLE
2506  021356  062737  001000  001240    ADD     #1000,$TMP1
2507  021364  005003                    CLR     R3              ;CLEAR R3 FOR COUNT
2508  021366  062702  000010            ADD     #10,R2          ;SET R2 FOR POINTER TO CYLINDER ENTRY
2509  021372  005712                1$: TST     (R2)            ;TEST IF ALL ONES
```

```
2510  021374  100430                        BMI     5$                      ;YES-DONE
2511  021376  020012                        CMP     R0,(R2)                 ;TEST IF BAD SECTOR IN PRESENT CYL
2512  021400  001406                        BEQ     3$                      ;YES-GO CHECK TRACK
2513  021402  062702  000004                ADD     #4,R2                   ;ELSE BUMP POINTER
2514  021406  020237  001240                CMP     R2,$TMP1                ;TEST IF OUT OF TABLE
2515  021412  002021                        BGE     5$                      ;YES-EXIT
2516  021414  000766                        BR      1$                      ;LOOP
2517  021416  005722                3$:     TST     (R2)+                   ;TRACK CHECK - PUT POINTER AT TRK/SEC BYTE
2518  021420  011237  001256                MOV     (R2),$TMP10             ;GET TRK/SEC WORD
2519  021424  042737  174377  001256        BIC     #174377,$TMP10          ;CLEAR ALL BUT TRACK
2520  021432  123701  001257                CMPB    $TMP10+1,R1             ;CHECK IF BAD SECTOR IN THIS TRACK
2521  021436  001402                        BEQ     4$                      ;YES - GO PUT SECTOR NUMBER ON STACK
2522  021440  005722                        TST     (R2)+                   ;ELSE BUMP POINTER TO NEXT CYL WORD
2523  021442  000753                        BR      1$                      ;GO TEST NEXT CYL
2524  021444  005203                4$:     INC     R3                      ;BUMP BAD SECTOR COUNT
2525  021446  012246                        MOV     (R2)+,-(SP)             ;PUT TRK/SEC WORD ON STACK
2526  021450  042716  177700                BIC     #177700,(SP)            ;CLEAR ALL BUT SECTOR NUMBER
2527  021454  000746                        BR      1$                      ;GO CHECK REST OF FILE
2528  021456  010346                5$:     MOV     R3,-(SP)                ;EXIT - PUT NUMBER OF BAD
2529  021460  013702  001242                MOV     $TMP2,R2                ;SECTORS ON STACK-RESTORE
2530  021464  013703  001244                MOV     $TMP3,R3                ;REGISTERS
2531  021470  013746  001246                MOV     $TMP4,-(SP)             ;PUT RETURN ON STACK
2532  021474  000204                        RTS     R4                      ;RETURN
2533                                  ;;*****************************************************************
2534                                  .SBTTL  BAD SECTOR CHECK ROUTINE
2535                                  ;*THIS ROUTINE WILL SEARCH BOTH TABLES TO DETERMINE IF A
2536                                  ;*SPECIFIC SECTOR IS LISTED AS BAD. IF THE SECTOR IS LISTED IN THE
2537                                  ;*TABLES THE ROUTINE SETS THE "BADSEC" FLAG AND RETURNS. IF THE SECTOR
2538                                  ;*IS NOT LISTED THE FLAG IS RESET.
2539                                  ;;*****************************************************************
2540  021476  104411                BDSRCK: SAVREG
2541  021500  012737  004660  021512        MOV     #BSFACT,1$              ;SET TABLE TO SEARCH
2542  021506  004437  021334        2$:     JSR     R4,SRHTBS               ;GO SEARCH IT
2543  021512  000000                1$:     .WORD                           ;TABLE ADDRESS GOES HERE
2544  021514  012603                        MOV     (SP)+,R3                ;GET NUMBER OF BAD SECTORS, IF ANY
2545  021516  001015                        BNE     6$                      ;IF ANY, GO TEST WHICH ONES
2546  021520  023727  021512  003660  7$:     CMP     1$,#BSSOFT              ;ELSE TEST IF OTHER TABLE ALREADY SEARCHED
2547  021526  001404                        BEQ     3$                      ;IF YES-EXIT
2548  021530  012737  003660  021512        MOV     #BSSOFT,1$              ;SET OTHER TABLE FOR SEARCH
2549  021536  000763                        BR      2$                      ;GO SEARCH IT
2550  021540  042737  001000  006364  3$:     BIC     #BADSEC,RECODE          ;CLEAR BAD SECTOR BIT
2551  021546  104412                4$:     RESREG
2552  021550  000207                        RTS     PC                      ;RETURN
2553  021552  022602                6$:     CMP     (SP)+,R2                ;THIS SECTOR IN TABLE?
2554  021554  001403                        BEQ     8$                      ;YES-GO SET BIT & EXIT
2555  021556  005303                        DEC     R3                      ;DECREMENT BAD SECTOR COUNT
2556  021560  001374                        BNE     6$                      ;IF NOT ZERO-CHECK NEXT ENTRY
2557  021562  000756                        BR      7$                      ;ELSE GO SEARCH OTHER TABLE
2558  021564  052737  001000  006364  8$:     BIS     #BADSEC,RECODE          ;SET BAD SECTOR BIT
2559  021572  005303                9$:     DEC     R3                      ;CLEAR STACK OF OTHER BAD SECTOR
2560  021574  001764                        BEQ     4$                      ;NUMBER
2561  021576  005726                        TST     (SP)+
2562  021600  000774                        BR      9$                      ;EXIT
2563                                  ;;*****************************************************************
2564                                  .SBTTL  INSERT BAD SECTOR IN BAD SECTOR SOFTWARE TABLE ROUTINE
2565                                  ;;*****************************************************************
```

```
 2566  021602  104411                  BSSINS: SAVREG
 2567  021604  012704  003670                  MOV     #BSSOFT+10,R4   ;GET ADDRESS OF BAD SOFTWARE
 2568  021610  005724                  1$:     TST     (R4)+           ;SECTOR TABLE. FIND FIRST
 2569  021612  100376                          BPL     1$              ;EMPTY ENTRY
 2570  021614  005744                          TST     -(R4)           ;BACK UP POINTER
 2571  021616  010024                          MOV     R0,(R4)+        ;INSERT CYLINDER
 2572  021620  012703  000010                  MOV     #10,R3          ;SET COUNT FOR SHIFT TO ALIGN
 2573  021624  006301                  2$:     ASL     R1              ;SHIFT TRACK TO ALIGN FOR ENTRY
 2574  021626  005303                          DEC     R3              ;DEC SHIFT COUNT
 2575  021630  001375                          BNE     2$              ;TEST IF SHIFT DONE
 2576  021632  060201                          ADD     R2,R1           ;ADD IN SECTOR NUMBER
 2577  021634  010114                          MOV     R1,(R4)         ;INSERT TRACK/SECTOR
 2578  021636  104412                          RESREG
 2579  021640  000207                          RTS     PC
 2580                                  ;;*****************************************************************
 2581                                  .SBTTL  WRITE OPERATION CONTROL ROUTINE
 2582                                  ;;*****************************************************************
 2583  021642  105037  002651          WRTOP:  CLRB    OPCOMP          ;CLEAR DONE FLAG
 2584  021646  012737  021724  001110          MOV     #WERROR,$LPERR  ;SETUP FOR LOOP ON ERROR
 2585  021654  013700  006356                  MOV     SCINUS,R0       ;GET STARTING ADDRESS
 2586  021660  013701  006352                  MOV     STINUS,R1
 2587  021664  010037  006370          WRTLUP: MOV     R0,CCINUS       ;STORE CURRENT CYL
 2588  021670  010137  006366                  MOV     R1,CTINUS       ;STORE CURRENT TRACK
 2589  021674  032777  001000  157236          BIT     #SW9,@SWR       ;TEST IF LOOP ON ERROR
 2590  021702  001002                          BNE     1$              ;IF YES - DO NOT CLEAR ERROR FLAG
 2591  021704  105037  001103                  CLRB    $ERFLG          ;ELSE CLEAR FLAG
 2592  021710  004737  021120          1$:     JSR     PC,BLDHDR       ;GO BUILD HEADERS
 2593  021714  004737  021764                  JSR     PC,WRTHDR       ;GO WRITE HEADERS
 2594  021720  004737  022070                  JSR     PC,WRTTRK       ;GO WRITE DATA
 2595  021724  032777  001000  157206  WERROR: BIT     #SW9,@SWR       ;TEST IF LOOP ON ERROR
 2596  021732  001406                          BEQ     2$              ;IF NO - SKIP ERROR FLAG TEST
 2597  021734  105737  001103                  TSTB    $ERFLG          ;ELSE TEST ERROR FLAG
 2598  021740  001403                          BEQ     2$              ;NO - SKIP
 2599  021742  004737  031126                  JSR     PC,PREPLP       ;GO PREPARE FOR LOOP
 2600  021746  000746                          BR      WRTLUP          ;GO TO LOOP
 2601  021750  004737  021036          2$:     JSR     PC,TRKINC       ;BUMP TO NEXT ADDRESS
 2602  021754  105737  002651                  TSTB    OPCOMP          ;WRITE OPERATION DONE?
 2603  021760  001741                          BEQ     WRTLUP          ;NO-LOOP
 2604  021762  000207                          RTS     PC              ;ELSE RETURN TO DISPATCH
 2605                                  ;;*****************************************************************
 2606                                  .SBTTL  WRITE HEADERS ROUTINE
 2607                                  ;;*****************************************************************
 2608  021764                          WRTHDR:
 2609  021764  104411                          SAVREG
 2610  021766  005065  000014                  CLR     P.PRST(R5)      ;CLEAR PROG STATUS REGISTER
 2611  021772  112765  000127  000001          MOVB    #WRHEAD,P.CMND(R5) ;LOAD PARAMETER BLOCK WITH
 2612  022000  012765  005674  000010          MOV     #BUFFO,P.BALO(R5) ;VALUES TO DO THE WRITE HEADER.
 2613  022006  010065  000002                  MOV     R0,P.CYLN(R5)
 2614  022012  012765  000102  000012          MOV     #102,P.WC(R5)   ;LOAD THE WORD COUNT WITH
 2615  022020  132737  000020  002653          BITB    #B.CFMT,TYPFMT  ;THE PROPER VALUE FOR THE FORMAT
 2616  022026  001403                          BEQ     1$              ;TYPE
 2617  022030  012765  000074  000012          MOV     #74,P.WC(R5)
 2618  022036  005465  000012          1$:     NEG     P.WC(R5)
 2619  022042  110165  000005                  MOVB    R1,P.TRCK(R5)
 2620  022046  004737  025220                  JSR     PC,DRVCAL       ;CALL DRIVER
 2621  022052  032737  100000  006364          BIT     #ANYDER,RECODE  ;TEST FOR ANY DATA ERRORS
```

```
2622  022060  001401                        BEQ     2$
2623                                ;INSERT CODE FOR ANY DATA TYPE ERROR ON WRITE HEADERS
2624  022062  000000                        HALT
2625  022064  104412                  2$:   RESREG
2626  022066  000207                        RTS     PC                      ;RETURN
2627                                ;;*****************************************************************
2628                                .SBTTL  WRITE ONE TRACK ROUTINE
2629                                ;;*****************************************************************
2630  022070                        WRTTRK:
2631  022070  104411                        SAVREG
2632  022072  105037  002650        8$:     CLRB    DERCNT                  ;CLEAR ERROR COUNT
2633  022076  013765  006306  000012        MOV     TKWDCT,P.WC(R5) ;SET WORD COUNT FOR ONE TRACK
2634  022104  105065  000004              CLRB    P.SECT(R5)              ;SET SECTOR ZERO
2635  022110  012765  006346  000010  4$:   MOV     #WEINUS,P.BALO(R5) ;SET BUFFER ADDRESS TO ODD
2636  022116  032700  000001              BIT     #BIT0,R0                ;OR EVEN CYLINDER DATA DEPENDING
2637  022122  001403                      BEQ     1$                      ;ON THE CYLINDER NUMBER
2638  022124  012765  006350  000010      MOV     #WOINUS,P.BALO(R5)
2639  022132  010065  000002        1$:   MOV     R0,P.CYLN(R5)   ;SET UP THE PARAMETER BLOCK
2640  022136  110165  000005              MOVB    R1,P.TRCK(R5)   ;TO WRITE THE ENTIRE TRACK
2641  022142  112765  000123  000001      MOVB    #WRDATA,P.CMND(R5)
2642  022150  005065  000014              CLR     P.PRST(R5)
2643  022154  052765  100000  000014      BIS     #DTBAII,P.PRST(R5)
2644  022162  004737  025220              JSR     PC,DRVCAL               ;CALL DRIVER
2645  022166  032737  100000  006364      BIT     #ANYDER,RECODE  ;ANY DATA ERROR?
2646  022174  001002                      BNE     12$                     ;YES - PROCESS ERROR
2647  022176  000137  022636              JMP     7$                      ;NO-WRITE A-OK, EXIT
2648  022202  032737  000016  006364  12$:   BIT     #HVRCER!OPIERR!BSERR,RECODE ;VRC OR OPI ERROR
2649  022210  001002                      BNE     14$                     ;YES-SKIP
2650  022212  000137  022624              JMP     6$                      ;GO PRINT FUNNY MESSAGE
2651  022216  016537  000026  001252  14$:   MOV     P.DTS(R5),$TMP6 ;STORE TRK/SEC
2652  022224  042737  174377  001252      BIC     #174377,$TMP6   ;CLEAR ALL BUT TRACK
2653  022232  113701  001253              MOVB    $TMP6+1,R1      ;STORE TRACK
2654  022236  016537  000022  001252      MOV     P.WCR(R5),$TMP6 ;STORE WORD COUNT
2655  022244  042737  000377  001252      BIC     #377,$TMP6      ;CLEAR PARTIAL COUNT
2656  022252  062737  000400  001252      ADD     #400,$TMP6      ;BUMP COUNT TO NEXT SECTOR
2657  022260  016502  000026              MOV     P.DTS(R5),R2    ;& GET BAD SECTOR
2658  022264  042702  177740              BIC     #177740,R2
2659  022270  032737  000002  006364      BIT     #BSERR,RECODE   ;TEST IF BAD SECTOR
2660  022276  001076                      BNE     3$              ;GO SET UP TO WRITE REMAINDER IF TRACK
2661  022300  105237  002650        2$:   INCB    DERCNT                  ;BUMP ERROR COUNT
2662  022304  112765  000123  000001      MOVB    #WRDATA,P.CMND(R5) ;SET UP PARAM BLOCK
2663  022312  012765  177400  000012      MOV     #177400,P.WC(R5) ;TO WRITE ONE SECTOR
2664  022320  010065  000002              MOV     R0,P.CYLN(R5)   ;THIS IS THE RETRY
2665  022324  110165  000005              MOVB    R1,P.TRCK(R5)
2666  022330  110265  000004              MOVB    R2,P.SECT(R5)
2667  022334  052765  100000  000014      BIS     #DTBAII,P.PRST(R5)
2668  022342  004737  025220              JSR     PC,DRVCAL
2669  022346  032737  100000  006364      BIT     #ANYDER,RECODE  ;ERROR IN RETRY?
2670  022354  001441                      BEQ     5$              ;NO-GO PRINT RETRY SUCCESSFUL
2671  022356  122737  000007  002650      CMPB    #7,DERCNT               ;TEST IF ERROR COUNT TO BIG
2672  022364  002345                      BGE     2$                      ;IF NO - GO RETRY AGAIN
2673  022366  005037  001174              CLR     $REG5
2674  022372  113737  002650  001174      MOVB    DERCNT,$REG5
2675  022400  104102                      ERROR   102
2676  022402  004737  021476              JSR     PC,BDSRCK               ;GO TEST IF THIS SECTOR ALREADY MARKED BAD
2677  022406  032737  001000  006364      BIT     #BADSEC,RECODE  ;TEST RESULTS OF TEST
```

```
 2678  022414  001411                          BEQ     11$             ;NOT BAD BEFORE - SKIP
 2679  022416  010037  001174                  MOV     R0,$REG5        ;SET UP TO PRINT ERROR
 2680  022422  010137  001176                  MOV     R1,$REG6
 2681  022426  010237  001200                  MOV     R2,$REG7
 2682  022432  104107                          ERROR   107             ;ERROR- BAD HEADER AREA
 2683  022434  000137  016100                  JMP     START
 2684  022440  104076                  11$:    ERROR   76              ;REPORT MARKING SECTOR BAD
 2685  022442  004737  021602                  JSR     PC,BSSINS       ;INSERT BAD SECTOR IN BSSOFT
 2686  022446  004737  021120                  JSR     PC,BLDHDR       ;BUILD NEW HEADERS
 2687  022452  004737  021764                  JSR     PC,WRTHDR       ;WRITE NEW HEADERS
 2688  022456  000605                          BR      8$              ;RESTART WRITE TRACK
 2689  022460  005037  001174          5$:     CLR     $REG5           ;CLEAR REG 5 FOR COUNT
 2690  022464  113737  002650  001174          MOVB    DERCNT,$REG5    ;INSERT RETRY COUNT
 2691  022472  104101                          ERROR   101             ;PRINT RETRY SUCCESFUL
 2692  022474  005202                  3$:     INC     R2              ;THIS CODE SETS UP THE
 2693  022476  020237  006342                  CMP     R2,SEINUS       ;PARANETER BLOCK TO PICK
 2694  022502  002055                          BGE     7$              ;UP THE TRACK WRITE
 2695  022504  010265  000004                  MOV     R2,P.SECT(R5)   ;AFTER THE BAD SECTOR
 2696  022510  012704  000010                  MOV     #10,R4
 2697
 2698                                          ;THE FOLLOWING BLOCK OF CODE CHECKS THAT THE WORD COUNT IS
 2699                                          ;CORRECTLY ADJUSTED TO PICK UP THE OPERATION AFTER THE SECTOR
 2700                                          ;THAT IS CAUSING THE ERROR IS PROCESSED. IF THE WORD COUNT
 2701                                          ;SOMEHOW GETS SCREWED UP THE PROGRAM HALTS. CONTINUE CAN BE
 2702                                          ;DEPRESSED TO CAUSE THE WORD COUNT TO BE CORRECTED AND THE
 2703                                          ;PROGRAM ADJUSTS THE COUNT AND CONTINUES THE OPERATION.
 2704  022514  006302                  9$:     ASL     R2
 2705  022516  005304                          DEC     R4
 2706  022520  001375                          BNE     9$
 2707  022522  012704  013000                  MOV     #13000,R4
 2708  022526  023727  006342  000026          CMP     SEINUS,#26      ;ARE WE IN 26 SECTOR FORMAT?
 2709  022534  001402                          BEQ     13$             ;YES - SKIP
 2710  022536  012704  012000                  MOV     #12000,R4       ;ELSE CHANGE VALUE FOR 24 SECTOR FORMAT
 2711  022542  160204                  13$:    SUB     R2,R4
 2712  022544  005404                          NEG     R4
 2713  022546  016502  000004                  MOV     P.SECT(R5),R2
 2714  022552  020437  001252                  CMP     R4,$TMP6
 2715  022556  001413                          BEQ     10$
 2716  022560  010437  001204                  MOV     R4,$REG11
 2717  022564  013737  001252  001206          MOV     $TMP6,$REG12
 2718  022572  010237  001202                  MOV     R2,$REG10
 2719  022576  104105                          ERROR   105
 2720  022600  000000                          HALT
 2721  022602  010437  001252                  MOV     R4,$TMP6
 2722
 2723                                          ;END OF BLOCK OF CODE TO CHECK WORD COUNT
 2724
 2725  022606  013765  001252  000012  10$:     MOV     $TMP6,P.WC(R5)
 2726  022614  105037  002650                  CLRB    DERCNT
 2727  022620  000137  022110                  JMP     4$
 2728  022624  104401  015160          6$:     TYPE    ,IMPERR         ;TYPE IMPOSSIBLE ERROR
 2729  022630  000000                          HALT                    ;HALT
 2730  022632  000137  016100                  JMP     START
 2731  022636  104412                  7$:     RESREG
 2732  022640  000207                          RTS     PC
 2733                                  ;;*****************************************************************
```

```
                                        .SBTTL  VERIFY OPERATION CONTROL ROUTINE
                                        ;;*****************************************************************
022642                                  VEROP:
022642  012737  022756  001110                MOV     #VERROR,$LPERR  ;SET UP TO LOOP ON ERROR
022650  013700  006356                        MOV     SCINUS,R0       ;GET STARTING CYLINDER
022654  013701  006352                        MOV     STINUS,R1       ;AND TRACK
022660  105037  002651                        CLRB    OPCOMP
022664  010037  006370                VERLUP: MOV     R0,CCINUS       ;STORE CURRENT CYL
022670  010137  006366                        MOV     R1,CTINUS       ;STORE CURRENT TRACK
022674  032777  001000  156236                BIT     #SW9,@SWR       ;TEST IF LOOP ON ERROR
022702  001002                              BNE     1$              ;IF YES - DON'T CLEAR ERROR FLAG
022704  105037  001103                      CLRB    $ERFLG
022710  004737  023016                1$:   JSR     PC,RDHDRS
022714  004737  021120                      JSR     PC,BLDHDR
022720  004737  023354                      JSR     PC,HDRCMP
022724  132737  000010  002656              BITB    #RCDASW,OPCONT
022732  001403                              BEQ     2$
022734  004737  024516                      JSR     PC,RDOP         ;GO DO READ FOR DATA VERIFICATION
022740  000406                              BR      VERROR
022742  132737  000004  002656        2$:   BITB    #WCDASW,OPCONT  ;CHECK IF WRITE CHECK DATA
022750  001402                              BEQ     VERROR          ;IF NO - SKIP WRITE CHECK
022752  004737  023472                      JSR     PC,WCOP         ;ELSE GO DO WRITE CHECK FOR DATA VERIF.
022756  032777  001000  156154  VERROR:     BIT     #SW9,@SWR       ;TEST IF LOOP ON ERROR
022764  001406                              BEQ     4$              ;IF NO - GO DO NEXT TRACK
022766  105737  001103                      TSTB    $ERFLG          ;ELSE CHECK ERROR FLAG
022772  001403                              BEQ     4$              ;NO SKIP
022774  004737  031126                      JSR     PC,PREPLP       ;GO PREPARE FOR LOOPING
023000  000731                              BR      VERLUP          ;LOOP
023002  004737  021036                4$:   JSR     PC,TRKINC
023006  105737  002651                      TSTB    OPCOMP
023012  001724                              BEQ     VERLUP
023014  000207                              RTS     PC
                                        ;;*****************************************************************
                                        .SBTTL  READ HEADERS ROUTINE
                                        ;*THE HEADERS FOR A GIVEN CYLINDER AND TRACK (SPECIFIED IN R0 AND R1,
                                        ;*RESPECTIVELY) ARE READ IN AND SORTED INTO THE REQUIRED SEQUENCE.
                                        ;;*****************************************************************
023016                                  RDHDRS:
023016  104411                              SAVREG
023020  004737  031040                      JSR     PC,POSOFF           ;GO POSITION AND OFFSET
023024  012765  002660  000010              MOV     #RDBUF,P.BALO(R5)   ;LOAD ADDRESS FOR HEADERS
023032  005065  000014                      CLR     P.PRST(R5)          ;CLEAR PROGRAM STATUS
023036  010065  000002                2$:   MOV     R0,P.CYLN(R5)       ;SET UP TO READ HEADER
023042  110165  000005                      MOVB    R1,P.TRCK(R5)       ;OF DESIRED TRACK & CYLINDER
023046  005077  156072                      CLR     @$TKS               ;TURN OFF INTERRUPTS FROM KEYBOARD
023052  112765  000164  000001              MOVB    #RDALHD,P.CMND(R5)
023060  004737  025220                      JSR     PC,DRVCAL           ;DO IT
023064  012777  000100  156052              MOV     #100,@$TKS          ;TURN ON KEYBOARD
023072  012737  000204  001236              MOV     #204,$TMP0          ;SET UP TEMP STORE FOR THE NUMBER OF
023100  023727  006342  000026              CMP     SEINUS,#26          ;HEADER WORDS THAT MUST BE STORED FOR
023106  001403                              BEQ     9$                  ;THE FORMAT IN USE.
023110  012737  000170  001236              MOV     #170,$TMP0
023116  005003                        9$:   CLR     R3
023120  026300  002660                4$:   CMP     RDBUF(R3),R0        ;TEST IF FIRST HEADER WORD IS RIGHT CYL
023124  001027                              BNE     3$                  ;NO - GO BUMP TO NEXT
023126  016304  002662                      MOV     RDBUF+2(R3),R4      ;GET TRACK/SECTOR WORD
```

```
2790  023132  042704  177437              BIC     #177437,R4      ;CLEAR ALL BUT TRACK BITS
2791  023136  006204                      ASR     R4              ;SHIFT TRACK TO ALIGN FOR TESTING
2792  023140  006204                      ASR     R4
2793  023142  006204                      ASR     R4
2794  023144  006204                      ASR     R4
2795  023146  006204                      ASR     R4
2796  023150  020401                      CMP     R4,R1           ;CHECK IF TRACK IS CORRECT
2797  023152  001014                      BNE     3$              ;NO - GO BUMP TO NEXT HEADER
2798  023154  016304  002662              MOV     RDBUF+2(R3),R4  ;GET SECOND WORD
2799  023160  016302  002660              MOV     RDBUF(R3),R2    ;GET FIRST WORD
2800  023164  046304  002660              BIC     RDBUF(R3),R4    ;COMPUTE VRC OF HEADER
2801  023170  046302  002662              BIC     RDBUF+2(R3),R2
2802  023174  050402                      BIS     R4,R2
2803  023176  026302  002664              CMP     RDBUF+4(R3),R2  ;CHECK IF VRC IS CORRECT
2804  023202  001414                      BEQ     5$              ;YES - SKIP BUMP TO NEXT HEADER
2805  023204  062703  000006      3$:     ADD     #6,R3           ;BUMP TO NEXT HEADER
2806  023210  020337  001236              CMP     R3,$TMPO        ;TEST IF END OF HEADERS REACHED
2807  023214  003741                      BLE     4$              ;NO - GO TEST NEXT HEADER
2808  023216  010037  001202              MOV     R0,$REG10       ;SET UP CYLINDER AND TRACK FOR REPORT
2809  023222  010137  001204              MOV     R1,$REG11
2810  023226  104103                      ERROR   103             ;REPORT ERROR
2811  023230  000137  016100              JMP     START
2812  023234  016304  002662      5$:     MOV     RDBUF+2(R3),R4  ;GET TRACK/SECTOR AGAIN
2813  023240  042704  177740              BIC     #177740,R4      ;CLEAR ALL BUT SECTOR
2814  023244  020437  006342              CMP     R4,SEINUS       ;TEST IF A VALID SECTOR NUMBER
2815  023250  002355                      BGE     3$              ;NO - GO BUMP TO NEXT SECTOR
2816  023252  013702  006342              MOV     SEINUS,R2       ;INITIALIZE COUNTER FOR MOV
2817  023256  010401                      MOV     R4,R1           ;SET THE INDEX VALUE BY MULT SECTOR READ BY 6
2818  023260  006304                      ASL     R4
2819  023262  006304                      ASL     R4
2820  023264  060104                      ADD     R1,R4
2821  023266  060104                      ADD     R1,R4
2822  023270  016364  002660  006100  6$: MOV     RDBUF(R3),BUFF1(R4)     ;MOV WORD ONE
2823  023276  016364  002662  006102      MOV     RDBUF+2(R3),BUFF1+2(R4) ;MOV WORD TWO
2824  023304  016364  002664  006104      MOV     RDBUF+4(R3),BUFF1+4(R4) ;MOV WORD THREE
2825  023312  005302                      DEC     R2              ;DECREMENT COUNT
2826  023314  001415                      BEQ     8$              ;COUNT ZERO MOVE DONE, EXIT
2827  023316  062704  000006              ADD     #6,R4           ;BUMP INDEX POINTERS TO NEXT HEADER
2828  023322  062703  000006              ADD     #6,R3
2829  023326  020337  001236              CMP     R3,$TMPO        ;TEST IF END OF RDBUF REACHED
2830  023332  002401                      BLT     7$              ;NO - SKIP RESET TO START OF BUFFER
2831  023334  005003                      CLR     R3              ;ELSE RESET R3
2832  023336  020437  001236      7$:     CMP     R4,$TMPO        ;SAME TEST FOR BUFF1
2833  023342  002752                      BLT     6$              ;IF NOT END OF BUFFER, GO DO NEST MOVE
2834  023344  005004                      CLR     R4              ;ELSE RESET R4 TO START OF BUFFER
2835  023346  000750                      BR      6$              ;THEN GO DO NEXT MOVE
2836  023350  104412              8$:     RESREG
2837  023352  000207                      RTS     PC
2838                              ;;*****************************************************************
2839                              .SBTTL  HEADER COMPARE ROUTINE
2840                              ;;*****************************************************************
2841  023354                      HDRCMP:
2842  023354  104411                      SAVREG
2843  023356  005002                      CLR     R2
2844  023360  013704  006342              MOV     SEINUS,R4       ;GET NUM OF SECTORS
2845  023364  026262  005674  006100  1$: CMP     BUFF0(R2),BUFF1(R2) ;COMPARE HDR WORD 1
```

```
2846  023372  001015                        BNE     2$
2847  023374  026262  005676  006102        CMP     BUFF0+2(R2),BUFF1+2(R2) ;COMPARE HDR WORD 2
2848  023402  001011                        BNE     2$
2849  023404  026262  005700  006104        CMP     BUFF0+4(R2),BUFF1+4(R2) ;COMPARE HDR WORD 3
2850  023412  001005                        BNE     2$
2851  023414  062702  000006          3$:   ADD     #6,R2                   ;BUMP INDEX
2852  023420  005304                        DEC     R4                      ;DEC COUNT
2853  023422  001360                        BNE     1$                      ;LOOP UNTIL ZERO
2854  023424  000420                        BR      4$                      ;EXIT
2855  023426  012703  001174          2$:   MOV     #$REG5,R3               ;SET UP FOR ERROR REPORT
2856  023432  016223  005674                MOV     BUFF0(R2),(R3)+
2857  023436  016223  005676                MOV     BUFF0+2(R2),(R3)+
2858  023442  016223  005700                MOV     BUFF0+4(R2),(R3)+
2859  023446  016223  006100                MOV     BUFF1(R2),(R3)+
2860  023452  016223  006102                MOV     BUFF1+2(R2),(R3)+
2861  023456  016223  006104                MOV     BUFF1+4(R2),(R3)+
2862  023462  104062                        ERROR   62                      ;REPORT ERROR
2863  023464  000753                        BR      3$                      ;GO TEST NEXT HEADER
2864  023466  104412                  4$:   RESREG                          ;EXIT
2865  023470  000207                        RTS     PC
2866                                  ;;*****************************************************************
2867                                  .SBTTL  DATA VERIFY WITH WRITE CHECK OPERATION ROUTINE
2868                                  ;;*****************************************************************
2869  023472                          WCOP:
2870  023472  104411                        SAVREG
2871  023474  013765  006306  000012  12$:   MOV     TKWDCT,P.WC(R5) ;SET WORD COUNT FOR 1 TRACK
2872  023502  005037  002650                CLR     DERCNT          ;CLEAR ERROR COUNT
2873  023506  105065  000004                CLRB    P.SECT(R5)      ;SET TO SECTOR 0
2874  023512  004737  031040          5$:   JSR     PC,POSOFF       ;GO POSITION AND OFFSET
2875  023516  012765  006346  000010        MOV     #WEINUS,P.BALO(R5) ;SET UP FOR EVEN OR ODD
2876  023524  032700  000001                BIT     #BIT0,R0        ;CYLINDER DATA
2877  023530  001403                        BEQ     1$
2878  023532  012765  006350  000010        MOV     #WOINUS,P.BALO(R5)
2879
2880  023540  010065  000002          1$:   MOV     R0,P.CYLN(R5)   ;SET UP PARAMETER BLOCK
2881  023544  110165  000005                MOVB    R1,P.TRCK(R5)   ;FOR 1 TRACK WRITE CHECK
2882  023550  005065  000014                CLR     P.PRST(R5)
2883  023554  052765  100000  000014        BIS     #DTBAII,P.PRST(R5)
2884  023562  112765  000131  000001        MOVB    #WRTCHK,P.CMND(R5)
2885  023570  004737  025220                JSR     PC,DRVCAL
2886  023574  032737  100000  006364        BIT     #ANYDER,RECODE  ;TEST IF ANY ERROR
2887  023602  001002                        BNE     16$
2888  023604  000137  024344                JMP     15$             ;GO TO EXIT
2889  023610  016503  000022          16$:  MOV     P.WCR(R5),R3    ;STORE WORD COUNT & SECTOR
2890  023614  016502  000026                MOV     P.DTS(R5),R2    ;NUMBER AT ERROR TIME
2891  023620  042702  177740                BIC     #177740,R2      ;CLEAR ALL BUT SECTOR NUMBER
2892  023624  042703  000377                BIC     #377,R3         ; MAKE THE WORD COUNT
2893                                                                ;MODULO 400
2894  023630  032737  000016  006364        BIT     #OPIERR!HVRCER!BSERR,RECODE ;TEST IF OPI OR HVRC
2895  023636  001013                        BNE     4$              ;IF YES-SKIP
2896  023640  032737  000020  006364        BIT     #DCKERR,RECODE  ;TEST IF DATA CHECK
2897  023646  001402                        BEQ     2$              ;IF NO, WD COUNT IS NOW CORRECT-SKIP
2898  023650  162703  000400                SUB     #400,R3         ;ELSE INCREASE WORD COUNT
2899  023654  005702                  2$:   TST     R2              ;TEST IF SECTOR NOW ZERO
2900  023656  001002                        BNE     3$              ;IF NO-GO DECREMENT IT
2901  023660  013702  006342                MOV     SEINUS,R2       ;ELSE GET MAX SECTOR +1 VALUE
```

```
 2902  023664  005302                    3$:     DEC     R2                      ;NOW DECREMENT IT
 2903  023666  032737  000002  006364    4$:     BIT     #BSERR,RECODE           ;TEST IF BAD SECTOR
 2904  023674  001453                            BEQ     6$                      ;IF YES - GO CONTINUE TRACK
 2905  023676  005202                    8$:     INC     R2                      ;ELSE BUMP TO NEXT SECTOR
 2906  023700  020237  006342                    CMP     R2,SEINUS               ;IF NOT AT LAST SECTOR -
 2907  023704  002402                            BLT     19$                     ;GO TO EXIT IF LAST SECTOR IN ERROR
 2908  023706  000137  024344                    JMP     15$
 2909  023712  105037  002650            19$:    CLRB    DERCNT                  ;CLEAR ERROR COUNT
 2910  023716  010265  000004                    MOV     R2,P.SECT(R5)           ;SET UP FOR CONTINUE OF WC
 2911  023722  062703  000400                    ADD     #400,R3                 ;BUMP COUNT TO NEXT SECTOR
 2912  023726  012704  000010                    MOV     #10,R4
 2913
 2914                                            ;THE FOLLOWING BLOCK OF CODE CHECKS THAT THE WORD COUNT IS
 2915                                            ;CORRECTLY ADJUSTED TO PICK UP THE OPERATION AFTER THE SECTOR
 2916                                            ;THAT IS CAUSING THE ERROR IS PROCESSED. IF THE WORD COUNT
 2917                                            ;SOMEHOW GETS SCREWED UP THE PROGRAM HALTS. CONTINUE CAN BE
 2918                                            ;DEPRESSED TO CAUSE THE WORD COUNT TO BE CORRECTED AND THE
 2919                                            ;PROGRAM ADJUSTS THE COUNT AND CONTINUES THE OPERATION.
 2920  023732  006302                    17$:    ASL     R2
 2921  023734  005304                            DEC     R4
 2922  023736  001375                            BNE     17$
 2923  023740  012704  013000                    MOV     #13000,R4
 2924  023744  023727  006342  000026          CMP     SEINUS,#26              ;TEST IF WE ARE IN 26 SECTOR FORMAT
 2925  023752  001402                            BEQ     20$                     ;YES - SKIP
 2926  023754  012704  012000                    MOV     #12000,R4               ;ELSE SET COUNT FOR 24 SECTOR FORMAT
 2927  023760  160204                    20$:    SUB     R2,R4
 2928  023762  005404                            NEG     R4
 2929  023764  016502  000004                    MOV     P.SECT(R5),R2
 2930  023770  020403                            CMP     R4,R3
 2931  023772  001411                            BEQ     18$
 2932  023774  010437  001204                    MOV     R4,$REG11
 2933  024000  010337  001206                    MOV     R3,$REG12
 2934  024004  010237  001202                    MOV     R2,$REG10
 2935  024010  104105                            ERROR   10$
 2936  024012  000000                            HALT
 2937  024014  010403                            MOV     R4,R3
 2938
 2939                                            ;END OF CODE TO CHECK WORD COUNT
 2940
 2941  024016  010365  000012            18$:    MOV     R3,P.WC(R5)             ;AT NEXT SECTOR
 2942  024022  000633                            BR      5$                      ;GO DO IT
 2943  024024  105237  002650            6$:     INCB    DERCNT                  ;RETRY SEQUENCE-
 2944  024030  004737  031040                    JSR     PC,POSOFF               ;GO POSITION AND OFFSET
 2945  024034  112765  000131  000001          MOVB    #WRTCHK,P.CMND(R5) ;COUNT THE ERROR
 2946  024042  010065  000002                    MOV     R0,P.CYLN(R5)           ;SET UP PARAM BLOCK
 2947  024046  110165  000005                    MOVB    R1,P.TRCK(R5)
 2948  024052  110265  000004                    MOVB    R2,P.SECT(R5)
 2949  024056  012765  177400  000012          MOV     #177400,P.WC(R5)
 2950  024064  052765  100000  000014          BIS     #DTBAII,P.PRST(R5)
 2951  024072  004737  025220                    JSR     PC,DRVCAL
 2952  024076  032737  100000  006364          BIT     #ANYDER,RECODE          ;TEST IF ANY ERROR
 2953  024104  001507                            BEQ     14$                     ;GO TYPE RETRY SUCCESSFUL IF NO
 2954  024106  022737  000007  002650          CMP     #7,DERCNT               ;ELSE TEST IF ENOUGH RETRIES
 2955  024114  002343                            BGE     6$                      ;NO-LOOP
 2956  024116  032737  000100  006364          BIT     #WCERR,RECODE           ;TEST IF WCE, IF
 2957  024124  001446                            BEQ     10$                     ;NO GO CHECK FOR DATA CHECK
```

```
2958  024126  004737  031040                    JSR     PC,POSOFF       ;GO POSITION AND OFFSET
2959  024132  112765  000121  000001            MOVB    #RDDATA,P.CMND(R5) ;ELSE DO A READ OF SECTOR
2960  024140  012765  177400  000012            MOV     #177400,P.WC(R5) ;IN ERROR
2961  024146  010065  000002                    MOV     R0,P.CYLN(R5)
2962  024152  110165  000005                    MOVB    R1,P.TRCK(R5)
2963  024156  110265  000004                    MOVB    R2,P.SECT(R5)
2964  024162  005065  000014                    CLR     P.PRST(R5)      ;CLEAR PROGRAM STATUS WORD (BAII)
2965  024166  012765  002660  000010            MOV     #RDBUF,P.BALO(R5)
2966  024174  004737  025220                    JSR     PC,DRVCAL
2967  024200  032737  100000  006364            BIT     #ANYDER,RECODE  ;IF NO ERROR ON READ
2968  024206  001442                            BEQ     13$
2969  024210  032737  000020  006364            BIT     #DCKERR,RECODE  ;OR NO DATA CHECK
2970  024216  001436                            BEQ     13$             ;GO PRINT FUNNY MESSAGE
2971  024220  004737  024350                    JSR     PC,DACOMP       ;ELSE COMPARE DATA & PRINT ERRORS
2972  024224  032737  000040  006364  9$:       BIT     #ECCNC,RECODE   ;CHECK IF NON-CORRECTABLE
2973  024232  001007                            BNE     11$             ;IF YES-GO TO BAD SECTOR PROCESSING
2974  024234  104401  015233                    TYPE    ,CORABL         ;ELSE TYPE CORRECTABLE MESSAGE
2975  024240  000616                            BR      8$              ;& GO WC REST OF TRACK
2976  024242  032737  000020  006364  10$:      BIT     #DCKERR,RECODE  ;THIS IS TEST IF DAK WITH NO WCE
2977  024250  001365                            BNE     9$              ;IF YES-GO TEST FOR CORRECTABLE
2978  024252  005037  001174          11$:      CLR     $REG5
2979  024256  113737  002650  001174            MOVB    DERCNT,$REG5
2980  024264  104102                            ERROR   102
2981  024266  104076                            ERROR   76              ;REPORT MARKING SECTOR BAD
2982  024270  004737  021602                    JSR     PC,BSSINS       ;PUT BAD SECTOR IN BAD SECTOR
2983  024274  004737  021120                    JSR     PC,BLDHDR       ;TABLE, COMPUTE NEW HEADERS,
2984  024300  004737  021764                    JSR     PC,WRTHDR       ;WRITE THEM, REWRITE THE
2985  024304  004737  022070                    JSR     PC,WRTTRK       ;TRACK
2986  024310  000137  023474                    JMP     12$             ;& REDO THE WRITE CHECK
2987  024314  104401  015066          13$:      TYPE    ,IMPER2         ;TYPE FUNNY MESSAGE
2988  024320  000137  023474                    JMP     12$             ;GO RESTART THE WRITE CHECK
2989  024324  005037  001174          14$:      CLR     $REG5
2990  024330  113737  002650  001174            MOVB    DERCNT,$REG5    ;GET RETRY COUNT
2991  024336  104101                            ERROR   101             ;RETRY SUCCESFUL MESSAGE
2992  024340  000137  023676                    JMP     8$
2993  024344  104412                  15$:      RESREG                  ;EXIT
2994  024346  000207                            RTS     PC
2995                                ;;*****************************************************************
2996                                .SBTTL  DATA COMPARE ROUTINE
2997                                ;;*****************************************************************
2998  024350                        DACOMP:
2999  024350  104411                            SAVREG
3000  024352  005005                            CLR     R5              ;CLEAR R5 FOR ERROR COUNTING
3001  024354  012703  006346                    MOV     #WEINUS,R3      ;GET THE WORD THAT SHOULD BE
3002  024360  032700  000001                    BIT     #BIT0,R0        ;WRITTEN THROUGHOUT THE CYLINDER
3003  024364  001402                            BEQ     1$
3004  024366  012703  006350                    MOV     #WOINUS,R3
3005  024372  012704  002660          1$:       MOV     #RDBUF,R4       ;SET POINTER TO READ DATA
3006  024376  010037  001174                    MOV     R0,$REG5        ;STORE PACK ADDRESS FOR REPORT
3007  024402  010137  001176                    MOV     R1,$REG6
3008  024406  010237  001200                    MOV     R2,$REG7
3009  024412  012701  000001                    MOV     #1,R1           ;PRESET REGISTERS
3010  024416  012702  000001                    MOV     #1,R2           ;FOR COUNTING
3011  024422  012700  000400                    MOV     #400,R0
3012  024426  021324                  2$:       CMP     (R3),(R4)+      ;COMPARE DATA
3013  024430  001005                            BNE     3$              ;GO REPORT IF NOT EQUAL
```

```
3014  024432  005202                  5$:     INC     R2              ;ELSE BUMP COUNTERS
3015  024434  005300                          DEC     R0
3016  024436  001373                          BNE     2$              ;LOOP UNTIL DONE
3017  024440  104412                  6$:     RESREG
3018  024442  000207                          RTS     PC
3019  024444  005205                  3$:     INC     R5              ;BUMP ERROR COUNT
3020  024446  005301                          DEC     R1              ;BUMP R1 TO ZERO
3021  024450  011337  001202                  MOV     (R3),$REG10     ;STORE GOOD, BAD, & WORD NUM
3022  024454  014437  001204                  MOV     -(R4),$REG11
3023  024460  010237  001206                  MOV     R2,$REG12
3024  024464  005724                          TST     (R4)+           ;BUMP R4 TO NEXT DATA WORD
3025  024466  005701                          TST     R1              ;TEST R1 SWITCH
3026  024470  001001                          BNE     4$              ;IF NOT ZERO, THIS IS NOT FIRST PASS
3027  024472  104034                          ERROR   34              ;TYPE HEADING
3028  024474  032777  000002  154436  4$:     BIT     #SW1,@SWR       ;TEST IF NO LIMIT ON ERRORS
3029  024502  001003                          BNE     7$              ;SWITCH 1 OFF - LIMIT TO 10
3030  024504  020527  000010                  CMP     R5,#10  ;ELSE TEST IF 10 ERRORS REPORTED
3031  024510  001753                          BEQ     6$              ;YES - EXIT
3032  024512  104063                  7$:     ERROR   63              ;TYPE DATA
3033  024514  000746                          BR      5$              ;GO CONTINUE COMPARE
3034                                  ;;*****************************************************************
3035                                  .SBTTL  DATA VERIFY WITH READ OPERATION ROUTINE
3036                                  ;;*****************************************************************
3037  024516                          RDOP:
3038  024516  104411                          SAVREG
3039  024520  105065  000004                  CLRB    P.SECT(R5)      ;SET UP TO READ A FULL TRACK
3040  024524  105037  002650                  CLRB    DERCNT          ;WITH BUS INCREMENT INHIBIT
3041  024530  013765  006306  000012          MOV     TKWDCT,P.WC(R5)
3042  024536  004737  031040          6$:     JSR     PC,POSOFF       ;GO POSITION AND OFFSET
3043  024542  010065  000002                  MOV     R0,P.CYLN(R5)
3044  024546  110165  000005                  MOVB    R1,P.TRCK(R5)
3045  024552  005065  000014                  CLR     P.PRST(R5)
3046  024556  052765  100000  000014          BIS     #DTBAII,P.PRST(R5)
3047  024564  112765  000121  000001          MOVB    #RDDATA,P.CMND(R5)
3048  024572  004737  025220                  JSR     PC,DF           ;DO IT
3049  024576  032737  100000  006364          BIT     #ANY            ECODE   ;TEST IF ANY DATA ERROR
3050  024604  001002                          BNE     11$             ;READ A-OK, EXIT
3051  024606  000137  025214                  JMP     8$
3052  024612  016537  000022  001254  11$:    MOV     P.WCR(R5),$TMP7 ;STORE WORD COUNT AT ERROR
3053  024620  016502  000026                  MOV     P.DTS(R5),R2    ;STORE SECTOR & TRACK
3054  024624  042702  177400                  BIC     #177400,R2      ;CLEAR ALL BUT SECTOR
3055  024630  062737  000400  001254          ADD     #400,$TMP7      ;BUMP STORED WORD COUNT BY 1 SECTOR
3056  024636  032737  000002  006364          BIT     #BSERR,RECODE   ;CHECK IF BAD SECTOR
3057  024644  001106                          BNE     5$              ;YES - GO CONTINUE READ
3058  024646  032737  000020  006364          BIT     #DCKERR,RECODE  ;CHECK IF DATA CHECK
3059  024654  001410                          BEQ     3$              ;IF YES, DECREMENT SECTOR COUNT BY
3060  024656  005702                          TST     R2              ;1 TO GET SECTOR IN ERROR BUT BE
3061  024660  001002                          BNE     1$              ;SURE SECTOR COUNT IS NOT ZERO
3062  024662  013702  006342                  MOV     SEINUS,R2
3063  024666  005302                  1$:     DEC     R2
3064  024670  162737  000400  001254          SUB     #400,$TMP7      ;IF DCK, ADD WORD COUNT BACK IN
3065  024676  105237  002650          3$:     INCB    DERCNT          ;BUMP ERROR COUNT
3066  024702  004737  031040                  JSR     PC,POSOFF       ;GO POSITION AND OFFSET
3067  024706  112765  000121  000001          MOVB    #RDDATA,P.CMND(R5) ;ELSE SET UP TO RETRY
3068  024714  010065  000002                  MOV     R0,P.CYLN(R5)
3069  024720  110165  000005                  MOVB    R1,P.TRCK(R5)
```

```
3070  024724  110265  000004                      MOVB    R2,P.SECT(R5)
3071  024730  012765  177400  000012              MOV     #177400,P.WC(R5)
3072  024736  012765  001240  000010              MOV     #$TMP1,P.BAL0(R5)
3073  024744  052765  100000  000014              BIS     #DTBAII,P.PRST(R5)
3074  024752  004737  025220                      JSR     PC,DRVCAL
3075  024756  032737  100000  006364              BIT     #ANYDER,RECODE  ;ANY DATA ERROR IN RETRY?
3076  024764  001504                              BEQ     7$              ;NO-GO TYPE RETRY SUCCESSFUL
3077  024766  122737  000007  002650              CMPB    #7,DERCNT       ;ELSE CHECK IF ERROR COUNT TO BIG
3078  024774  002340                              BGE     3$              ;IF YES-TYPE UNSUCCESSFUL
3079  024776  005037  001174                      CLR     $REG5
3080  025002  113737  002650  001174              MOVB    DERCNT,$REG5
3081  025010  104102                              ERROR   102
3082  025012  032737  000020  006364              BIT     #DCKERR,RECODE  ;TEST IF THIS IS A DATA ERROR
3083  025020  001420                              BEQ     5$              ;IF NO - SKIP ECC PRINTOUT
3084  025022  016537  000060  001174              MOV     P.EPOS(R5),$REG5  ;STORE ECC PATTERN AND POSITION
3085  025030  016537  000062  001176              MOV     P.EPAT(R5),$REG6  ;FOR ERROR REPORT
3086  025036  005037  001200                      CLR     $REG7           ;SET $REG7 FOR ECC UNCORRECTABLE
3087  025042  032737  000040  006364              BIT     #ECCNC,RECODE   ;WAS THAT CORRECT?
3088  025050  001003                              BNE     4$              ;YES - GO REPORT
3089  025052  052737  000001  001200              BIS     #1,$REG7        ;NO - SET FOR CORRECTABLE
3090  025060  104077                      4$:     ERROR   77              ;REPORT ERROR
3091  025062  005202                      5$:     INC     R2              ;SET UP TO CONTINUE ON NEXT SECTOR
3092  025064  020237  006342                      CMP     R2,SEINUS       ;CHECK IF ERROR SECTOR WAS LAST SECTOR
3093  025070  002051                              BGE     8$              ;EXIT IF YES
3094  025072  010265  000004                      MOV     R2,P.SECT(R5)   ;READ OF REST OF TRACK BY
3095  025076  012704  000010                      MOV     #10,R4
3096  025102  006302                      9$:     ASL     R2
3097  025104  005304                              DEC     R4
3098  025106  001375                              BNE     9$
3099  025110  012704  013000                      MOV     #13000,R4
3100  025114  160204                              SUB     R2,R4
3101  025116  005404                              NEG     R4
3102  025120  016502  000004                      MOV     P.SECT(R5),R2
3103  025124  020437  001254                      CMP     R4,$TMP7
3104  025130  001413                              BEQ     10$
3105  025132  010437  001204                      MOV     R4,$REG11
3106  025136  013737  001254  001206              MOV     $TMP7,$REG12
3107  025144  010237  001202                      MOV     R2,$REG10
3108  025150  104105                              ERROR   105
3109  025152  000000                              HALT
3110  025154  010437  001254                      MOV     R4,$TMP7
3111  025160  013765  001254  000012  10$:        MOV     $TMP7,P.WC(R5)  ;BUMPING TO THE NEXT SECTOR
3112  025166  105037  002650                      CLRB    DERCNT          ;INSERTING THE WORD COUNT &
3113  025172  000137  024536                      JMP     6$              ;GO START RD DATA
3114  025176  005037  001174              7$:     CLR     $REG5           ;CLEAR REG 5 FOR RETRY COUNT BYTE
3115  025202  113737  002650  001174              MOVB    DERCNT,$REG5    ;GET RETRY COUNT
3116  025210  104101                              ERROR   101             ;REPORT RETRY SUCCESSFUL
3117  025212  000723                              BR      5$
3118  025214  104412                      8$:     RESREG
3119  025216  000207                              RTS     PC
3120                                  ;;*****************************************************************
3121                                  .SBTTL  CALL DRIVER ROUTINE
3122                                  ;*ENTRY JSR PC,DRVCAL
3123                                  ;*              WITH R5 POINTING TO PARAMETER BLOCK
3124                                  ;*RETURN RTS PC
3125                                  ;*
```

```
                                        ;*THIS ROUTINE IS USED TO INITIATE A SUBSYSTEM OPERATION BY
                                        ;*CALLING THE DRIVER. THE PARAMETER BLOCK MUST BE SET UP
                                        ;*BY THE CALLER AND R5 MUST POINT TO THE PARAMETER
                                        ;*BLOCK WHEN THE ROUTINE IS CALLED.
                                        ;*
                                        ;*THIS ROUTINE WAITS FOR THE OPERATION TO BE COMPLETED.
                                        ;*WHILE WAITING THE WATCHDOG TIMER IS CALLED TO PREVENT
                                        ;*SILENT DEATH IN CASE THE SUBSYSTEM DOES NOT PROVIDE AN
                                        ;*INTERRUPT. THE TERMINATION HANDLER ROUTINES WILL SET THE DONE
                                        ;*FLAG WHICH KEYS THE ROUTINE TO RETURN TO THE CALLER.
                                        ;;*****************************************************************
025220  105037  002652          DRVCAL: CLRB    DONE            ;CLEAR DONE FLAG
025224  010146                          MOV     R1,-(SP)        ;STORE REGS
025226  010546                          MOV     R5,-(SP)
025230  012701  005660                  MOV     #COMSTR,R1      ;GET ADDRESS OF COMMAND STORAGE
025234  012521                          MOV     (R5)+,(R1)+     ;STORE PARAMETER VALUES
025236  012521                          MOV     (R5)+,(R1)+     ;IN COMMAND STORAGE
025240  012521                          MOV     (R5)+,(R1)+
025242  012521                          MOV     (R5)+,(R1)+
025244  012521                          MOV     (R5)+,(R1)+
025246  012521                          MOV     (R5)+,(R1)+
025250  012605                          MOV     (SP)+,R5
025252  012601                          MOV     (SP)+,R1
025254  010537  025264                  MOV     R5,1$           ;GET PARAM BLOCK ADDRESS
025260  004737  035532                  JSR     PC,C.INIT       ;CALL DRIVER
025264  000000          1$:     .WORD                           ;P.B. ADDRESS GOES HERE
025266  004737  032102          2$:     JSR     PC,W.WTCH       ;CALL WATCH DOG
025272  105737  002652                  TSTB    DONE            ;DONE SET?
025276  001773                          BEQ     2$              ;NO-LOOP
025300  000207                          RTS     PC              ;YES-RETURN
                                        ;;*****************************************************************
                                        .SBTTL  DRIVE ERROR FREE RETURN ROUTINE
                                        ;*THIS ROUTINE IS CALLED BY THE DRIVER WHEN NO ERROR
                                        ;*HAS BEEN DETECTED IN THE OPERATION. THE ROUTINE SETS THE
                                        ;*DONE FLAG THAT IS TESTED IN THE DRIVER CALLING
                                        ;*ROUTINE.
                                        ;;*****************************************************************
025302  152737  000377  002652  ERRFRE: BISB    #377,DONE       ;SET THE DONE FLAG
025310  032737  100000  006364          BIT     #ANYDER,RECODE  ;TEST IF ANY DATA ERROR
025316  001403                          BEQ     2$              ;IF NO - DO ERROR RECOVERY PRINT TEST
025320  105037  002654                  CLRB    ERRCNT          ;CLEAR ERROR COUNT
025324  000413                          BR      1$              ;EXIT
025326  105737  002654          2$:     TSTB    ERRCNT          ;CHECK IF ANY ERRORS HAVE OCCURRED
025332  001410                          BEQ     1$              ;NO - SKIP TO EXIT
025334  005037  001174                  CLR     $REGS
025340  113737  002654  001174          MOVB    ERRCNT,$REGS    ;GET RETRY COUNT
025346  104101                          ERROR   101             ;PRINT RETRY SUCCESSFUL MESSAGE
025350  105037  002654                  CLRB    ERRCNT          ;CLEAR ERROR COUNT
025354  005037  006364          1$:     CLR     RECODE          ;CLEAR RECOVERY FLAGS
025360  000207                          RTS     PC              ;RETURN
                                        ;;*****************************************************************
                                        .SBTTL  TYPE ERROR ROUTINE
                                        ;*ENTRY JSR PC,TYP ERR
                                        ;*RETURN RTS PC
                                        ;*
                                        ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
```

```
                                        ;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
                                        ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
                                        ;*THE ERROR.
                                        ;;*****************************************************************
025362  104411                  TYPERR: SAVREG
025364  032777  020000  153546          BIT     #SW13,@SWR      ;INHIBIT ERROR TYPEOUTS?
025372  001107                          BNE     20$             ;YES-BRANCH
025374  113700  001114                  MOVB    $ITEMB,R0       ;ENTER ERROR NUMBER
025400  042700  177400                  BIC     #177400,R0      ;CLEAR UPPER BITS
025404  005300                          DEC     R0              ;FORM INDEX FOR ERROR TABLE
025406  006300                          ASL     R0
025410  006300                          ASL     R0
025412  006300                          ASL     R0
025414  062700  001272          1$:     ADD     #$ERRTB,R0      ;FORM ADDRESS OF ERROR ENTRY
025420  012037  025434                  MOV     (R0)+,2$        ;GET EM POINTER
025424  001404                          BEQ     3$              ;BRANCH IF THERE ISN'T ONE
025426  104401  001267                  TYPE    ,$CRLF          ;TYPE CARRIAGE RETURN LINE FEED
025432  104401                          TYPE                    ;TYPE ERROR MESSAGE (EM)
025434  000000          2$:     .WORD   0                       ;EM POINTER GOES HERE
025436  012037  025452          3$:     MOV     (R0)+,4$        ;GET DH POINTER
025442  001404                          BEQ     5$              ;BRANCH IF THERE ISN'T ONE
025444  104401  001267                  TYPE    ,$CRLF          ;TYPE CR-LF
025450  104401                          TYPE                    ;TYPE DATA HEADER
025452  000000          4$:     .WORD   0                       ;DH POINTER GOES HERE
025454  012001          5$:     MOV     (R0)+,R1        ;GET DT POINTER
025456  001455                  BEQ     20$             ;BRANCH IF THERE ARE NONE
025460  005004                  CLR     R4              ;SET INDENT SWITCH
025462  012000                  MOV     (R0)+,R0        ;GET DF POINTER
025464  012002                  MOV     (R0)+,R2        ;STORE NUMBER OF DH'S
025466  001446                  BEQ     17$             ;DH NUM IS 0-BRANCH
025470  005104                  COM     R4              ;NO INDENT
025472  104401  001267                  TYPE    ,$CRLF
025476  112003          10$:    MOVB    (R0)+,R3        ;GET & STORE NUMBER OF DATA WORDS
025500  105720                  TSTB    (R0)+           ;BUMP PAST FORMAT WORD
025502  005703                  TST     R3              ;TEST IF ANY DATA FOR THIS HEADER
025504  001407                  BEQ     14$             ;NO - SKIP DATA PRINT
025506  013146          11$:    MOV     @(R1)+,-(SP)    ;PUT FIRST DATA WORD ON STACK
025510  104402                  TYPOC                   ;TYPE IT
025512  005303                  DEC     R3              ;MORE DATA WORDS
025514  001403                  BEQ     14$             ;NO-BRANCH
025516  104401  012744                  TYPE    ,$SPACE2        ;TYPE SEPARATORS
025522  000771                  BR      11$             ;LOOP
```

```
3224  025524  005302                  14$:    DEC     R2              ;MORE DH'S?
3225  025526  003431                          BLE     20$             ;NO-BRANCH
3226  025530  104401  001267                  TYPE    ,$CRLF
3227  025534  005760  000002                  TST     2(R0)           ;ONLY A DH IN THIS REQUEST?
3228  025540  001404                          BEQ     15$             ;YES-BRANCH BYPASS INDENT
3229  025542  005104                          COM     R4              ;INDENT?
3230  025544  001002                          BNE     15$             ;NO-BRANCH
3231  025546  104401  012744                  TYPE    ,SPACE2         ;YES-TYPE SPACES
3232  025552  012037  025560          15$:    MOV     (R0)+,16$       ;GET NEXT DH POINTER
3233  025556  104401                          TYPE                    ;TYPE DH
3234  025560  000000                  16$:    .WORD   0               ;DH POINTER GOES HERE
3235  025562  105710                          TSTB    (R0)            ;TYPE A DT?
3236  025564  001003                          BNE     21$             ;YES-BRANCH
3237  025566  062700  000002                  ADD     #2,R0           ;INCREMENT DF POINTER
3238  025572  000754                          BR      14$             ;SEE IF END OF DF BLOCK
3239  025574  104401  001267          21$:    TYPE    ,$CRLF
3240  025600  005704                          TST     R4              ;INDENT?
3241  025602  001335                          BNE     10$             ;NO-BRANCH
3242  025604  104401  012744          17$:    TYPE    ,SPACE2         ;YES-TYPE SPACES
3243  025610  000732                          BR      10$             ;LOOP
3244  025612  104412                  20$:    RESREG
3245  025614  000207                          RTS     PC
3246                                  ;;*****************************************************************
3247                                  .SBTTL  CONTROLLER ERROR REPORTER ROUTINE
3248                                  ;*ENTRY:        JSR PC, CONERR
3249                                  ;*RETURN:       RTS PC
3250                                  ;*
3251                                  ;*THIS ROUTINE DECODES THE CONTROLLER ERROR WORD AND
3252                                  ;*REPORTS THE APPROPRIATE MESSAGE. THE RK611 REGISTERS ARE
3253                                  ;*RETRIEVED FROM THE RK611 AND PLACED IN THE PARAMETER
3254                                  ;*BLOCK. THIS IS DONE BECAUSE PARM 0 MAY NOT BE VALID
3255                                  ;*AT THIS TIME.
3256                                  ;;*****************************************************************
3257  025616  104411                  CONERR: SAVREG
3258  025620  152737  000377  002652          BISB    #377,DONE       ;SET DONE FLAG
3259  025626  105237  002654                  INCB    ERRCNT          ;BUMP ERROR COUNTER
3260  025632  032765  001000  000014          BIT     #PBSVAL,P.PRST(R5) ;TEST IF STATUS VALID
3261  025640  001403                          BEQ     12$             ;YES - SKIP
3262  025642  004737  026060                  JSR     PC,REPSUP       ;CALL FOR GETTING STATUS
3263  025646  000402                          BR      13$
3264  025650  004737  030336          12$:    JSR     PC,TOPROC       ;LOAD RK REGS INTO $REGS
3265  025654  032737  000001  002604  13$:    BIT     #BIT0,E.CONT    ;ERROR 0?
3266  025662  001402                          BEQ     1$              ;NO-BRANCH
3267  025664  104064                          ERROR   64              ;CLEAR CONT DID NOT CLEAR ERROR
3268  025666  000463                          BR      7$
3269  025670  032737  000002  002604  1$:     BIT     #BIT1,E.CONT    ;ERROR 1?
3270  025676  001402                          BEQ     2$              ;NO-BRANCH
3271  025700  104065                          ERROR   65              ;NO ATTENTION IN ATTENTION SUM REG
3272  025702  000455                          BR      7$
3273  025704  032737  000004  002604  2$:     BIT     #BIT2,E.CONT    ;ERROR 2?
3274  025712  001402                          BEQ     3$              ;NO-BRANCH
3275  025714  104066                          ERROR   66              ;UNSOLICITED ATTENTION
3276  025716  000447                          BR      7$
```

```
3277  025720  032737  000010  002604  3$:     BIT     #BIT3,E.CONT    ;ERROR 3?
3278  025726  001402                          BEQ     4$              ;NO-BRANCH
3279  025730  104067                          ERROR   67              ;UNEXPECTED DATA TYPE ERROR
3280  025732  000441                          BR      7$
3281  025734  032737  000020  002604  4$:     BIT     #BIT4,E.CONT    ;ERROR 4?
3282  025742  001402                          BEQ     5$              ;NO-BRANCH
3283  025744  104070                          ERROR   70              ;ATTENTION DID NOT RESET WITH CLEAR
3284  025746  000433                          BR      7$
3285  025750  032737  000040  002604  5$:     BIT     #BIT5,E.CONT    ;ERROR 5?
3286  025756  001402                          BEQ     6$              ;NO-BRANCH
3287  025760  104071                          ERROR   71              ;SUBSYS CLEAR DIDN'T CLEAR DRIVE ATTENTION
3288  025762  000425                          BR      7$
3289  025764  032737  000400  002604  6$:     BIT     #BIT8,E.CONT
3290  025772  001401                          BEQ     8$
3291  025774  104072                          ERROR   72              ;DATA LATE WHEN UNLOADING HEADER
3292  025776  032737  001000  002604  8$:     BIT     #BIT9,E.CONT
3293  026004  001401                          BEQ     9$
3294  026006  104073                          ERROR   73              ;CONTROLLER ERROR DURING DRIVER SERVICE
3295  026010  032737  002000  002604  9$:     BIT     #BIT10,E.CONT
3296  026016  001401                          BEQ     10$
3297  026020  104074                          ERROR   74              ;DRIVE DETECTED PARITY ERROR
3298  026022  032737  100000  002604  10$:    BIT     #BIT15,E.CONT
3299  026030  001401                          BEQ     11$
3300  026032  104052                          ERROR   52              ;MULTIPLE DRIVE SELECT
3301  026034  104075                  11$:    ERROR   75              ;UNDEFINED ERROR
3302  026036  032737  000400  006364  7$:     BIT     #LEV2ER,RECODE  ;TEST IF LEVEL 2 ERROR
3303  026044  001401                          BEQ     14$             ;NO - SKIP
3304  026046  104106                          ERROR   106
3305  026050  005037  002604          14$:    CLR     E.CONT          ;CLEAR CONTROLLER ERROR WORD
3306  026054  000137  030174                  JMP     CERTY           ;GO DO RETRY
3307                                          ;;*****************************************************************
3308                                          .SBTTL  REPORT SUPPORT ROUTINE
3309                                          ;*THIS ROUTINE LOADS ALL THE PARAMETER BLOCK DATA TO BE REPORTED
3310                                          ;*INTO THE PROPER TEMPORARY REGISTERS FOR REPORTING.  ALL THESE MAY
3311                                          ;*NOT BE INCLUDED IN THE REPORT, BUT THEY ARE LOADED ANYWAY.
3312  026060                                  REPSUP:
3313  026060  104411                                  SAVREG
3314  026062  005037  006372                          CLR     ERRCOM          ;CLEAR ERROR COMMAND STORE
3315  026066  116537  000001  006372                  MOVB    P.CMND(R5),ERRCOM ;STORE COMMAND START VALUES
3316  026074  012700  001162                          MOV     #$REGO,R0       ;FOR REPORTING
3317  026100  016520  000002                          MOV     P.CYLN(R5),(R0)+
3318  026104  116520  000005                          MOVB    P.TRCK(R5),(R0)+
3319  026110  105020                                  CLRB    (R0)+
3320  026112  116520  000004                          MOVB    P.SECT(R5),(R0)+
3321  026116  105020                                  CLRB    (R0)+
3322  026120  016520  000012                          MOV     P.WC(R5),(R0)+
3323  026124  016520  000010                          MOV     P.BALO(R5),(R0)+
3324  026130  016520  000016                          MOV     P.CS1(R5),(R0)+ ;GET ALL THE VALUES FROM THE
3325  026134  016520  000020                          MOV     P.CS2(R5),(R0)+ ;PARAMETER BLOCK AND LOAD
3326  026140  016520  000030                          MOV     P.DCYL(R5),(R0)+ ;THE TEMPORARY REGISTERS
3327  026144  016520  000026                          MOV     P.DTS(R5),(R0)+ ;FOR REPORTING. ALL THIS
3328  026150  016520  000022                          MOV     P.WCR(R5),(R0)+ ;DATA MAY NOT BE VALID
3329                                                                          ;FOR ALL REPORTS (TO BE
3330  026154  016520  000024                          MOV     P.BAR(R5),(R0)+ ;DETERMINED LATER) BUT IT IS
3331  026160  016520  000032                          MOV     P.ASOF(R5),(R0)+ ;STORED ANY WAY.
3332  026164  016520  000036                          MOV     P.DS(R5),(R0)+
```

```
 3333  026170  016520  000034                  MOV     P.ER(R5),(R0)+
 3334  026174  016520  000040                  MOV     P.A00(R5),(R0)+
 3335  026200  016520  000042                  MOV     P.B00(R5),(R0)+
 3336  026204  016520  000044                  MOV     P.A01(R5),(R0)+
 3337  026210  016520  000046                  MOV     P.B01(R5),(R0)+
 3338  026214  016520  000050                  MOV     P.A10(R5),(R0)+
 3339  026220  016520  000052                  MOV     P.B10(R5),(R0)+
 3340  026224  016520  000054                  MOV     P.A11(R5),(R0)+
 3341  026230  016520  000056                  MOV     P.B11(R5),(R0)+
 3342  026234  104412                          RESREG
 3343  026236  000207                          RTS     PC
 3344                                  ;;*****************************************************************
 3345                                  .SBTTL  REPORT ERROR ROUTINE
 3346                                  ;* ENTRY         JSR     PC, ERRHDL
 3347                                  ;*RETURN        RTS     PC
 3348                                  ;*
 3349                                  ;*THIS ROUTINE IS CALLED BY THE DRIVER WHEN AN ERROR IS DETECTED
 3350                                  ;*IN THE OPERATION.  THE ROUTINE DETERMINES WHICH COMMAND WAS
 3351                                  ;*BEING EXECUTED AND GENERATES THE PROPER ERROR MESSAGE.
 3352                                  ;;*****************************************************************
 3353
 3354  026240  104411                  ERRHDL: SAVREG
 3355  026242  152737  000377  002652          BISB    #377,DONE       ;SET DONE FLAG
 3356  026250  105237  002654                  INCB    ERRCNT          ;INCREMENT ERROR COUNT
 3357  026254  005037  006364                  CLR     RECODE          ;CLEAR RECOVERY CODE WORD
 3358  026260  032737  000400  006364  ER2ENT: BIT     #LEV2ER,RECODE  ;TEST IF 2ND LEVEL ERROR
 3359  026266  001402                          BEQ     52$             ;NO - SKIP PARAM BLOCK CHANGE
 3360  026270  012705  002446                  MOV     #PARM1,R5       ;ELSE SET R5 TO PARAMETER BLOCK 1
 3361  026274  012737  030310  002600  52$:    MOV     #RETANL,A.ABNL  ;SET NOW ABNORMAL AND NORMAL RETURN FOR
 3362  026302  012737  030326  002576          MOV     #RETNML,A.NORM  ;DRIVER OPERATIONS IN ERROR PROCESSING
 3363  026310  004737  026060                  JSR     PC,REPSUP       ;GO SET UP REGISTERS FOR REPORT
 3364                                                                  ;NOW BEGIN TESTING THE ERROR
 3365                                                                  ;BITS. THE SEQUENCE IN
 3366                                                                  ;WHICH THEY ARE TESTED IS
 3367                                                                  ;CONSIDERED SIGNIFICANT IN
 3368                                                                  ;THAT ERRORS OF A MORE
 3369                                                                  ;CATASTROPHIC NATURE ARE FIRST
 3370                                                                  ;TESTED.
 3371                                                                  ;
 3372                                                                  ;IF AN ERROR IS FOUND SET,
 3373                                                                  ;THAT ERROR IS REPORTED AND
 3374                                                                  ;THE REPORTING IS TERMINATED.
 3375                                                                  ;IF ADDITIONAL ERRORS ARE SET,
 3376                                                                  ;THE RK611 REGISTER PRINTOUTS
 3377                                                                  ;WILL SHOW THIS BUT THE
 3378                                                                  ;REGISTER CONTENTS MUST BE
 3379                                                                  ;MANUALLY DECODED TO LOCATE THE
 3380                                                                  ;SECOND ERROR
 3381  026314  016504  000034                  MOV     P.ER(R5),R4     ;SET R4 TO ERROR REGISTER
 3382  026320  032765  020000  000020          BIT     #UPE,P.CS2(R5)  ;TEST UPE. IF UES-SET
 3383  026326  001406                          BEQ     1$              ;REPORT ERROR
 3384  026330  104001                          ERROR   1
 3385  026332  052737  000200  006364          BIS     #ABORT,RECODE
 3386  026340  000137  027664                  JMP     37$
 3387  026344  032765  004000  000020  1$:     BIT     #NEM,P.CS2(R5)  ;TEST NON-EXISTANT MEMORY
 3388  026352  001406                          BEQ     2$
```

```
3389  026354  104002                        ERROR   2
3390  026356  052737  000200  006364        BIS     #ABORT,RECODE
3391  026364  000137  027664                JMP     37$
3392  026370  032765  010000  000020  2$:   BIT     #NED,P.CS2(R5)  ;TEST NON-EXISTANT DRIVE
3393  026376  001412                        BEQ     3$
3394  026400  032765  000400  000020        BIT     #UFE,P.CS2(R5)  ;TEST IF NED & UFE BOTH SET
3395  026406  001403                        BEQ     38$
3396  026410  104035                        ERROR   35              ;NED & UFE BOTH SET ERROR
3397  026412  000137  027664                JMP     37$
3398  026416  104003                38$:    ERROR   3               ;NED ONLY
3399  026420  000137  027664                JMP     37$
3400  026424  032765  000400  000020  3$:   BIT     #UFE,P.CS2(R5)  ;TEST UNIT FIELD ERROR
3401  026432  001412                        BEQ     4$
3402  026434  032765  010000  000020        BIT     #NED,P.CS2(R5)  ;TEST IF UFE & NED BOTH SET
3403  026442  001403                        BEQ     39$             ;NO-SKIP
3404  026444  104035                        ERROR   35              ;REPORT NED & UFE BOTH SET
3405  026446  000137  027664                JMP     37$
3406  026452  104004                39$:    ERROR   4               ;REPORT UFE ONLY
3407  026454  000137  027664                JMP     37$
3408  026460  032765  000100  000014  4$:   BIT     #CMDTO,P.PRST(R5) ;
3409  026466  001414                        BEQ     5$
3410  026470  004737  030336                JSR     PC,TOPROC       ;GO PROCESS TIMEOUT
3411  026474  032737  000400  006364        BIT     #LEV2ER,RECODE  ;TEST IF LEVEL 2 ERROR
3412  026502  001003                        BNE     41$             ;YES - SKIP TO ERROR 57
3413  026504  104005                        ERROR   5               ;ELSE MAKE FULL TIMEOUT REPORT
3414  026506  000137  027664                JMP     37$
3415  026512  104057                41$:    ERROR   57              ;2ND LEVEL ERROR REPORT
3416  026514  000137  027664                JMP     37$
3417  026520  032765  020000  000016  5$:   BIT     #SPAR,P.CS1(R5) ;TEST DRIVE BUS PARITY ERROR
3418  026526  001406                        BEQ     6$
3419  026530  104006                        ERROR   6
3420  026532  052737  004000  006364        BIS     #RCLREQ,RECODE
3421  026540  000137  027664                JMP     37$
3422  026544  032704  000010          6$:   BIT     #DRPAR,R4       ;TEST DRIVE DETECTED PARITY ERROR
3423  026550  001406                        BEQ     7$
3424  026552  104007                        ERROR   7
3425  026554  052737  004000  006364        BIS     #RCLREQ,RECODE
3426  026562  000137  027664                JMP     37$
3427  026566  032765  000010  000036  7$:   BIT     #ACLO,P.DS(R5)  ;TEST AC LOW
3428  026574  001403                        BEQ     8$
3429  026576  104010                        ERROR   10
3430  026600  000137  027664                JMP     37$
3431  026604  032765  000020  000036  8$:   BIT     #SPDLSS,P.DS(R5)        ;TEST SPEED LOSS
3432  026612  001403                        BEQ     24$
3433  026614  104011                        ERROR   11
3434  026616  000137  027664                JMP     37$
3435  026622  032765  C04000  000016  24$:  BIT     #CTO,P.CS1(R5)  ;TEST FOR CONTROLLER TIMEOUT
3436  026630  001401                        BEQ     25$
3437  026632  104027                        ERROR   27
3438  026634  032704  000001          25$:  BIT     #ILC,R4         ;TEST ILLEGAL FUNCTION CODE
3439  026640  001403                        BEQ     10$
3440  026642  104012                        ERROR   12
3441  026644  000137  027664                JMP     37$
3442  026650  032765  002000  000020  10$:  BIT     #PGE,P.CS2(R5)  ;TEST PROGRAMMING ERROR
3443  026656  001403                        BEQ     11$
3444  026660  104013                        ERROR   13
```

```
3445  026662  000137  027664                  JMP     37$
3446  026666  032704  000004          11$:    BIT     #NXF,R4         ;TEST ILLEGAL DRIVE FUNCTION
3447  026672  001403                          BEQ     12$
3448  026674  104014                          ERROR   14
3449  026676  000137  027664                  JMP     37$
3450  026702  032704  000040          12$:    BIT     #DTYE,R4        ;TEST DRIVE TYPE ERROR
3451  026706  001403                          BEQ     13$
3452  026710  104015                          ERROR   15
3453  026712  000137  027664                  JMP     37$
3454  026716  032704  000020          13$:    BIT     #FMTE,R4        ;TEST FORMAT ERROR
3455  026722  001403                          BEQ     14$
3456  026724  104016                          ERROR   16
3457  026726  000137  027664                  JMP     37$
3458  026732  032704  004000          14$:    BIT     #WLE,R4         ;TEST WRITE LOCK ERROR
3459  026736  001403                          BEQ     15$
3460  026740  104017                          ERROR   17
3461  026742  000137  027664                  JMP     37$
3462  026746  032704  040000          15$:    BIT     #UNS,R4         ;TEST DRIVE UNSAFE
3463  026752  001406                          BEQ     16$
3464  026754  104020                          ERROR   20
3465  026756  052737  000200  006364          BIS     #ABORT,RECODE
3466  026764  000137  027764                  JMP     ALLTRM
3467  026770  032704  000002          16$:    BIT     #SKI,R4         ;TEST SEEK INCOMPLETE
3468  026774  001406                          BEQ     17$
3469  026776  104021                          ERROR   21
3470  027000  052737  004000  006364          BIS     #RCLREQ,RECODE
3471  027006  000137  027664                  JMP     37$
3472  027012  032704  001000          17$:    BIT     #COE,R4         ;TEST CYLINDER OVERFLOW
3473  027016  001406                          BEQ     18$
3474  027020  104022                          ERROR   22
3475  027022  052737  004000  006364          BIS     #RCLREQ,RECODE
3476  027030  000137  027664                  JMP     37$
3477  027034  032704  002000          18$:    BIT     #IDAE,R4        ;TEST ILLEGAL CYLINDER
3478  027040  001406                          BEQ     19$
3479  027042  104023                          ERROR   23
3480  027044  052737  004000  006364          BIS     #RCLREQ,RECODE
3481  027052  000137  027664                  JMP     37$
3482  027056  032765  000040  000036  19$:    BIT     #DROT,P.DS(R5)  ;TEST DRIVE OFF TRACK
3483  027064  001406                          BEQ     20$
3484  027066  104024                          ERROR   24
3485  027070  052737  004000  006364          BIS     #RCLREQ,RECODE
3486  027076  000137  027664                  JMP     37$
3487  027102  032704  010000          20$:    BIT     #DTE,R4         ;TEST DRIVE TIMING ERROR
3488  027106  001406                          BEQ     21$
3489  027110  104025                          ERROR   25
3490  027112  052737  000200  006364          BIS     #ABORT,RECODE
3491  027120  000137  027664                  JMP     37$
3492  027124  032765  100000  000020  21$:    BIT     #DLT,P.CS2(R5)  ;TEST DATA LATE
3493  027132  001403                          BEQ     22$
3494  027134  104026                          ERROR   26
3495  027136  000137  027664                  JMP     37$
3496  027142  032704  020000          22$:    BIT     #OPI,R4         ;TEST IF OPI ERROR
3497  027146  001457                          BEQ     29$
3498  027150  052737  000010  006364          BIS     #OPIERR,RECODE
3499  027156  105737  002650                  TSTB    DERCNT          ;TEST IF FIRST ERROR
3500  027162  001402                          BEQ     50$
```

```
3501  027164  000137  027664                    JMP     37$             ;NO - SKIP REPORT
3502  027170  032737  000400  006364  50$:      BIT     #LEV2ER,RECODE  ;TEST IF A SECOND LEVEL 2 ERROR
3503  027176  001403                            BEQ     26$             ;HAS ALREADY OCCURRED
3504  027200  104054                  27$:      ERROR   54
3505  027202  000137  027664                    JMP     37$             ;GET OUT OF ERROR REPORT
3506  027206  004737  031324          26$:      JSR     PC,BLDEXH       ;GO BUILD EXPECTED HEADER
3507  027212  004737  030700                    JSR     PC,RDHD0        ;GET HEADER OF SECTOR 0
3508  027216  032737  000400  006364            BIT     #LEV2ER,RECODE  ;TEST IF ERROR GETTING HDR
3509  027224  001025                            BNE     28$             ;IF YES-GO MAKE ABREVIATED REPORT
3510  027226  013702  002570                    MOV     RKBAS,R2        ;STORE HEADER 0 INTO REGISTERS
3511  027232  042762  000100  000000            BIC     #IE,RKCS1(R2)   ;RESET INTERRUPT ENABLE
3512  027240  016237  000024  001202            MOV     RKDB(R2),$REG10 ;FOR REPORTING
3513  027246  016237  000024  001204            MOV     RKDB(R2),$REG11
3514  027254  016237  000024  001206            MOV     RKDB(R2),$REG12
3515  027262  032762  100000  000000            BIT     #CERR,RKCS1(R2) ;TEST IF ERROR DURING STORAGE
3516  027270  001343                            BNE     27$
3517  027272  104030                            ERROR   30              ;MAKE OPI REPORT
3518  027274  000137  027664                    JMP     37$
3519  027300  104054                  28$:      ERROR   54
3520  027302  000137  026260                    JMP     ER2ENT          ;GO MAKE 2ND LEVEL REPORT
3521  027306  032704  000400          29$:      BIT     #HVRC,R4        ;TEST IF HVRC ERROR
3522  027312  001457                            BEQ     23$
3523  027314  052737  000004  006364            BIS     #HVRCER,RECODE
3524  027322  105737  002650                    TSTB    DERCNT          ;TEST IF FIRST ERROR
3525  027326  001402                            BEQ     30$             ;YES - REPORT
3526  027330  000137  027664                    JMP     37$             ;JUMP TO RETURN
3527  027334  032737  000400  006364  30$:      BIT     #LEV2ER,RECODE  ;TEST IF A 2ND LEVEL ERROR HAS ALREADY
3528  027342  001403                            BEQ     31$             ;OCCURRED. NO-SKIP EXIT
3529  027344  104055                  51$:      ERROR   55
3530  027346  000137  027664                    JMP     37$             ;GET OUT OF ERROR REPORT
3531  027352  004737  031324          31$:      JSR     PC,BLDEXH       ;GO BUILD EXPECTED HEADER
3532  027356  004737  030700                    JSR     PC,RDHD0        ;GO GET HDR 0
3533  027362  032737  000400  006364            BIT     #LEV2ER,RECODE  ;TEST IF ERROR IN GETTING HDR
3534  027370  001025                            BNE     32$             ;IF YES-GO MAKE ABREVIATED REPORT
3535  027372  013702  002570                    MOV     RKBAS,R2        ;GET RK611 BASE ADDRESS
3536  027376  042762  000100  000000            BIC     #IE,RKCS1(R2)   ;CLEAR INTERRUPT ENABLE
3537  027404  016237  000024  001202            MOV     RKDB(R2),$REG10 ;STORE OFF HEADER
3538  027412  016237  000024  001204            MOV     RKDB(R2),$REG11
3539  027420  016237  000024  001206            MOV     RKDB(R2),$REG12
3540  027426  032762  100000  000000            BIT     #CERR,RKCS1(R2) ;TEST IN ANY ERROR IN UNLOAD
3541  027434  001343                            BNE     51$             ;IY YES - GO MAKE SHORT REPORT
3542  027436  104031                            ERROR   31              ;MAKE FULL REPORT
3543  027440  000137  027664                    JMP     37$
3544  027444  104055                  32$:      ERROR   55              ;ABREVIATED HVRC ERROR RPORT
3545  027446  000137  026260                    JMP     ER2ENT          ;GO REPORT 2ND LEVEL ERROR
3546  027452  032704  000200          23$:      BIT     #BSE,R4         ;TEST FOR BAD SECTOR ERROR
3547  027456  001422                            BEQ     33$             ;NO - SKIP
3548  027460  052737  000002  006364            BIS     #BSERR,RECODE   ;SET ERROR FLAG
3549  027466  016502  000026                    MOV     P.DTS(R5),R2    ;GET SECTOR IN ERROR
3550  027472  042702  177740                    BIC     #177740,R2      ;CLEAR ALL BITS EXCEPT SECTOR
3551  027476  004737  021476                    JSR     PC,BDSRCK       ;GO SEE IF THIS SECTOR LISTED
3552  027502  032737  001000  006364            BIT     #BADSEC,RECODE  ;TEST RESULT
3553  027510  001065                            BNE     37$             ;YES - EXIT, NO ERROR OR REPORT
3554  027512  104104                            ERROR   104             ;ELSE REPORT BAD BSE
3555  027514  042737  000002  006364            BIC     #BSERR,RECODE   ;RESET BSE ERROR FLAG
3556  027522  000460                            BR      37$             ;GO EXIT
```

```
3557  027524  032704  100000          33$:   BIT   #DCK,R4           ;TEST IF DATA CHECK
3558  027530  001435                         BEQ   36$
3559  027532  052737  000020  006364         BIS   #DCKERR,RECODE    ;SET DATA CHECK ERROR IN RECOVERY CODE
3560  027540  032704  000100                 BIT   #ECH,R4           ;TEST IF ECC IS HARD. IF
3561  027544  001406                         BEQ   34$              ;YES SET UNCORRECTABLE IN
3562  027546  052737  000040  006364         BIS   #ECCNC,RECODE     ;RECOVERY FLAG AND A 0 IN
3563  027554  005037  001206                 CLR   $REG12           ;REG12 TO INDICATE UNCORRECTABLE,
3564  027560  000403                         BR    35$              ;A 1 IN REG 12 FOR CORRECTABLE
3565  027562  012737  000001  001206   34$:   MOV   #1,$REG12
3566  027570  105737  002650          35$:   TSTB  DERCNT            ;TEST IF FIRST ERROR
3567  027574  001033                         BNE   37$              ;NO SKIP REPORT
3568  027576  004737  031200                 JSR   PC,GTPKAD        ;GO GET PACK ADDRESS OF ERROR
3569  027602  016537  000060  001202         MOV   P.EPOS(R5),$REG10 ;STORE ECC POSITION &
3570  027610  016537  000062  001204         MOV   P.EPAT(R5),$REG11 ;PATTERN
3571  027616  104032                         ERROR 32               ;REPORT DCK ERROR
3572  027620  000137  027664                 JMP   37$
3573  027624  032765  040000  000020   36$:   BIT   #WCE,P.CS2(R5)  ;TEST WRITE CHECK ERROR
3574  027632  001414                         BEQ   37$
3575  027634  042737  000200  006364         BIC   #ABORT,RECODE    ;CLEAR ABORT & SET WRITE
3576  027642  052737  000100  006364         BIS   #WCERR,RECODE    ;CHECK ERROR IN RECODE
3577  027650  105737  002650                 TSTB  DERCNT           ;TEST IF FIRST ERROR
3578  027654  001003                         BNE   37$              ;NO - SKIP
3579  027656  004737  031200                 JSR   PC,GTPKAD        ;GO GET ADDRESS OF ERROR
3580  027662  104023                         ERROR 33               ;REPORT WCE
3581
3582  027664  032765  000020  000014   37$:   BIT   #DRVHRD,P.PRST(R5) ;TEST HARD ERROR
3583  027672  001404                         BEQ   43$
3584  027674  104036                         ERROR 36
3585  027676  052737  000200  006364         BIS   #ABORT,RECODE
3586
3587  027704  032765  000040  000014   43$:   BIT   #DRVDSC,P.PRST(R5) ;TEST STATUS CHANGE NOT CLEARED
3588  027712  001404                         BEQ   44$
3589  027714  104037                         ERROR 37
3590  027716  052737  000200  006364         BIS   #ABORT,RECODE
3591
3592  027724  032765  004000  000014   44$:   BIT   #MODSC,P.PRST(R5) ;IFST ATTENTION BUT NO FAULT OR DSC
3593  027732  001404                         BEQ   46$
3594  027734  104040                         ERROR 40
3595  027736  052737  000200  006364         BIS   #ABORT,RECODE
3596  027744  032765  000010  000014   46$:   BIT   #UEXATT,P.PRST(R5) ;TEST UNEXPECTED ATTENTION
3597  027752  001404                         BEQ   ALLTRM
3598  027754  104042                         ERROR 42
3599  027756  052737  000200  006364         BIS   #ABORT,RECODE
3600
3601
3602                                  ;ALL ERRORS MUST EXIT THROUGH THIS POINT
3603
3604  027764  012705  002362          ALLTRM: MOV   #PARM0,R5        ;RESTORE PARAMETER BLOCK SELECTION
3605  027770  012737  026240  002600         MOV   #ERRHDL,A.ABNL    ;RESTORE INTERRUPT VECTORS FOR RETRY
3606  027776  012737  025302  002576         MOV   #ERRFRE,A.NORM    ;DRIVER OPERATIONS, IF ANY
3607  030004  032737  000200  006364         BIT   #ABORT,RECODE     ;IF ABORT IS NOT SET AND
3608  030012  001050                         BNE   48$              ;THE DRIVE IS READY (HAS NOT
3609                                                                 ;CYCLED DOWN)
3610  030014  013702  002570                 MOV   RKBAS,R2         ;GET BASE ADDRESS
3611  030020  032762  000200  000012         BIT   #DRDY,RKDS(R2)   ;TEST IF DRIVE READY SET
3612  030026  001004                         BNE   47$              ;RECALIBRATE REQUIRED BIT IS SET
```

```
3613  030030  052737  000200  006364          BIS     #ABORT,RECODE   ;ELSE ABORT WITH ABORT MESSAGE
3614  030036  000436                          BR      46$
3615  030040  032737  004000  006364  47$:    BIT     #RCLREQ,RECODE  ;IF RECALIBRATE IS REQUIRED
3616  030046  001436                          BEQ     BGNRTY          ;FOR RETRY SET UP PARAM
3617  030050  012705  002446                  MOV     #PARM1,R5       ;SET TO PARAMETER BLOCK ONE
3618  030054  113765  000113  000001          MOVB    #RECAL,P.CMND(R5)       ;BLOCK TO DO IT.
3619  030062  012737  030310  002600          MOV     #RETANL,A.ABNL
3620  030070  004737  025220                  JSR     PC,DRVCAL
3621  030074  012737  026240  002600          MOV     #ERRHDL,A.ABNL  ;RESTORE ERROR RETURN
3622  030102  032737  000400  006364          BIT     #LEV2ER,RECODE  ;IF AN ERROR OCCURRED IN THE
3623  030110  001003                          BNE     49$             ;RECAL ATTEMP SET ABORT,
3624  030112  012705  002362                  MOV     #PARM0,R5       ;RESET PARAM TO BLOCK 0
3625  030116  000412                          BR      BGNRTY
3626  030120  052737  000200  006364  49$:    BIS     #ABORT,RECODE   ;PRINT THE RECAL ERROR MESSAGE, AND
3627  030126  104060                          ERROR   60              ;GO REPORT DETAILS
3628  030130  000137  026260                  JMP     ER2ENT
3629  030134  104061                  48$:    ERROR   61              ;REPORT ABORT-RETRY FAILED
3630
3631                                  ;THE PROGRAM WILL HALT HERE IF THE ABORT FLAG HAS BEEN SET OR
3632                                  ;IF THE DRIVE READY BIT IS RESET.
3633
3634  030136  000000                  HLTPRG: HALT
3635  030140  000137  016100                  JMP     START
3636
3637
3638                                  ;THE FOLLOWING CODE WILL DETERMINE IF AND DATA TYPE ERROR
3639                                  ;HAS OCCURRED. IF YES, THE RETRY IS NOT DONE HERE BUT RETURNS TO
3640                                  ;  THE INITIATING ROUTINE FOR RETRY.  ANY OTHER ERROR IS TO BE
3641                                  ;RETRIED HERE. IF RETRY IS UNSUCCESSFUL AFTER 4 ATTEMPTS, THE ABORT
3642                                          ;FLAG IS SET AND PROGRAM HALTS.
3643
3644  030144  032737  000136  006364  BGNRTY: BIT     #BSERR!HVRCER!OPIERR!DCKERR!WCERR,RECODE ;TEST IF ANY DATA ERROR
3645  030152  001404                          BEQ     DOL3
3646  030154  052737  100000  006364          BIS     #ANYDER,RECODE
3647  030162  000450                          BR      DOL2
3648  030164                          DOL3:
3649  030164  032737  001000  006364          BIT     #BADSEC,RECODE  ;TEST IF BAD SECTOR FLAG SET
3650  030172  001044                          BNE     DOL2            ;IF YES-EXIT TO CALLER
3651  030174  123737  002654  002655  CERTY:  CMPB    ERRCNT,ERRLMT   ;BEGIN RETRY IF ERROR COUNT HAS
3652  030202  001012                          BNE     DOL1            ;NOT BEEN EXCEEDED
3653  030204  005037  001174                  CLR     $REGS
3654  030210  113737  002654  001174          MOVB    ERRCNT,$REGS    ;GET ERROR RETRY COUNT
3655  030216  104102                          ERROR   102             ;REPORT RETRY UNSUCCESSFUL
3656  030220  052737  000200  006364          BIS     #ABORT,RECODE   ;SET ABORT & QUIT
3657  030226  000743                          BR      HLTPRG
3658  030230  013702  002570          DOL1:   MOV     RKBAS,R2        ;GET BASE ADDRESS
3659  030234  012762  000040  000010          MOV     #SCLR,RKCS2(R2) ;CLEAR SUBSYSTEM
3660  030242  013746  000000                  MOV     0,-(SP) ;;PUT NEW PS ON STACK
3661  030246  012746  030254                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
3662  030252  000002                          RTI                     ;;POP NEW PC AND PS
3663  030254                          64$:
3664  030254  012700  005660                  MOV     #COMSTR,R0      ;GO AND REESTABLISH THE COMMAND
3665  030260  012025                          MOV     (R0)+,(R5)+     ;AS IT WAS ENTERED INTO THE
3666  030262  012025                          MOV     (R0)+,(R5)+     ;PARAMETER BLOCK
3667  030264  012025                          MOV     (R0)+,(R5)+
3668  030266  012025                          MOV     (R0)+,(R5)+
```

```
3669  030270  012025                        MOV     (R0)+,(R5)+
3670  030272  012025                        MOV     (R0)+,(R5)+
3671  030274  012705  002362                MOV     #PARM0,R5
3672  030300  004737  025220                JSR     PC,DRVCAL       ;CALL DRIVER
3673  030304  104412                DOL2:   RESREG                  ;IF RETURN GETS HERE, NO ERROR
3674  030306  000207                        RTS     PC              ;OCCURRED, RECOVERY WAS SUCCESSFUL
3675  030310  152737  000377  002652 RETANL: BISB    #377,DONE       ;SET DONE
3676  030316  052737  000400  006364        BIS     #LEV2ER,RECODE  ;SET LEVEL TWO ERROR
3677  030324  000207                        RTS     PC
3678  030326  152737  000377  002652 RETNML: BISB    #377,DONE       ;SET DONE
3679  030334  000207                        RTS     PC
3680                                ;;*****************************************************************
3681                                .SBTTL  TIME OUT PROCESSOR ROUTINE
3682                                ;*THIS ROUTINE SUPPORTS THE ERROR HANDLER BY PROCESSING TIME OUT STATUS
3683                                ;*GATHERING DUTIES.
3684                                ;;*****************************************************************
3685  030336                        TOPROC:
3686  030336  104411                        SAVREG
3687  030340  013702  002570                MOV     RKBAS,R2
3688  030344  012701  001162                MOV     #$REGD,R1       ;SET UP R1 FOR RK REGISTER STORAGE
3689  030350  005037  006372                CLR     ERRCOM          ;CLEAR FOR COMMAND
3690  030354  116537  000001  006372        MOVB    P.CMND(R5),ERRCOM ;STORE COMMAND
3691  030362  016521  000002                MOV     P.CYLN(R5),(R1)+ ;STORE CYLINDER
3692  030366  116521  000005                MOVB    P.TRCK(R5),(R1)+ ;STORE TRACK
3693  030372  105021                        CLRB    (R1)+
3694  030374  116521  000004                MOVB    P.SECT(R5),(R1)+ ;STORE SECTOR
3695  030400  105021                        CLRB    (R1)+
3696  030402  016521  000012                MOV     P.WC(R5),(R1)+ ;STORE WORD COUNT
3697  030406  016521  000010                MOV     P.BALO(R5),(R1)+ ;STORE BUS ADDRESS
3698  030412  016221  000000                MOV     RKCS1(R2),(R1)+ ;STORE ALL PARAMERERS AND VALID RK611
3699  030416  016221  000010                MOV     RKCS2(R2),(R1)+ ;REGISTERS
3700  030422  016221  000020                MOV     RKDC(R2),(R1)+
3701  030426  016221  000006                MOV     RKDA(R2),(R1)+
3702  030432  016221  000002                MOV     RKWC(R2),(R1)+
3703  030436  016221  000004                MOV     RKBA(R2),(R1)+
3704  030442  016221  000016                MOV     RKASOF(R2),(R1)+
3705  030446  016221  000012                MOV     RKDS(R2),(R1)+
3706  030452  016221  000014                MOV     RKER(R2),(R1)+
3707                                                                ;THIS CODE WILL ATTEMPT TC
3708                                                                ;RETRIEVE THE STATUS FROM THE
3709                                                                ;DRIVE.
3710  030456  004737  030466                JSR     PC,GETDRST      ;GO GET DRIVE STATUS
3711  030462  104412                        RESREG
3712  030464  000207                        RTS     PC
3713                                ;;*****************************************************************
3714                                .SBTTL  RETRIEVE DRIVE STATUS ROUTINE
3715
3716  030466  104411                GETDRST:        SAVREG
3717  030470  005000                        CLR     R0
3718  030472  012762  100000  000000        MOV     #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
3719  030500  010146                        MOV     R1,-(SP)        ;STORE R5
3720  030502  012703  000010                MOV     #10,R3          ;SET COUNT FOR STATUS CLEAR
3721  030506  012721  123456        1$:     MOV     #123456,(R1)+   ;INSERT CLEAR WORD
3722  030512  005303                        DEC     R3              ;BUMP COUNTER
3723  030514  001374                        BNE     1$              ;LOOP UNTIL 0
3724  030516  012601                        MOV     (SP)+,R1        ;RESTORE R5
```

```
3725  030520  152737  000377  002632          B1SB    #377,W.TIME     ;GIVE WATCH DOG SOMETIHING TO TIME
3726  030526  013746  002600                  MOV     A.ABNL,-(SP)    ;STORE ERROR RETURN ADDRESS
3727  030532  012737  030310  002600          MOV     #RETANL,A.ABNL  ;SET UP NEW ERROR RETURN
3728  030540  013762  006340  000010          MOV     DRINUS,RKCS2(R2)        ;LOAD DRIVE NUMBER
3729  030546  032762  100000  000000          BIT     #CERR,RKCS1(R2) ;TEST IF CONT ERROR SET
3730  030554  001037                          BNE     4$              ;YES - GET OUT
3731  030556  010062  000022          3$:     MOV     R0,RKMR1(R2)    ;SET UP STATUS WORD TO READ
3732  030562  012762  000001  000000          MOV     #1,RKCS1(R2)    ;DO DRIVE SELECT
3733  030570  016204  000000                  MOV     RKCS1(R2),R4    ;GET CS1
3734  030574  032704  000200                  BIT     #RDY,R4         ;TEST IF READY
3735  030600  001007                          BNE     2$              ;YES - SKIP
3736  030602  004737  032102                  JSR     PC,W.WTCH       ;CALL WATCH DOG
3737  030606  032765  000100  000014          BIT     #CMDTO,P.PRST(R5)       ;TEST IF TIMEOUT
3738  030614  001360                          BNE     3$              ;NO - LOOP
3739  030616  000416                          BR      4$              ;ELSE GET OUT
3740  030620  016204  000000          2$:     MOV     RKCS1(R2),R4    ;GET CS1
3741  030624  032704  100000                  BIT     #CERR,R4        ;TEST IF ERROR
3742  030630  001011                          BNE     4$              ;YES - GET OUT
3743  030632  016221  000034                  MOV     RKMR2(R2),(R1)+ ;ELSE STORE STATUS WORDS JUST READ
3744  030636  016221  000036                  MOV     RKMR3(R2),(R1)+
3745  030642  005200                          INC     R0              ;BUMP TO NEXT STATLS PAIR
3746  030644  020027  000004                  CMP     R0,#4           ;TEST IF ALL HAVE BEEN READ
3747  030650  001342                          BNE     3$              ;YES - GOOD EXIT
3748  030652  000406                          BR      5$
3749  030654  052737  000400  006364  4$:     BIS     #LEV2ER,RECODE  ;SET LEVEL 2 ERROR
3750  030662  012762  000040  000010          MOV     #SCLR,RKCS2(R2) ;CLEAR SUBSYSTEM
3751  030670  012637  002600          5$:     MOV     (SP)+,A.ABNL    ;RESTORE OLD ERROR RETURN
3752  030674  104412                          RESREG
3753  030676  000207                          RTS     PC
3754                                  ;;**********************************************************************
3755                                  .SBTTL  READ HEADER 0 ROUTINE
3756                                  ;;**********************************************************************
3757  030700                          RDHD0:
3758  030700  104411                          SAVREG
3759  030702  016501  000026                  MOV     P.DTS(R5),R1    ;STORE TRACK AND SECTOR
3760  030706  016500  000052                  MOV     P.B10(R5),R0    ;GET THE CYLINDER ADDRESS
3761  030712  042700  160013                  BIC     #160013,R0      ;FROM THE DRIVE STATUS. CLEAR
3762  030716  006200                          ASR     R0              ;OFF UNUSED BITS AND POSITION
3763  030720  006200                          ASR     R0              ;FOR USE AS THE DESIRED
3764  030722  006200                          ASR     R0              ;CYLINDER IN THE READ
3765  030724  006200                          ASR     R0              ;HEADER COMMAND
3766  030726  012705  002446                  MOV     #PARM1,R5       ;SET UP TO USE PARM 1
3767  030732  010165  000004                  MOV     R1,P.SECT(R5)   ;INSERT TRK/SECT FOR READ HEADER
3768  030736  010065  000002                  MOV     R0,P.CYLN(R5)
3769  030742  132737  000020  002653          BITB    #B.CFMT,TYPFMT  ;TEST PRESENT FORMAT & CHANGE
3770  030750  001404                          BEQ     1$              ;TO THE OPPOSITE. THIS WILL CAUSE
3771  030752  142765  000020  000007          BICB    #B.CFMT,P.CS1H(R5) ;A READ OFSECTOR 0 HEADER ON
3772  030760  000403                          BR      2$              ;THE READ HEADER COMMAND
3773  030762  152765  000020  000007  1$:     BISB    #B.CFMT,P.CS1H(R5)
3774  030770  112765  000125  000001  2$:     MOVB    #RDHEAD,P.CMND(R5) ;SET UP TO READ HEADER
3775  030776  013746  000000                  MOV     0,-(SP) ;;PUT NEW PS ON STACK
3776  031002  012746  031010                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
3777  031006  000002                          RTI                     ;;POP NEW PC AND PS
3778  031010                          64$:
3779  031010  004737  025220                  JSR     PC,DRVCAL       ;CALL DRIVER
3780  031014  142765  000020  000007          BICB    #B.CFMT,P.CS1H(R5)      ;CLEAR THE FORMAT BIT
```

```
                                        BISB    TYPFMT,P.CS1H(R5)       ;RESTORE TYPE AND FORMAT IN USE
                                        MOV     #PARM0,R5       ;RESTORE PARM 0 BLOCK
                                        RESREG          ;RETURN
                                        RTS     PC
                                ;;*****************************************************************
                                .SBTTL  POSITION AND OFFSET ROUTINE
                                ;*THIS ROUTINE CHECKS IF OFFSET IS REQUIRED. IF IT IS, THE HEADS
                                ;*POSITIONED TO THE CYLINDER SPECIFIED IN R0 AND THE HEADS ARE OFFSET
                                ;*TO THE AMOUNT SPECIFIED IN "OFINUS".
                                ;;*****************************************************************

                                POSOFF: SAVREG
                                        BITB    #OREQSW,OPCONT  ;TEST IF OFFSET REQUIRED
                                        BEQ     1$              ;IF NO - SKIP TO EXIT
                                        MOVB    #SEEK,P.CMND(R5)  ; SET UP TO DO THE SEEK
                                        MOV     R0,P.CYLN(R5)   ;SET CYLINDER
                                        JSR     PC,DRVCAL       ;GO DO IT
                                        MOVB    #OFFSET,P.CMND(R5)  ;SET UP TO OFFSET
                                        MOVB    OFINUS,P.OFST(R5)  ;SET OFFSET VALUE
                                        COMB    P.ASOF(R5)      ;COMPLEMENT ASOF
                                        BICB    #200,P.ASOF(R5) ;CLEAR UPPER BIT
                                        JSR     PC,DRVCAL       ;GO DO IT
                                1$:     RESREG
                                        RTS     PC

                                ;;*****************************************************************
                                ;;*****************************************************************
                                .SBTTL  PREPARE FOR LOOP ON ERROR
                                PREPLP: CLRB    ERRCNT          ;CLEAR ERROR COUNT
                                        CLR     RECODE          ;CLEAR RECODE
                                        MOV     CCINUS,R0       ;SET CURRENT CYLINDER
                                        MOV     CTINUS,R1       ;SET CURRENT TRACK
                                        MOV     #ERRHDL,A.ABNL  ;SET UP ERROR RETURN
                                        MOV     #ERRFRE,A.NORM  ;SET UP ERROR FREE RETURN
                                        MOV     #STACK-4,R4     ;SET R4 TO STACK BASE -4
                                        MOV     (SP),(R4)+      ;STORE RETURN ON STACK
                                        CLR     (R4)            ;CLEAR FOR 0 PRIORITY
                                        MOV     #STACK-4,SP     ;SET STACK POINTER FOR RETURN
                                        RTI                     ;DO RTI TO RETURN TO LOOPING ROUTINE

                                .SBTTL  GET PACK ADDRESS ROUTINE
                                ;;*****************************************************************
                                GTPKAD: MOV     P.DCYL(R5),$REG5 ;GET CYLINDER NUMBER
                                        CLR     $REG6           ;CLEAR REGISTERS FOR
                                        CLR     $REG7           ;TRACK & SECTOR STORAGE
                                        MOVB    P.DTS(R5),$REG7 ;STORE THE TRACK AND SECTOR
                                        MOVB    P.DTS+1(R5),$REG6
                                        TST     $REG7           ;ADJUST THE ADDRESS CONTAINED IN
                                                                ;THE RK REGISTERS FOR THE AUTOMATIC
                                        BEQ     1$              ;INCREMENT
                                        DEC     $REG7
                                        BR      5$
                                1$:     BIT     #CFMT,P.CS1(R5)
                                        BEQ     2$
                                        MOV     #19.,$REG7
                                        BR      3$
```

```
3837  031266  012737  000025  001200  2$:     MOV     #21.,$REG7
3838  031274  005737  001176          3$:     TST     $REG6
3839  031300  001403                          BEQ     4$
3840  031302  005337  001176                  DEC     $REG6
3841  031306  000405                          BR      5$
3842  031310  012737  000002  001176  4$:     MOV     #2,$REG6
3843  031316  005337  001174                  DEC     $REG5
3844  031322  000207                  5$:     RTS     PC
3845                                  ;;*****************************************************************
3846                                  .SBTTL  BUILD EXPECTED HEADER
3847                                  ;*USES DESIRED CYLINDER, TRACK AND SECTOR REGISTERS TO DETERMINE
3848                                  ;*WHICH HEADER WAS EXPECTED. LOADS EXPECTED VALUES IN $REG5, 6, AND
3849                                  ;*7 FOR REPORTING.
3850                                  ;;*****************************************************************
3851  031324  104411                  BLDEXH: SAVREG
3852  031326  016537  000030  001174          MOV     P.DCYL(R5),$REG5 ;CONSTRUCT EXPECTED HDR
3853  031334  016501  000026                  MOV     P.DTS(R5),R1 ;DESIRED CYLINDER & DESIRED TRACK
3854  031340  042701  174377                  BIC     #174377,R1      ;CLEAR ALL BUT TRACK BITS
3855  031344  006201                          ASR     R1              ;AND SECTOR. SHIFT THE TRACK
3856  031346  006201                          ASR     R1              ;OVER TO CONFORM TO HEADER FORMAT
3857  031350  006201                          ASR     R1              ;CHECK THE FORMAT BIT AND
3858                                                                  ;IF SET, SET THE HEADER FORMAT
3859  031352  016537  000026  001176          MOV     P.DTS(R5),$REG6 ;BIT.
3860  031360  042737  177740  001176          BIC     #177740,$REG6   ;CLEAR ALL BUT SECTOR
3861  031366  060137  001176                  ADD     R1,$REG6        ;ADD TRACK AND SECTOR TOGETHER
3862  031372  052737  140000  001176          BIS     #140000,$REG6   ;INSERT BSE BITS
3863  031400  032765  010000  000016          BIT     #CFMT,P.CS1(R5)
3864  031406  001403                          BEQ     23$
3865  031410  052737  001000  001176          BIS     #1000,$REG6
3866  031416  013737  001174  001200  23$:    MOV     $REG5,$REG7     ;COMPUTE THE HEADER VRC
3867  031424  013701  001176                  MOV     $REG6,R1
3868  031430  043737  001176  001200          BIC     $REG6,$REG7
3869  031436  043701  001174                  BIC     $REG5,R1
3870  031442  050137  001200                  BIS     R1,$REG7
3871  031446  104412                          RESREG
3872  031450  000207                          RTS     PC
3873                                  ;;*****************************************************************
3874                                  .SBTTL  WRITE BAD SECTOR FILE ROUTINE
3875                                  ;*THIS ROUTINE WRITES THE BSSOFT (BAD SECTOR FILE DETECTED BY SOFTWARE)
3876                                  ;*INTO THE APPROPRIATE BAD SECTOR FILE. IT DOES NOT WRITE THE FACTORY
3877                                  ;*DETECTED FILES.
3878                                  ;;*****************************************************************
3879  031452                          BDSTWT:
3880  031452  104411                          SAVREG
3881  031454  105037  002650                  CLRB    DERCNT          ;CLEAR ERROR COUNT
3882  031460  012703  000006                  MOV     #6,R3           ;PRESET FOR 6 SECTORS TO BE WRITTEN
3883  031464  012702  000012                  MOV     #12,R2          ;PRESET TO 22 SECTOR/TRACK
3884  031470  005065  000014                  CLR     P.PRST(R5)      ;CLEAR PROG STAT REG
3885  031474  132737  000020  002653          BITB    #B.CFMT,TYPFMT  ;TEST IF 20 OR 22 SEC/TRACK
3886  031502  001404                          BEQ     1$              ;22 SECTOR - SKIP
3887  031504  005202                          INC     R2              ;20 SECTOR - BUMP TO NEXT SECTOR (13)
3888  031506  142765  000020  000007          BICB    #B.CFMT,P.CS1H(R5)  ;CLEAR FOR 22 SECTOR WRITE
3889  031514  012765  003660  000010  1$:     MOV     #BSSOFT,P.BALO(R5)  ;SET UP PARAMETER BLOCK
3890  031522  112765  000123  000001          MOVB    #WRDATA,P.CMND(R5)  ;TO WRITE THE SECTORS OF BAD
3891  031530  012765  000632  000002          MOV     #632,P.CYLN(R5) ;SECTOR TABLE (SOFTWARE DETECTED)
3892  031536  112765  000002  000005          MOVB    #2,P.TRCK(R5)
```

```
 3893  031544  110265  000004                  MOVB    R2,P.SECT(R5)   ;INSERT SECTOR NUMBER
 3894  031550  012765  177400  000012          MOV     #177400,P.WC(R5)
 3895  031556  004737  025220                  JSR     PC,DRVCAL
 3896  031562  032737  100000  006364          BIT     #ANYDER,RECODE  ;CHECK IF ANY ERROR IN WRITE
 3897  031570  001005                          BNE     3$              ;IF YES - SKIP
 3898  031572  062702  000002          2$:     ADD     #2,R2           ;ELSE BUMP SECTOR BY TWO
 3899  031576  005303                          DEC     R3              ;DECREMENT COUNTER
 3900  031600  001345                          BNE     1$              ;EXIT IF DONE
 3901  031602  000417                          BR      4$
 3902  031604  105237  002650          3$:     INCB    DERCNT          ;COUNT ERROR FOR RETR.
 3903  031610  122737  000007  002650          CMPB    #7,DERCNT       ;IF TO MANY TYPE MESSAGE
 3904  031616  002736                          BLT     1$              ;ELSE RETRY
 3905  031620  104401  015402                  TYPE    ,BSWERR         ;ERROR MESSAGE
 3906  031624  104401  015470                  TYPE    ,SECINE
 3907  031630  010246                          MOV     R2,-(SP)
 3908  031632  104402                          TYPOC
 3909  031634  104401  001267                  TYPE    ,$CRLF
 3910  031640  000754                          BR      2$
 3911  031642  104401  015512          4$:     TYPE    ,BSWTFI         ;TYPE FINISHED MESSAGE
 3912  031646  104412                          RESREG
 3913  031650  000207                          RTS     PC
 3914                                  ;;*****************************************************************
 3915                                  .SBTTL  BAD SECTOR REPORT ROUTINE
 3916                                  ;;*****************************************************************
 3917  031652                          BSREPT:
 3918  031652  104411                          SAVREG
 3919  031654  005037  001242                  CLR     $TMP2           ;CLEAR TEMP 2 FOR PASS SWITCH
 3920  031660  104401  015547                  TYPE    ,BSHEAD         ;TYPE HEADER LINE
 3921  031664  005000                          CLR     R0              ;CLEAR R0 FOR COUNTER
 3922  031666  012737  015763  031714          MOV     #TWENT2,2$      ;CHOSE THE MODE TO REPORT
 3923  031674  132737  000020  002653          BITB    #B.CFMT,TYPFMT  ;20 OR 22 SECTOR FORMAT
 3924  031702  001403                          BEQ     1$              ;QUESSED RIGHT - SKIP
 3925  031704  012737  015760  031714          MOV     #TWENTY,2$      ;QUESSED WRONG, SET TO 22 SECTOR
 3926  031712  104401                  1$:     TYPE
 3927  031714  000000                  2$:     .WORD
 3928  031716  104401  015617                  TYPE    ,BSTAIL
 3929  031722  104401  015645                  TYPE    ,FBSLAB         ;TYPE FACTORY LABEL
 3930  031726  012704  004670                  MOV     #BSFACT+10,R4   ;SET R4 TO POINT TO FACTORY TABLE
 3931  031732  104401  015766          6$:     TYPE    ,COLHD          ;TYPE COLUMN HEADER
 3932  031736  104401  016023          3$:     TYPE    ,SPACE4         ;SET FORMAT OF REPORT
 3933  031742  012446                          MOV     (R4)+,-(SP)     ;GET FIRST CYLINDER WORD
 3934  031744  100417                          BMI     4$              ;IF ALL 1'S EXIT THIS LOOP
 3935  031746  104402                          TYPOC                   ;ELSE TYPE VALUE
 3936  031750  104401  012744                  TYPE    ,SPACE2         ;FORMAT
 3937  031754  105724                          TSTB    (R4)+           ;BUMP R4 TO THE TRACK IN TRK/SEC WORD
 3938  031756  111446                          MOVB    (R4),-(SP)      ;PUT TRACK ON THE STACK
 3939  031760  104402                          TYPOC                   ;TYPE TRACK NUMBER
 3940  031762  104401  012744                  TYPE    ,SPACE2         ;FORMAT
 3941  031766  114446                          MOVB    -(R4),-(SP)     ;PUT SECTOR ON STACK
 3942  031770  104402                          TYPOC                   ;TYPE SECTOR NUMBER
 3943  031772  104401  001267                  TYPE    ,$CRLF          ;RETURN THE CARRIAGE
 3944  031776  005724                          TST     (R4)+           ;PUT R4 AT NEXT FILE ENTRY
 3945  032000  005200                          INC     R0              ;COUNT BAD SECTORS
 3946  032002  000755                          BR      3$              ;LOOP
 3947  032004  005726                  4$:     TST     (SP)+           ;CLEAN OFF STACK
 3948  032006  005737  001242                  TST     $TMP2           ;TEST IF 2ND PASS, IF YES
```

```
 3949  032012  001016                        BNE     5$                ;EXIT
 3950  032014  052737  000001  001242        BIS     #1,$TMP2          ;ELSE SET 2ND PASS SWITCH
 3951  032022  012704  003670                MOV     #BSSOFT+10,R4     ;SWITCH TO OTHER TABLE
 3952  032026  005700                        TST     R0
 3953  032030  001004                        BNE     7$
 3954  032032  104401  001267                TYPE    ,$CRLF
 3955  032036  104401  015751                TYPE    ,NONE
 3956  032042  104401  015706        7$:     TYPE    ,SBSLAB           ;TYPE SOFTWARE LABEL
 3957  032046  000731                        BR      6$                ;GO BACK TO LOOP
 3958  032050  104401  001267        5$:     TYPE    ,$CRLF
 3959  032054  005700                        TST     R0
 3960  032056  001405                        BEQ     9$
 3961  032060  010046                        MOV     R0,-(SP)          ;EXIT ROUTINE - TYPE THE
 3962  032062  104402                        TYPOC                     ;NUMBER OF BAD SECTORS
 3963  032064  104401  016034                TYPE    ,TOTMES           ;AND RETURN
 3964  032070  000402                        BR      8$
 3965  032072  104401  015751        9$:     TYPE    ,NONE
 3966  032076  104412                8$:     RESREG
 3967  032100  000207                        RTS     PC
```

```
                                .SBTTL  RK611/RK06 UNIBUS DRIVER FOR SEQUENTIAL OPERATIONS (REV. 0.08)

                                ;*COPYRIGHT (C) 1975
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MA. 01754
                                ;*AUTHOR: ROY SPITZER

                                .SBTTL   *WATCH-DOG TIMER

                                ;;*****************************************************************
                                ;*
                                ;*      THE WATCH-DOG TIMER DOES A PSEUDO-TIMING OF RK06 UNIBUS
                                ;*      SUBSYSTEM COMMAND. SINCE ONE CAN NOT GUARANTEE THAT A
                                ;*      REAL-TIME CLOCK (KW11-P OR KW11-L) IS ON THE SYSTEM
                                ;*      THE RK06 DRIVER WILL USE THE LOCATION W.MTIM FOR
                                ;*      MILLI-SECOND TIMING. WHEN W.MTIM REACHES ZERO THE
                                ;*      WATCH-DOG TIMER WILL SCAN THE DRIVES IN USE AS
                                ;*      DETERMINED BY THE LOCATION W.TIME. THE TIMER COUNTS
                                ;*      (ONE FOR EACH DRIVE) ARE KEPT IN THE TABLE W.DRV.
                                ;*      IF ANY COUNT IN THE TABLE W.DRV REACHES ZERO A COMMAND
                                ;*      TIME-OUT WILL BE DESIGNATED IN THE PROGRAM DEVICE STATUS
                                ;*      REGISTER OF THAT DRIVE'S PARAMETER BLOCK.
                                ;*
                                ;*      THE DRIVER WILL USE THE LOCATION W.MIN AS THE NUMBER
                                ;*      OF MILLISECONDS FOR AN UNLOAD OR START SPINDLE COMMAND.
                                ;*      THE DRIVER WILL USE THE LOCATION W.SEC AS THE TIME
                                ;*      LIMIT FOR ALL OTHER COMMANDS.
                                ;*
                                ;*      FOR QUEUED OPERATIONS THE WATCH-DOG TIMER WILL
                                ;*      WATCH UP TO 8 OPERATIONS SIMULATEOUSLY. FOR SEQUENTIAL
                                ;*      OPERATIONS ONLY ONE OPERATION WILL BE WATCHED.
                                ;*
                                ;*CALL   JSR     PC,W.WTCH
                                ;*      RETURN          IF NO DRIVE ORDER EXCEEDED ITS TIME LIMIT
                                ;*
                                ;*      OTHERWISE AN ABNORMAL RETURN TO THE ROUTINE ADDRESS
                                ;*      BY LOCATION A.ABNL WILL OCCUR AND THE CMDTO FLAG
                                ;*      IN THE PROGRAM DEVICE STATUS REGISTER OF THE
                                ;*      APPROPRIATE PARAMETER BLOCK WILL BE SET.
                                ;*
                                ;;*****************************************************************

032102  010546          W.WTCH: MOV     R5,-(SP)        ;SAVE R5 ON THE STACK
032104  010446                  MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
032106  010346                  MOV     R3,-(SP)        ;SAVE R3 ON THE STACK
032110  010246                  MOV     R2,-(SP)        ;SAVE R2 ON STACK
032112  013746  177776          MOV     PS,-(SP)        ;SAVE PROGRAM STATUS WORD ON STACK
032116  005337  002610          DEC     W.MTIM          ;DECREMENT MILLISECOND TIMER
032122  001034                  BNE     20$             ;IF NOT ZERO RETURN
032124  013737  002612  002610  MOV     W.MILI,W.MTIM   ;REINITIALIZE MILLISECOND TIMER
032132  105737  002632          TSTB    W.TIME          ;CHECK IF DRIVE IS BEING TIMED
032136  001426                  BEQ     20$             ;NO, RETURN
032140  013737  002574  177776  MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
032146  013702  002570          MOV     RKBAS,R2        ;LOAD BASE OF RK06 REGISTERS
032152  005337  002646          DEC     W.DRV           ;DECREMENT COMMAND TIMER
032156  001016                  BNE     20$             ;RETURN IF NO TIME OUT
```

```
 4024  032160  105037  002632                 CLRB    W.TIME          ;RESET TIMING INDICATOR
 4025  032164  013705  002644                 MOV     P3LKT,R5        ;LOAD ADDRESS OF PARAMETER BLOCK
 4026                                                                  ;   TABLE FOR INDEXING
 4027  032170  052765  000100  000014         BIS     #CMDTO,P.PRST(R5) ;SET COMMAND TIME OUT
 4028  032176  020537  002606                 CMP     R5,0.WAIT       ;CHECK IF DRIVER IS WAITING FOR
 4029                                                                  ;   COMMAND COMPLETION
 4030  032202  001002                         BNE     5$              ;NO, DO NOT ALTER WAITING FOR
 4031                                                                  ;   COMMAND COMPLETION
 4032  032204  005037  002606                 CLR     0.WAIT          ;CLEAR WAIT FOR COMMAND COMPLETION
 4033  032210  004737  035464          5$:    JSR     PC,R.ABNL       ;BRANCH TO ERROR ROUTINE
 4034  032214  012637  177776          20$:   MOV     (SP)+,PS        ;RESTORE PSW
 4035  032220  012602                         MOV     (SP)+,R2        ;RESTORE R2
 4036  032222  012603                         MOV     (SP)+,R3        ;RESTORE R3
 4037  032224  012604                         MOV     (SP)+,R4        ;RESTORE R4
 4038  032226  012605                         MOV     (SP)+,R5        ;RESTORE R5
 4039  032230  000207                         RTS     PC              ;RETURN
```

```
                                .SBTTL   *RK06 INTERRUPT SERVICE ROUTINE

                                ;;*****************************************************************
                                ;*
                                ;*      THIS ROUTINE WILL SERVICE ALL RK06 INTERRUPTS.
                                ;*
                                ;*      UPON RECEIVING AN INTERRUPT, THIS ROUTINE WILL
                                ;*      PERFORM ONE OF THE FOLLOWING SERVICES:
                                ;*
                                ;*      1.) SERVICE PORT WAS SEIZED BY OTHER PORT
                                ;*      2.) SERVICE DRIVER IS WAIT FOR COMMAND COMPLETION
                                ;*      3.) SERVICE POSITIONING COMPLETION
                                ;*      4.) REQUEUE COMMAND IF DRIVE WAS RELEASED
                                ;*          FOR THE QUEUED RK06 DRIVER.
                                ;*      5.) IF NO SERVICE IS REQUIRED, THE COMMAND WILL BE ISSUED
                                ;*          FOR THE QUEUED RK06 DRIVER.
                                ;*
                                ;*      THREE LINKS ARE PROVIDED TO THE DRIVING PROGRAM.
                                ;*      THEY ARE:
                                ;*
                                ;*      1.) A.NORM ADDRESS OF NORMAL RETURN (SUCESSFUL COMPLETION OF COMMAND)
                                ;*      2.) A.ABNL ADDRESS OF ABNORMAL RETURN (UNSUCESSFUL COMPLETION OF COMMAND)
                                ;*      3.) A.CONT ADDRESS OF CONTROL ERROR RETURN
                                ;*
                                ;*      FOR NORMAL AND ABNORMAL RETURNS, THE ADDRESS OF THE APPROPRIATE
                                ;*      PARAMETER BLOCK WILL BE IN R5.
                                ;*
                                ;*      FOR THE CONTROLLER ERROR RETURN, THE LOCATION E.CONT CONTAINS
                                ;*      THE REASON FOR THE CONTROLLER ERROR.
                                ;*
                                ;*      ROUTINES USED:
                                ;*              C.OPT  (QUEUED ONLY)
                                ;*              Q.PUSH (QUEUED ONLY)
                                ;*              Q.RMOV (QUEUED ONLY)
                                ;*              R.CONT (SEQUENTIAL ONLY)
                                ;*              R.NORM (SEQUENTIAL ONLY)
                                ;*              R.ABNL (SEQUENTIAL ONLY)
                                ;*              I.CSTS
                                ;*              I.STAT
                                ;*              I.ISSU
                                ;*              I.CCLR
                                ;*
                                ;;*****************************************************************

032232  010546                  I.INTR: MOV     R5,-(SP)        ;STORE R5 ON THE STACK
032234  010446                          MOV     R4,-(SP)        ;STORE R4 ON THE STACK
032236  010346                          MOV     R3,-(SP)        ;STORE R3 ON THE STACK
032240  010246                          MOV     R2,-(SP)        ;STORE R2 ON THE STACK
032242  010146                          MOV     R1,-(SP)        ;STORE R1 ON THE STACK
032244  010046                          MOV     R0,-(SP)        ;STORE R0 ON THE STACK
032246  013702  002570                  MOV     RKBAS,R2        ;LOAD R2 TO ADDRESS RK06 REGISTER
032252  016237  000010  002534          MOV     RKCS2(R2),T.CS2 ;STORE CS2
032260  032737  001000  002534          BIT     #MDS,T.CS2      ;CHECK IF MULTIPLE DRIVE SELECT
032266  001407                          BEQ     1$              ;NO, CONTINUE PROCESSING
032270  052737  100000  002604          BIS     #E.MDS,E.CONT   ;SET MULTIPLE DRIVE SELECT
032276  004737  035510                  JSR     PC,R.CONT       ;REPORT ERROR
```

```
4096  032302  000137  034462                    JMP     I.RTRN          ;RETURN
4097
4098  032306  105737  002626          1$:       TSTB    I.ISRL          ;CHECK IF INTERRUPT OR RELEASE
4099  032312  001410                             BEQ     6$              ;NO, CHECK IF DRIVE AVAILABLE
4100  032314  100403                             BMI     5$              ;CHECK IF RELEASE COMMAND
4101  032316  105037  002626                     CLRB    I.ISRL          ;YES,CLEAR FLAG
4102  032322  000473                             BR      I.IOO           ;CONTINUE PROCESSING INTERRUPT
4103
4104  032324  105037  002626          5$:       CLRB    I.ISRL          ;CLEAR FLAG
4105  032330  000137  033444                     JMP     I.ATTN          ;GO PROCESS DRIVE ATTENTIONS
4106
4107  032334  032737  010400  002534  6$:       BIT     #NED!UFE,T.CS2  ;CHECK FOR NON-EXISTENT DRIVE OR
4108                                                                    ;  UNIT FIELD ERROR
4109  032342  001413                             BEQ     7$              ;NO, WAIT FOR DUAL ACCESS INTERRUPT
4110  032344  013704  002534                     MOV     T.CS2,R4        ;LOAD R4 FOR DRIVE NUMBER
4111  032350  042704  177770                     BIC     #^C<DRVMSK>,R4  ;KEEP DRIVE BITS
4112  032354  013705  002644                     MOV     PBLKT,R5        ;STORE PARAMETER BLOCK ADDRESS
4113  032360  016237  000000  002532             MOV     RKCS1(R2),T.CS1 ;LOAD TEMPORARY CS1 FOR STATUS REPORT
4114  032366  000137  032654                     JMP     I.ERRC          ;REPORT ERROR
4115
4116  032372  016237  000012  002552  7$:       MOV     RKDS(R2),T.DS   ;STORE STATUS REGISTER FOR COMPARISON
4117  032400  032737  000001  002552             BIT     #DRA,T.DS       ;CHECK IF DRIVE SEIZED BY OTHER
4118                                                                    ;  PORT
4119  032406  001041                             BNE     I.IOO           ;NO, CONTINUE PROCESSING INTERRUPT
4120
4121                                             ;CHECK IF ANY DATA TRANSFER ERROR EXISTS
4122  032410  032737  164000  002534             BIT     #DLT!WCE!UPE!NEM,T.CS2
4123
4124  032416  001007                             BNE     10$             ;INDICATE ERROR
4125  032420  016237  000014  002550             MOV     RKER(R2),T.ER   ;STORE ERROR REGISTER
4126
4127                                  ;          CHECK FOR DATA TRANSFER ERROR TYPE ERROR
4128  032426  032737  125700  002550             BIT     #DCK!OPI!WLE!COE!HVRC!BSE!ECH,T.ER
4129
4130  032434  001407                             BEQ     11$             ;NO, WAIT FOR RELEASE OF RK06 DRIVE
4131
4132  032436  052737  000010  002604  10$:       BIS     #E.UDAT,E.CONT  ;SET UNEXPECTED DATA TYPE ERROR
4133  032444  004737  035510                     JSR     PC,R.CONT       ;REPORT ERROR
4134  032450  000137  034462                     JMP     I.RTRN          ;RESTORE REGISTERS
4135
4136  032454  105037  002632          11$:      CLRB    W.TIME          ;RESET TIMING ON THIS DRIVE
4137  032460  005037  002646                     CLR     W.DRV           ;CLEAR TIMING COUNT FOR THIS DRIVE
4138  032464  013705  002644                     MOV     PBLKT,R5        ;LOAD R5 WITH PARAMETER BLOCK
4139                                                                    ;  ADDRESS
4140  032470  052765  010000  000014             BIS     #DRVSZD,P.PRST(R5) ;SET DRIVE SEIZED IN THE
4141                                                                    ;  PROGRAM DRIVE STATUS REGISTER
4142  032476  005037  002606                     CLR     O.WAIT          ;CLEAR WAIT FOR COMMAND COMPLETION
4143  032502  004737  035464                     JSR     PC,R.ABNL       ;INDICATE ABNORMAL TERMINATION
4144  032506  000137  034462                     JMP     I.RTRN          ;GO RESTORE REGISTERS
4145
4146  032512  013705  002606          I.IOO:    MOV     O.WAIT,R5       ;LOAD PARAMETER BLOCK ADDRESS INTO R5
4147  032516  001002                             BNE     2$              ;IS COMMAND WAITING PROCESSING
4148                                                                    ;  YES, DO PROCESSING
4149  032520  000137  033444                     JMP     I.ATTN          ;NO, PROCESS ATTENTION
4150
4151  032524  013704  002534          2$:       MOV     T.CS2,R4        ;STORE RKCS2 FOR DRIVE NUMBER
```

```
4152  032530  042704  177770                    BIC     #^C<DRVMSK>,R4  ;MASK OUT UNNECESSARY BITS
4153
4154
4155  032534  126504  000000                    CMPB    P.DRVN(R5),R4   ;CHECK IF DRIVE NUMBER IS EXPECTED
4156  032540  001401                            BEQ     3$              ;YES, CONTINUE
4157  032542  000000                            HALT                    ;NO, DRIVER ERROR
4158  032544  122765  000164  000001  3$:     CMPB    #RDALHD,P.CMND(R5) ;CHECK IF READ ALL HEADERS
4159  032552  001002                            BNE     10$             ;NO, EXECUTE NORMAL DATA TRANSFER
4160  032554  000137  033112                    JMP     I.HDAL          ;GO EXECUTE SPECIAL HEADER SEQUENCE
4161
4162  032560  005037  002606          10$:     CLR     O.WAIT          ;CLEAR WAIT FOR COMMAND COMPLETION
4163  032564  005037  002646                    CLR     W.DRV           ;CLEAR WATCH-DOG TIME
4164  032570  105037  002632                    CLRB    W.TIME          ;RESET TIMING ON THIS DRIVE
4165  032574  016237  000000  002532            MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REGISTER 1
4166  032602  032737  100000  002532            BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR
4167  032610  001021                            BNE     I.ERRC          ;YES, PROCESS ERROR
4168  032612  016237  000016  002546            MOV     RKASOF(R2),T.ASOF ;STORE ATTENTION SUMMARY
4169  032620  133737  002633  002547            BITB    INTMSK,T.ASOF+1 ;CHECK IF DRIVE ATTENTION SET
4170  032626  001004                            BNE     15$             ;YES, REPORT ERROR
4171  032630  004737  035476                    JSR     PC,R NORM       ;INDICATE NORMAL RETURN
4172  032634  000137  034462                    JMP     I.RTRN          ;RESTORE REGISTERS
4173
4174  032640  052765  000010  000014  15$:     BIS     #UEXATT,P.PRST(R5) ;SET UNEXPECTED ATTENTION
4175
4176  032646  004737  035132          I.ERRA:  JSR     PC,I.CSTS       ;STORE CONTROLLER STATUS
4177  032652  000405                            BR      I.ERR           ;STORE PATTERN AND POSITION INFORMATION
4178
4179  032654  013765  002532  000016  I.ERRC:  MOV     T.CS1,P.CS1(R5) ;GET ERROR RKCS1
4180  032662  004737  035154                    JSR     PC,I.CST1       ;GET REST OF CONTROLLER STATUS
4181  032666  016265  000032  000062  I.ERR:   MOV     RKECPT(R2),P.EPAT(R5) ;STORE ECC PATTERN
4182  032674  016265  000030  000060           MOV     RKECPS(R2),P.EPOS(R5) ;STORE ECC POSITION
4183  032702  004037  034500                    JSR     R0,I.CCLR       ;CLEAR CONTROLLER
4184  032706  034462                            I.RTRN                  ;ERROR RETURN
4185  032710  032765  010400  000020            BIT     #NED!UFE,P.CS2(R5) ;CHECK IF IT WAS NON-EXISTENT DRIVE OR
4186                                                                    ;  UNIT FIELD ERROR
4187  032716  001046                            BNE     5$              ;YES, REPORT ERROR
4188  032720  004037  035236                    JSR     R0,I.STAT       ;GATHER DRIVE STATUS
4189  032724  034462                            I.RTRN                  ;ERROR RETURN
4190  032726  112737  000005  002532            MOVB    #DR.CLR,T.CS1   ;LOAD COMMAND
4191  032734  004037  034562                    JSR     R0,I.ISSU       ;ISSUE DRIVE CLEAR
4192  032740  034462                            I.RTRN                  ;ERROR RETURN
4193  032742  133737  002633  002547            BITB    INTMSK,T.ASOF+1 ;CHECK IF ATTENTION RESET
4194  032750  001407                            BEQ     2$              ;NO, INDICATE DRIVE ERROR
4195  032752  052737  000020  002604            BIS     #E.CLAT,E.CONT  ;SET ATTENTION DID NOT RESET
4196                                                                    ;  WITH CLEAR
4197  032760  004737  035510                    JSR     PC,R.CONT       ;REPORT CONTROLLER ERROR
4198  032764  000137  034462                    JMP     I.RTRN          ;GO RESTORE REGISTERS
4199
4200  032770  032737  040000  002556  2$:      BIT     #S.DSC,T.MR2    ;CHECK IF DRIVE STATUS CHANGE CLEARED
4201  032776  001403                            BEQ     3$              ;YES, CHECK FAULT
4202  033000  052765  000040  000014           BIS     #DRVDSC,P.PRST(R5) ;SET DSC DID NOT CLEAR
4203  033006  032737  001000  002560  3$:      BIT     #S.PAR,T.MR3    ;CHECK IF DRIVE PARITY ERROR
4204  033014  001407                            BEQ     5$              ;NO, INDICATE ABNORMAL TERMINATION
4205  033016  052737  002000  002604           BIS     #E.DPAR,E.CONT  ;SET DRIVE PARITY ERROR
4206  033024  004737  035510                    JSR     PC,R.CONT       ;INDICATE CONTROLLER ERROR
4207  033030  000137  034462                    JMP     I.RTRN          ;RETURN
```

```
4208
4209  033034  032765  000020  000014  5$:     BIT     #DRVHRD,P.PRS1(R5) ;CHECK IF HARD DRIVE ERROR
4210  033042  001017                          BNE     10$                ;YES, GO REPORT ERROR
4211  033044  032737  020000  002556          BIT     #S.PIP,T.MR2       ;CHECK IF DRIVE IS CYCLING DOWN
4212  033052  001413                          BEQ     10$                ;NO, REPORT ERROR
4213  033054  052765  020000  000014          BIS     #E.UNLD,P.PRST(R5) ;SET DRIVE UNLOADING
4214  033062  113737  002633  002632          MOVB    INTMSK,W.TIME      ;SET UP 8 SECONDS FOR DRIVE TO CYCLE UP
4215  033070  013737  002616  002646          MOV     W.BSEC,W.DRV
4216  033076  000137  034462                  JMP     I.RTRN             ;GO RESTORE REGISTERS
4217
4218  033102  004737  035464          10$:    JSR     PC,R.ABNL          ;GO REPORT ERROR
4219  033106  000137  034462                  JMP     I.RTRN             ;GO RESTORE REGISTERS
4220
4221                                  .SBTTL   *READ ALL HEADERS INTERRUPT SEQUENCE
4222
4223  033112  016237  000000  002532  I.HDAL: MOV     RKCS1(R2),T.CS1 ;STORE CS1 TO CHECK CONTROLLER
4224                                                                  ;  ERROR
4225  033120  032737  100000  002532          BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR
4226  033126  001422                          BEQ     5$              ;NO, CHECK FOR ATTENTION
4227
4228  033130  005037  002606                  CLR     0.WAIT          ;CLEAR WAITING FOR COMMAND COMPLETE
4229  033134  105037  002632                  CLRB    W.TIME          ;RESET TIMING ON DRIVE
4230  033140  005037  002646                  CLR     W.DRV           ;CLEAR TIME OUT COUNT
4231  033144  013765  002532  000016          MOV     T.CS1,P.CS1(R5) ;STORE ERROR RKCS1
4232  033152  004737  035154                  JSR     PC,I.CST1       ;STORE CONTROLLER REGISTERS
4233  033156  004037  034500                  JSR     R0,I.CCLR       ;CLEAR CONTROLLER
4234  033162  034462                          I.RTRN                  ;ERROR RETURN
4235  033164  004737  035464                  JSR     PC,R.ABNL       ;INDICATE ERROR RETURN
4236  033170  000137  034462                  JMP     I.RTRN          ;RESTORE REGISTERS
4237
4238  033174  016537  000016  002546  5$:     MOV     RKASOF(R5),T.ASOF ;STORE ATTENTION SUMMARY
4239  033202  133737  002633  002547          BITB    INTMSK,T.ASOF+1 ;CHECK IF DRIVE ATTENTION IS SET
4240  033210  001410                          BEQ     7$              ;NO, CHECK IF READ ALL HEADERS
4241  033212  005037  002606                  CLR     0.WAIT          ;CLEAR WAITING FOR COMMAND COMPLETION
4242  033216  105037  002632                  CLRB    W.TIME          ;RESET TIMING ON DRIVE
4243  033222  005037  002646                  CLR     W.DRV           ;CLEAR TIME OUT COUNT
4244  033226  000137  032646                  JMP     I.ERRA          ;GO REPORT ERROR
4245
4246  033232  013701  002622          7$:     MOV     HDR.AD,R1       ;GET MAIN MEMORY ADDRESS
4247  033236  016221  000024                  MOV     RKDB(R2),(R1)+  ;GET FIRST WORD OF HEADER
4248  033242  016221  000024                  MOV     RKDB(R2),(R1)+  ;GET SECOND WORD OF HEADER
4249  033246  016221  000024                  MOV     RKDB(R2),(R1)+  ;GET THIRD WORD OF HEADER
4250  033252  010137  002622                  MOV     R1,HDR.AD       ;STORE ADDRESS FOR NEXT HEADER
4251  033256  016237  000010  002534          MOV     RKCS2(R2),T.CS2 ;STORE CS2 TO CHECK FOR DATA LATE
4252  033264  032737  100000  002534          BIT     #DLT,T.CS2      ;CHECK FOR DATA LATE
4253  033272  001055                          BNE     35$             ;YES, REPORT ERROR
4254  033274  005337  002624                  DEC     HDR.CT          ;DECREMENT NUMBER OF HEADER YET TO READ
4255  033300  001026                          BNE     25$             ;IF NON-ZERO, GO ISSUE NEXT READ HEADER
4256  033302  005037  002606                  CLR     0.WAIT          ;CLEAR DRIVER WAITING FOR COMMAND COMPLETION
4257  033306  005037  002646                  CLR     W.DRV           ;CLEAR TIME OUT COUNT FOR THIS DRIVE
4258  033312  105037  002632                  CLRB    W.TIME          ;CLEAR WATCH DOG TIME ON THIS DRIVE
4259  033316  012762  000003  000022          MOV     #3,RKMR1(R2)    ;LOAD MAINTENANCE REGISTER FOR SECTOR COUNT
4260  033324  112737  000001  002532          MOVB    #DR.SEL,T.CS1   ;LOAD SELECT COMMAND
4261  033332  004037  034562                  JSR     R0,I.ISSU       ;GET SECTOR COUNT
4262  033336  034462                          I.RTRN                  ;ERROR RETURN
4263  033340  013765  002560  000056          MOV     T.MR3,P.B11(R5) ;LOAD SECTOR COUNT
```

```
4264  033346  004737  035476                  JSR     PC,R.NORM       ;INDICATE NORMAL TERMINATION
4265  033352  000137  034462                  JMP     I.RTRN          ;RESTORE REGISTERS
4266
4267  033356  016562  000002  000020  25$:    MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS REGISTER
4268  033364  016562  000004  000006          MOV     P.SECT(R5),RKDA(R2) ;LOAD SECTOR AND TRACK
4269  033372  116565  000007  000017          MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
4270  033400  042765  165777  000016          BIC     #^C<CDT!CFMT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT FORMAT AND
4271                                                  ;  DRIVE TYPE
4272  033406  112765  000125  000016          MOVB    #RDHEAD,P.CS1(R5)  ;STORE COMMAND ISSUED
4273  033414  016562  000016  000000          MOV     P.CS1(R5),RKCS1(R2)  ;ISSUE READ HEADER
4274  033422  000137  034462                  JMP     I.RTRN          ;RESTORE REGISTERS
4275
4276  033426  052737  000400  002604  35$:    BIS     #E.DLT,E.CONT   ;SET DATA LATE WHILE UNLOADING HEADER
4277  033434  004737  035510                  JSR     PC,R.CONT       ;REPORT ERROR
4278  033440  000137  034462                  JMP     I.RTF           ;RESTORE REGISTERS
4279
4280                                  .SBTTL   *DRIVE ATTENTION SCANNER
4281
4282  033444  016237  000000  002532  I.ATTN: MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS
4283                                                  ;  REGISTER 1 FOR COMPARISON
4284  033452  032737  100000  002532          BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR OCCURRED
4285  033460  001441                          BEQ     5$              ;NO, CHECK IF ATTENTION
4286
4287                                          ;CHECK IF ANY DATA TRANSFER TYPE ERROR EXISTS
4288  033462  032737  164000  002534          BIT     #DLT!WCE!UPE!NEM,T.CS2
4289
4290  033470  001007                          BNE     1$              ;INDICATE ERROR
4291  033472  016237  000014  002550          MOV     RKER(R2),T.ER   ;STORE ERROR REGISTER
4292
4293                                  ;       CHECK FOR DATA TRANSFER ERROR TYPE
4294  033500  032737  125700  002550          BIT     #DCK!OPI!WLE!COE!HVRC!BSE!ECH,T.ER
4295
4296  033506  001407                          BEQ     2$              ;NO DATA TRANSFER ERROR
4297
4298  033510  052737  000010  002604  1$:     BIS     #E.UDAT,E.CONT  ;SET UNEXPECTED DATA TYPE ERROR
4299  033516  004737  035510                  JSR     PC,R.CONT       ;REPORT ERROR
4300  033522  000137  034462                  JMP     I.RTRN          ;RESTORE REGISTERS
4301
4302  033526  013704  002534          2$:     MOV     T.CS2,R4        ;SAVE CS2 FOR REGISTER NUMBER
4303  033532  042704  177770                  BIC     #^C<DRVMSK>,R4  ;STRIP OFF JUNK
4304  033536  105037  002632                  CLRB    W.TIME          ;CLEAR WATCH DOG TIMER
4305  033542  005037  002646                  CLR     W.DRV           ;RESET TIMER VALUE
4306  033546  013705  002644                  MOV     PBLKT,R5        ;STORE PARAMETER BLOCK ADDRESS IN R5
4307
4308                                  ;       CLEAR DRIVE POSITIONING AND DRIVE POSITIONED FOR DATA TRANSFER
4309                                  ;         IN PROGRAM DEVICE STATUS REGISTER
4310  033552  042765  000006  000014          BIC     #DRVPOS!DRVPDT,P.PRST(R5)
4311
4312  033560  000137  032654                  JMP     I.ERRC          ;GO REPORT ERROR
4313
4314  033564  032737  040000  002532  5$:     BIT     #DI,T.CS1       ;CHECK IF ANY DRIVE ATTENTION
4315  033572  001002                          BNE     6$              ;YES, PROCESS INTERRUPT
4316  033574  000137  034462                  JMP     I.RTRN          ;RESTORE REGISTERS
4317
4318  033600  016237  000016  002546  6$:     MOV     RKASOF(R2),T.ASOF ;STORE ATTENTION SUMMARY
4319  033606  105737  002547                  TSTB    T.ASOF+1        ;CHECK IF ANY ATTENTIONS SET
```

```
 4320  033612  001007                        BNE     7$                      ;YES GO PROCESS INTERRUPT
 4321  033614  052737  000002  002604        BIS     #E.NOAT,E.CONT          ;SET NO ATTENTION IN ATTENTION SUMMARY
 4322  033622  004737  035510                JSR     PC,R.CONT               ;GO REPORT ERROR
 4323  033626  000137  034462                JMP     I.RTRN                  ;GO RESTORE REGISTERS
 4324
 4325  033632  133737  002633  002547  7$:   BITB    INTMSK,T.ASOF+1         ;CHECK IF DESIRED INTERRUPT
 4326  033640  001007                        BNE     8$                      ;YES, GO PROCESS IT
 4327  033642  052737  000004  002604        BIS     #E.UATT,E.CONT          ;SET UNSOLICATED ATTENTION
 4328  033650  004737  035510                JSR     PC,R.CONT               ;GO REPORT ERROR
 4329  033654  000137  034462                JMP     I.RTRN                  ;GO RESTORE REGISTERS
 4330
 4331  033660  013705  002644          8$:   MOV     PBLKT,R5                ;STORE PARAMETER BLOCK TABLE
 4332  033664  116504  000000                MOVB    P.DRVN(R5),R4           ;STORE DRIVE NUMBER
 4333  033670  032765  020000  000014        BIT     #E.UNLD,P.PRST(R5)  ;CHECK IF DRIVE UNLOADING
 4334  033676  001402                        BEQ     11$                     ;NO, CONTINUE
 4335  033700  000137  034362                JMP     I.UNLD                  ;SERVICE DRIVE IN POSITION AFTER ERROR
 4336
 4337  033704  042765  000002  000014  11$:   BIC     #DRVPOS,P.PRST(R5)  ;RESET DRIVE POSITIONING
 4338  033712  005062  000022                CLR     RKMR1(R2)               ;CLEAR MAINTENANCE REGISTER 1
 4339  033716  112737  000001  002532        MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
 4340  033724  004037  034562                JSR     R0,I.ISSU               ;SELECT DRIVE WITH ATTENTION HIGH
 4341  033730  034462                        I.RTRN                          ;ERROR RETURN
 4342  033732  013765  002560  000042        MOV     T.MR3,P.B00(R5)         ;STORE STATUS BYTE 00 MESS 8
 4343  033740  032765  000200  000042        BIT     #S.FLT,P.B00(R5)    ;CHECK IF DRIVE FAULT
 4344  033746  001401                        BEQ     12$                     ;NO, CHECK FOR DRIVE STATUS CHANGE
 4345  033750  000461                        BR      I.AERR                  ;PROCESS ERROR
 4346
 4347  033752  013765  002556  000040  12$:   MOV     T.MR2,P.A00(R5)         ;STORE MAINTENANCE REGISTER 2
 4348  033760  032765  040000  000040        BIT     #S.DSC,P.A00(R5)    ;CHECK FOR DRIVE STATUS CHANGE
 4349  033766  001004                        BNE     13$                     ;YES, PROCESS DRIVE STATUS CHANGE
 4350  033770  052765  004000  000014        BIS     #NODSC,P.PRST(R5)   ;SET NO DRIVE STATUS CHANGE
 4351  033776  000446                        BR      I.AERR                  ;PROCESS ERROR
 4352
 4353  034000  112737  000005  002532  13$:   MOVB    #DR.CLR,T.CS1           ;LOAD COMMAND
 4354  034006  004037  034562                JSR     R0,I.ISSU               ;CLEAR DRIVE STATUS CHANGE
 4355  034012  034462                        I.RTRN                          ;ERROR RETUN
 4356  034014  013765  002546  000032        MOV     T.ASOF,P.ASOF(R5)   ;STORE ATTENTION SUMMARY
 4357  034022  133765  002633  000033        BITB    INTMSK,P.ASOF+1(R5) ;CHECK IF ATTENTION RESET
 4358  034030  001407                        BEQ     15$                     ;YES, CONTINUE INTERRUPT PROCESSING
 4359  034032  052737  000020  002604        BIS     #E.CLAT,E.CONT          ;SET ATTENTION DID NOT RESET
 4360                                                                        ;  WITH DRIVE CLEAR
 4361  034040  004737  035510                JSR     PC,R.CONT               ;FLAG ERROR
 4362  034044  000137  034462                JMP     I.RTRN                  ;RESTORE REGISTERS
 4363
 4364  034050  013765  002556  000040  15$:   MOV     T.MR2,P.A00(R5)         ;STORE MAINTENANCE REGISTER 2
 4365  034056  032765  040000  000040        BIT     #S.DSC,P.A00(R5)    ;CHECK IF DRIVE STATUS CHANGE
 4366                                                                        ;  RESET
 4367  034064  001404                        BEQ     16$                     ;YES, CONTINUE INTERRUPT PROCESSING
 4368  034066  052765  000040  000014        BIS     #DRVDSC,P.PRST(R5)  ;SET DRIVE STATUS CHANGE DID NOT CLEAR
 4369  034074  000407                        BR      I.AERR                  ;GO PROCESS ERROR
 4370
 4371  034076  105037  002632          16$:   CLRB    W.TIME                  ;RESET TIMING ON THIS DRIVE
 4372  034102  005037  002646                CLR     W.DRV                   ;CLEAR DRIVE TIMING COUNT
 4373  034106  004737  035476                JSR     PC,R.NORM               ;REPORT SUCCESSFUL COMMAND COMPLETETION
 4374  034112  000563                        BR      I.RTRN                  ;RESTORE REGISTERS
 4375
```

```
                                        .SBTTL   *ATTENTION ERROR HANDLER

034114  042765  000004  000014  I.AERR: BIC     #DRVPDT,P.PRST(R5) ;RESET POSITIONING IN PROGRESS BECAUSE
                                                                   ;  OF DATA TRANSFER
034122  105037  002632                  CLRB    W.TIME             ;CLEAR TIMING FOR THIS DRIVE
034126  005037  002646                  CLR     W.DRV              ;RESET WATCH-DOG TIME
034132  042765  177741  000016          BIC     #177741,P.CS1(R5) ;KEEP COMMAND ISSUED
034140  042737  000036  002532          BIC     #36,T.CS1          ;KEEP CURRENT CONTROLLER STATUS
034146  053765  002532  000016          BIS     T.CS1,P.CS1(R5)    ;MAKE GOOD MESSAGE
034154  013765  002534  000020          MOV     T.CS2,P.CS2(R5)    ;STORE CONTROLLER REGISTERS
034162  013765  002536  000022          MOV     T.WCR,P.WCR(R5)
034170  013765  002540  000024          MOV     T.BA,P.BAR(R5)
034176  013765  002542  000026          MOV     T.DA,P.DTS(R5)
034204  013765  002544  000030          MOV     T.DC,P.DCYL(R5)
034212  013765  002546  000032          MOV     T.ASOF,P.ASOF(R5)
034220  013765  002550  000034          MOV     T.ER,P.ER(R5)
034226  013765  002552  000036          MOV     T.DS,P.DS(R5)
034234  004037  035236                  JSR     R0,I.STAT          ;GATHER DRIVE STATUS
034240  034462                          I.RTRN                     ;ERROR RETURN
034242  112737  000005  002532          MOVB    #DR.CLR,T.CS1      ;LOAD COMMAND
034250  004037  034562                  JSR     R0,I.ISSU          ;CLEAR DRIVE ERRORS
034254  034462                          I.RTRN                     ;ERROR RETURN
034256  133737  002633  002547          BITB    INTMSK,T.ASOF+1    ;CHECK IF ATTENTION RESET
034264  001407                          BEQ     2$                 ;YES, FLAG DRIVE ERROR
034266  052737  000020  002604          BIS     #E.CLAT,E.CONT     ;SET ATTENTION DID NOT RESET
034274  004737  035510                  JSR     PC,R.CONT          ;REPORT ERROR
034300  000137  034462                  JMP     I.RTRN             ;RESTORE REGISTERS

034304  032765  000020  000014  2$:     BIT     #DRVHRD,P.PRST(R5) ;CHECK IF AHARD DRIVE ERROR
034312  001017                          BNE     10$                ;YES, REPORT ERROR
034314  032737  020000  002556          BIT     #S.PIP,T.MR2       ;CHECK IF DRIVE IS UNLOADING
034322  001413                          BEQ     10$                ;NO, REPORT ERROR
034324  052765  020000  000014          BIS     #E.UNLD,P.PRST(R5) ;SET DRIVE UNLOADING DUE TO ERROR
034332  113737  002633  002632          MOVB    INTMSK,W.TIME      ;SET TIMING ON THIS DRIVE
034340  013737  002616  002646          MOV     W.8SEC,W.DRV       ;LOAD 8 SECONDS FOR CYCLE UP TIME
034346  000137  034462                  JMP     I.RTRN             ;RESTORE REGISTERS

034352  004737  035464          10$:    JSR     PC,R.ABNL          ;REPORT ERROR
034356  000137  034462                  JMP     I.RTRN             ;RESTORE REGISTERS

                                        .SBTTL   *ERROR CAUSING DRIVE TO UNLOAD

034362  052765  020000  000014  I.UNLD: BIS     #E.UNLD,P.PRST(R5) ;CLEAR DRIVE UNLOADING BECAUSE OF ERROR
034370  112737  000005  002532          MOVB    #DR.CLR,T.CS1      ;LOAD IN DRIVE CLEAR
034376  004037  034562                  JSR     R0,I.ISSU          ;GO ISSUE DRIVE CLEAR
034402  034462                          I.RTRN                     ;ERROR RETURN
034404  136437  002633  002547          BITB    INTMSK(R4),T.ASOF+1 ;CHECK IF ATTENTION CLEARED
034412  001406                          BEQ     15$                ;YES, CONTINUE
034414  012737  000020  002604          MOV     #E.CLAT,E.CONT     ;SET ATTENTION DID NOT RESET
034422  004737  035510                  JSR     PC,R.CONT          ;REPORT ERROR
034426  000415                          BR      I.RTRN             ;RESTORE REGISTERS

034430  032737  040000  002556  15$:    BIT     #S.DSC,T.MR2       ;CHECK IF DRIVE STAUS CHANGE RESET
034436  001403                          BEQ     20$                ;YES, CONTINUE
034440  052765  000040  000014          BIS     #DRVDSC,P.PRST(R5) ;SET DRIVE STAUS CHANGE DID NOT CLEAR
034446  105037  002632          20$:    CLRB    W.TIME             ;RESET TIMING ON THIS DRIVE
```

```
 4432  034452  005037  002646                CLR     W.DRV           ;CLEAR TIME COUNT
 4433  034456  004737  035464                JSR     PC,R.ABNL       ;REPORT ERROR
 4434
 4435  034462  012600              I.RTRN:   MOV     (SP)+,R0        ;RESTORE R0
 4436  034464  012601                        MOV     (SP)+,R1        ;RESTORE R1
 4437  034466  012602                        MOV     (SP)+,R2        ;RESTORE R2
 4438  034470  012603                        MOV     (SP)+,R3        ;RESTORE R3
 4439  034472  012604                        MOV     (SP)+,R4        ;RESTORE R4
 4440  034474  012605                        MOV     (SP)+,R5        ;RESTORE R5
 4441  034476  000002                        RTI                     ;RETURN
 4442
```

```
                        .SBTTL   *CONTROLLER CLEAR ROUTINE

                        ;;*****************************************************************
                        ;*
                        ;*      THIS ROUTINE WILL BE USED BY THE DRIVER TO CLEAR THE CONTROLLER
                        ;*      AND CHECK IF THE CONTROLLER ERRORS ARE RESET.  IF THE ERROR IS NOT
                        ;*      CLEARED, THE ROUTINE AS SPECIFIED IN A.CONT WILL BE CALLED WITH
                        ;*      E.CCLR SET IN E.CONT.
                        ;*
                        ;*      REGISTER        USE
                        ;*      --------        ---
                        ;*
                        ;*      R2              ADDRESS OF RK06 REGISTERS
                        ;*      R5              ADDRESS OF PARAMETER BLOCK
                        ;*
                        ;*CALL  JSR     R0,I.CCLR
                        ;*      (ADDRESS OF ERROR RETURN)
                        ;*      RETURN
                        ;*
                        ;;*****************************************************************

034500  012762  100000  000000  I.CCLR: MOV     #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
034506  016237  000000  002532          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REGISTER 1
034514  032737  100000  002532          BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER CLEAR DID
                                                                ;  CLEAR ERROR
034522  001407                          BEQ     5$              ;YES, RETURN TO DRIVER PROCESSING
034524  052737  000001  002604          BIS     #E.CCLR,E.CONT  ;SET CLEAR CONTROLLER DID NOT CLEAR ERROR
034532  004737  035510                  JSR     PC,R.CONT       ;REPORT CONTROLLER ERROR
034536  011000                          MOV     (R0),R0         ;SET UP ERROR RETURN
034540  000200                          RTS     R0              ;RETURN

034542  012762  000100  000000  5$:     MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
034550  112737  177777  002626          MOVB    #-1,I.ISRL      ;SET INTERRUPT ENABLE ISSUED
034556  005720                          TST     (R0)+           ;ADJUST FOR NORMAL RETURN
034560  000200                          RTS     R0              ;RETURN
```

```
                                .SBTTL   *COMMAND ISSUED BY DRIVER SERVICE ROUTINE

                                ;;*****************************************************************
                                ;*
                                ;*      THIS ROUTINE WILL ISSUE THE COMMAND AS SPECIFIED IN T.CS1
                                ;*      AND CHECK IF A CONTROLLER ERROR OCCURRED.  IF A CONTROLLER
                                ;*      ERROR OCCURRED, E.CERR WILL BE SET IN E.CONT AND
                                ;*      CONTROL WILL BE TURN OVER TO THE ROUTINE SPECIFIED BY THE
                                ;*      ADDRESS IN A.CONT.
                                ;*
                                ;*      REGISTER        USE
                                ;*      --------        ---
                                ;*
                                ;*      R2              ADDDRESS OF RK06 REGISTERS
                                ;*      R5              ADDRESS OF PARAMETER BLOCK
                                ;*
                                ;*CALL  JSR     R0,I.ISSU
                                ;*      <ADDRESS OF ERROR RETURN>
                                ;*      RETURN
                                ;*
                                ;*      ROUTINES USED:
                                ;*      --------------
                                ;*
                                ;*              I.CCLR
                                ;*              I.STOR
                                ;*
                                ;;*****************************************************************

034562  013746  002532          I.ISSU: MOV     T.CS1,-(SP)     ;STORE COMMAND ISSUED
034566  005037  002534                  CLR     T.CS2           ;CLEAR TEMPORARY CS2
034572  116537  000000  002534          MOVB    P.DRVN(R5),T.CS2 ;LOAD IN DRIVE NUMBER
034600  013762  002534  000010          MOV     T.CS2,RKCS2(R2) ;LOAD DRIVE NUMBER FOR COMMAND
034606  116537  000007  002533          MOVB    P.CS1H(R5),T.CS1+1 ;STORE BITS 8-15 OF CS1
034614  142737  177753  002533          BICB    #^C<B.CDT!B.CFMT>,T.CS1+1 ;CLEAR ALL BITS EXCEPT
                                                                ;  FORMAT AND DRIVE TYPE
034622  013762  002532  000000          MOV     T.CS1,RKCS1(R2) ;ISSUE COMMAND
034630  105762  000000          1$:     TSTB    RKCS1(R2)       ;WAIT FOR READY
034634  100375                  BPL     1$
034636  004737  035004          JSR     PC,I.STOR       ;GO STORE REGISTERS
034642  032737  100000  002532  BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR OCCUERED
034650  001437                  BEQ     5$              ;NO, RETURN
034652  032737  001000  002534  BIT     #MDS,T.CS2      ;CHECK IF MULTIPLE DRIVE SELECT
034660  001406                  BEQ     2$              ;NO, CHECK FOR OTHER CONTROLLER ERRORS
034662  052737  100000  002604  BIS     #E.MDS,E.CONT   ;SET MULTIPLE DRIVE SELECT FLAG
034670  004737  035510          JSR     PC,R.CONT       ;REPORT CONTROLLER ERROR
034674  000440                  BR      10$             ;RETURN

                                ;CHECK IF ANY CONTROLLER ERROR IS SET
034676  032737  024000  002532  2$:     BIT     #CTO!SPAR,T.CS1
034704  001027                  BNE     7$
034706  032737  176400  002534  BIT     #UFE!PGE!NEM!NED!UPE!WCE!DLT,T.CS2
034714  001023                  BNE     7$
034716  032737  131761  002550  BIT     #ILC!DTYE!FMTE!ECH!BSE!HVRC!COE!DTE!OPI!DCK,T.ER
034724  001017                  BNE     7$

034726  122716  000005          CMPB    #DR.CLR,(SP)    ;CHECK IF CLEAR DRIVE
```

```
 4534  034732  001003                        BNE     3$                  ;NO, DO NOT SET DRIVE HARD ERROR
 4535  034734  052765  000020  000014        BIS     #DRVHRD,P.PRST(R5)  ;SET HARD DRIVE ERROR
 4536  034742  004037  034500          3$:   JSR     R0,I.CCLR           ;GO ISSUE A CONTROLLER CLEAR
 4537  034746  034776                        10$                         ;ERROR RETURN
 4538  034750  012762  000100  000000  5$:   MOV     #IE,RKCS1(R2)       ;SET INTERRUPT ENABLE
 4539  034756  005726                        TST     (SP)+               ;ADJUST STACK
 4540  034760  005720                        TST     (R0)+               ;ADJUST R0 FOR NORMAL RETURN
 4541  034762  000200                        RTS     R0                  ;RETURN
 4542
 4543  034764  052737  001000  002604  7$:   BIS     #E.CERR,E.CONT      ;SET CONTROLLER ERROR DURING
 4544                                                                    ;  DRIVER SERVICING
 4545  034772  004737  035510                JSR     PC,R.CONT           ;REPORT ERROR
 4546  034776  005726                  10$:  TST     (SP)+               ;ADJUST STACK
 4547  035000  011000                        MOV     (R0),R0             ;ADJUST R0 FOR ERROR RETURN
 4548  035002  000200                        RTS     R0                  ;RETURN
```

```
                                .SBTTL   *STORE RK611 UNIBUS REGISTERS

                                ;;*****************************************************************
                                ;*
                                ;*      THIS SUBROUTINE IS CALLED BY THE RK06 DRIVER TO STORE ALL
                                ;*      RK611 REGISTER IN TEMPORARY LOCATIONS.
                                ;*
                                ;*CALL  JSR     PC,I.STOR
                                ;*      RETURN
                                ;*
                                ;*      REGISTER        USE
                                ;*      --------        ---
                                ;*
                                ;*      R2              ADDRESS OF RK611 REGISTERS
                                ;*
                                ;;*****************************************************************

035004  016237  000000  002532 I.STOR: MOV     RKCS1(R2),T.CS1 ;STORE ALL CONTROLLER REGISTERS
035012  016237  000010  002534         MOV     RKCS2(R2),T.CS2 ;  EXCEPT DATA BUFFER
035020  016237  000002  002536         MOV     RKWC(R2),T.WCR
035026  016237  000004  002540         MOV     RKBA(R2),T.BA
035034  016237  000006  002542         MOV     RKDA(R2),T.DA
035042  016237  000012  002552         MOV     RKDS(R2),T.DS
035050  016237  000014  002550         MOV     RKER(R2),T.ER
035056  016237  000016  002546         MOV     RKASOF(R2),T.ASOF
035064  016237  000020  002544         MOV     RKDCYL(R2),T.DC
035072  016237  000022  002554         MOV     RKMR1(R2),T.MR1
035100  016237  000034  002556         MOV     RKMR2(R2),T.MR2
035106  016237  000036  002560         MOV     RKMR3(R2),T.MR3
035114  016237  000030  002562         MOV     RKECPS(R2),T.POS
035122  016237  000032  002564         MOV     RKECPT(R2),T.PAT
035130  000207                         RTS     PC              ;RETURN
```

```
                                .SBTTL   *STORE CONTROLLER STATUS

                                ;;*****************************************************************
                                ;*
                                ;*      THIS SUBROUTINE IS CALLED BY THE RK06 DRIVER AT PRIORITY 7.
                                ;*      THE FOLLOWING REGISTERS WILL BE STORED:
                                ;*
                                ;*              COMMAND AND STATUS REGISTER 2
                                ;*              WORD COUNT REGISTER
                                ;*              BUS ADDRESS REGISTER
                                ;*              DESIRED TRACK AND SECTOR
                                ;*              STATUS REGISTER
                                ;*              ERROR REGISTER
                                ;*              ATTENTION SUMMARY/OFFSET REGISTER
                                ;*              CYLINDER ADDRESS REGISTER
                                ;*
                                ;*CALL  JSR     PC,I.CSTS
                                ;*      RETURN
                                ;*
                                ;*      THIS ROUTINE ASSUMES THE FOLLOWING REGISTERS CONTAIN:
                                ;*
                                ;*
                                ;*              REGISTER                CONTENTS
                                ;*              --------                --------
                                ;*
                                ;*              R2                      RK06 BASE ADDRESS
                                ;*              R5                      ADDRESS OF PARAMETER BLOCK
                                ;*
                                ;;*****************************************************************

035132  042765  177741  000016  I.CSTS: BIC     #177741,P.CS1(R5)       ;CLEAR ALL BITS EXCEPT FUNCTION
                                                                        ;OF LAST COMMAND ISSUED
035140  042737  000036  002532          BIC     #36,T.CS1               ;CLEAR FUNCTION OF CS1 STATUS
035146  053765  002532  000016          BIS     T.CS1,P.CS1(R5)         ;GENERATE CS1 STATUS INFORMATION
035154  016265  000010  000020  I.CST1: MOV     RKCS2(R2),P.CS2(R5)     ;STORE COMMAND AND STATUS REGISTER 2
035162  016265  000002  000022          MOV     RKWC(R2),P.WCR(R5)      ;STORE WORD COUNT REGISTER
035170  016265  000004  000024          MOV     RKBA(R2),P.BAR(R5)      ;STORE BUS ADDRESS REGISTER
035176  016265  000006  000026          MOV     RKDA(R2),P.DTS(R5)      ;STORE DESIRED TRACK AND SECTOR
035204  016265  000012  000036          MOV     RKDS(R2),P.DS(R5)       ;STORE DRIVE STATUS REGISTER
035212  016265  000014  000034          MOV     RKER(R2),P.ER(R5)       ;STORE ERROR REGISTER
035220  016265  000016  000032          MOV     RKASOF(R2),P.ASOF(R5)   ;STORE ATTENTION SUMMARY AND
                                                                        ;  OFFSET
035226  016265  000020  000030          MOV     RKDCYL(R2),P.DCYL(R5)   ;STORE CYLINDER ADDRESS
035234  000207                  RTS     PC                              ;RETURN
```

```
                                .SBTTL   *GATHER DRIVE STATUS

                                ;;*****************************************************************
                                ;*
                                ;*      THIS SUBROUTINE WILL BE USED TO GATHER DRIVE STATUS
                                ;*      BYTE 01, 10, AND 11. IT IS ASSUMED THAT THE DRIVE
                                ;*      HAS PREVIOUSLY BEEN SEIZED.  IT RUNS AT PRIORITY 7.
                                ;*
                                ;*CALL  JSR     R0,I.STAT
                                ;*      <ADDRESS OF ERROR RETURN>
                                ;*      RETURN
                                ;*
                                ;*      THIS ROUTINE ASSUMES THE FOLLOWING REGISTERS CONTAIN:
                                ;*
                                ;*
                                ;*              REGISTER                CONTENTS
                                ;*              --------                --------
                                ;*
                                ;*              R2                      RK06 BASE ADDRESS
                                ;*              R5                      ADDRESS OF PARAMETER BLOCK
                                ;*
                                ;*      ROUTINES USED:
                                ;*              I.ISSU
                                ;*
                                ;;*****************************************************************

035236  012762  000001  000022  I.STAT: MOV     #1,RKMR1(R2)            ;LOAD MAINTENANCE REGISTER 1
                                                                        ;  FOR STATUS BYTE 01
035244  112737  000001  002532          MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
035252  004037  034562                  JSR     R0,I.ISSU               ;GET STATUS BYTES 01
035256  035446                          3$                              ;ERROR RETURN
035260  013765  002556  000044          MOV     T.MR2,P.A01(R5)         ;STORE STATUS BYTE 01 MESS A
035266  013765  002560  000046          MOV     T.MR3,P.B01(R5)         ;STORE STATUS BYTE 01 MESS B
035274  012762  000002  000022          MOV     #2,RKMR1(R2)            ;LOAD MAINTENANCE REGISTER 1
                                                                        ;  FOR STATUS BYTE 10
035302  112737  000001  002532          MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
035310  004037  034562                  JSR     R0,I.ISSU               ;GET STATUS BYTES 10
035314  035446                          3$                              ;ERROR RETURN
035316  013765  002556  000050          MOV     T.MR2,P.A10(R5)         ;STORE STATUS BYTE 10 MESS A
035324  013765  002560  000052          MOV     T.MR3,P.B10(R5)         ;STORE STATUS BYTE 10 MESS B
035332  012762  000003  000022          MOV     #3,RKMR1(R2)            ;LOAD MAINTENANCE REGISTER
                                                                        ;  FOR STATUS BYTE 11
035340  112737  000001  002532          MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
035346  004037  034562                  JSR     R0,I.ISSU               ;GET STATUS BYTES 11
035352  035446                          3$                              ;ERROR RETURN
035354  013765  002556  000054          MOV     T.MR2,P.A11(R5)         ;STORE STATUS BYTE 11 MESS A
035362  013765  002560  000056          MOV     T.MR3,P.B11(R5)         ;STORE STATUS BYTE 11 MESS B
035370  005062  000022                  CLR     RKMR1(R2)               ;LOAD MAINTENANCE REGISTER 1
                                                                        ;  FOR STATUS BYTE 00
035374  112737  000001  002532          MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
035402  004037  034562                  JSR     R0,I.ISSU               ;GET STATUS BYTES 00
035406  035446                          3$                              ;ERROR RETURN
035410  013765  002556  000040          MOV     T.MR2,P.A00(R5)         ;STORE STATUS BYTE 00 MESS A
035416  013765  002560  000042          MOV     T.MR3,P.B00(R5)         ;STORE STATUS BYTE 00 MESS B
035424  032737  001000  002560          BIT     #S.PAR,T.MR3            ;CHECK IF BAD PARITY DETECTED BY DRIVE
035432  001407                          BEQ     5$                      ;NO, RETURN NORMALLY
```

```
 4682  035434  052737  002000  002604          BIS     #E.DPAR,E.CONT          ;INDICATE BAD PARITY DETECTED BY DRIVE
 4683  035442  004737  035510                  JSR     PC,R.CONT               ;REPORT ERROR
 4684  035446  011000                  3$:     MOV     (R0),R0                 ;LOAD R0 FOR ERROR RETURN
 4685  035450  000200                          RTS     R0                      ;RETURN
 4686
 4687  035452  052765  001000  000014  5$:     BIS     #PBSVAL,P.PRST(R5)      ;SET PARAMETER BLOCK STATUS VALID
 4688  035460  005720                          TST     (R0)+                   ;ADJUST R0 FOR NORMAL RETURN
 4689  035462  000200                          RTS     R0                      ;RETURN
 4690
```

```
 4691                                   .SBTTL   *COMMON DRIVER RETURNS
 4692
 4693  035464  105037  002633  R.ABNL: CLRB     INTMSK          ;INHIBIT FUTURE DRIVE INTERRUPT REPORTING
 4694  035470  004777  145104          JSR      PC,@A.ABNL      ;INDICATE ABNORMAL RETURN
 4695  035474  000207                  RTS      PC              ;RETURN
 4696
 4697  035476  105037  002633  R.NORM: CLRB     INTMSK          ;INHIBIT FUTURE DRIVE INTERRUPT REPORTING
 4698  035502  004777  145070          JSR      PC,@A.NORM      ;INDICATE NORMAL RETURN
 4699  035506  000207                  RTS      PC              ;RETURN
 4700
 4701  035510  105037  002633  R.CONT: CLRB     INTMSK          ;INHIBIT FUTURE DRIVE INTERRUPT REPORTING
 4702  035514  105037  002632          CLRB     W.TIME          ;RESET WATCH DOG TIMING ON THIS DRIVE
 4703  035520  005037  002646          CLR      W.DRV           ;CLEAR TIMING COUNT FOR THIS DRIVE
 4704  035524  004777  145052          JSR      PC,@A.CONT      ;INDICATE CONTROLLER ERROR RETURN
 4705  035530  000207                  RTS      PC              ;RETURN
```

```
                        .SBTTL   *COMMAND INITATOR

                        ;;*****************************************************************
                        ;*
                        ;*      THIS SUBROUTINE WILL INITIATE ALL COMMANDS AS SPECIFIED
                        ;*      BY THE COMMAND FIELD OF THE PARAMETER BLOCK. THE FOLLOWING
                        ;*      SPECIAL COMMAND ARE ALSO EXECUTED:
                        ;*
                        ;*              RELEASE
                        ;*              CONROLLER CLEAR
                        ;*              SUBSYSTEM CLEAR
                        ;*              READ ALL DRIVE STATUS
                        ;*              READ SPECIFIED HEADER
                        ;*
                        ;*      THE ABOVE COMMANDS ARE TRANSLATED INTO A SEQUENCE OF COMMANDS
                        ;*
                        ;*CALL  JSR     PC,C.INIT
                        ;*      (ADDRESS OF PARAMETER BLOCK)
                        ;*      RETURN
                        ;*
                        ;*      FOR THE SEQUENTIAL OPERATIONS, THE DRIVER WILL LOAD THE
                        ;*      LOCATIONS, PBLKT AND INTMSK.
                        ;*
                        ;*      ROUTINES USED:
                        ;*              W.WTCH
                        ;*              I.CSTS
                        ;*              I.STAT
                        ;*              I.CCLR
                        ;*
                        ;;*****************************************************************

004737  035532  010546                C.INIT: MOV     R5,-(SP)        ;STORE R5 ON STACK
004740  035534  010446                        MOV     R4,-(SP)        ;STORE R4 ON STACK
004741  035536  010346                        MOV     R3,-(SP)        ;STORE R3 ON STACK
004742  035540  010246                        MOV     R2,-(SP)        ;STORE R2 ON STACK
004743  035542  010146                        MOV     R1,-(SP)        ;STORE R1 ON STACK
004744  035544  010046                        MOV     R0,-(SP)        ;STORE R0 ON STACK
004745  035546  013746  177776               MOV     PS,-(SP)        ;STORE PSW ON STACK
004746  035552  013737  002574  177776       MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
004747  035560  017605  000016               MOV     @16(SP),R5      ;STORE PARAMETER BLOCK ADDRESS
004750  035564  062766  000002  000016       ADD     #2,16(SP)       ;ADJUST RETURN
004751  035572  016504  000000               MOV     P.DRVN(R5),R4   ;STORE DRIVE NUMBER
004752  035576  042704  177770               BIC     #^C<DRVMSK>,R4  ;MASK OUT JUNK
004753  035602  010537  002644               MOV     R5,PBLKT        ;LOAD PARAMETER BLOCK TABLE
004754  035606  116437  002634  002633       MOVB    I.DRV(R4),INTMSK ;LOAD INTERRUPT MASK
004755  035614  116437  002634  002632       MOVB    I.DRV(R4),W.TIME ;SET WATCH-DOG TIMER FLAG
004756  035622  013737  002614  002646       MOV     W.SEC,W.DRV     ;LOAD WATCH-DOG TIME
004757
004760  035630  013702  002570               MOV     RKBAS,R2        ;LOAD R2 WITH RK06 ADDRESS BASE
004761
                        ;       RESET ALL BITS IN PROGRAM DEVICE STATUS REGISTER EXCEPT
                        ;               DRIVE IN USE
                        ;               WRITE FOR WRITE CHECK
                        ;               NO CHECK
                        ;               DROP DRIVE FROM TEST SEQUENCE
                        ;               INHIBIT BUS ADDRESS INCREMENT
```

```
 4762  035634  042765  075176  000014          BIC     #^C<DRVUSE!W.WCK!NOCHK!DRPDRV!DTBAII>,P.PRST(R5)
 4763
 4764  035642  010500                          MOV     R5,R0           ;STORE PARAMETER BLOCK ADDRESS
 4765  035644  062700  000016                  ADD     #P.CS1,R0       ;CALCULATE FIRST LOCATION TO BE CLEARED
 4766  035650  010501                          MOV     R5,R1           ;STORE PARAMETER BLOCK ADDRESS
 4767  035652  062701  000062                  ADD     #P.EPAT,R1      ;CALCULATE LAST LOCATION TO BE CLEARED
 4768
 4769  035656  005020                  1$:     CLR     (R0)+           ;CLEAR RETURN PARAMETER
 4770  035660  020001                          CMP     R0,R1           ;CHECK IF FINISHED
 4771  035662  101775                          BLOS    1$              ;NO, CLEAR NEXT RETURN PARAMETER
 4772  035664  105037  002626                  CLRB    I.ISRL          ;CLEAR RELEASE OR INTERRUPT ISSUED
 4773  035670  010465  000020                  MOV     R4,P.CS2(R5)    ;STORE DRIVE NUMBER
 4774  035674  005062  000022                  CLR     RKMR1(R2)       ;CLEAR RK06 MAINTENANCE REGISTER 1
 4775  035700  132765  000040  000001          BITB    #BIT5,P.CMND(R5) ;CHECK IF SPECIAL COMMAND
 4776  035706  001402                          BEQ     3$              ;NO  PROCESS
 4777  035710  000137  036424                  JMP     C.SPEC          ;JUMP TO SPECIAL COMMAND PROCESSOR
 4778
 4779  035714  122765  000107  000001  3$:     CMPB    #UNLOAD,P.CMND(R5) ;CHECK IF POSITIONING COMMAND
 4780                                                                  ;  START SPINDLE
 4781                                                                  ;  RECALIBRATE
 4782                                                                  ;  OFFSET
 4783                                                                  ;  SEEK
 4784                                                                  ;  UNLOAD
 4785
 4786  035722  101174                          BHI     25$             ;NO, DRIVE COMMAND
 4787                                                                  ;  SELECT DRIVE
 4788                                                                  ;  PACK ACKNOWLEDGE
 4789                                                                  ;  CLEAR
 4790
 4791  035724  122765  000117  000001          CMPB    #SEEK,P.CMND(R5) ;CHECK IF DATA TRANSFER
 4792  035732  103540                          BLO     20$             ;YES, DATA TRANSFER COMMAND
 4793                                                                  ;  READ DATA
 4794                                                                  ;  WRITE DATA
 4795                                                                  ;  READ HEADER
 4796                                                                  ;  WRITE HEADER
 4797                                                                  ;  WRITE CHECK
 4798  035734  016562  000020  000010          MOV     P.CS2(R5),RKCS2(R2) ;LOAD DRIVE NUMBER
 4799  035742  052765  000002  000014          BIS     #DR.POS,P.PRST(R5) ;SET DRIVE POSITIONING
 4800  035750  005037  002606                  CLR     @.WAIT          ;CLEAR WAIT FOR COMMAND
 4801  035754  122765  000117  000001          CMPB    #SEEK,P.CMND(R5) ;CHECK IF SEEK
 4802  035762  001007                          BNE     5$              ;NO, CHECK FOR OFFSET
 4803  035764  016562  000002  000020          MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS
 4804  035772  016562  000004  000006          MOV     P.SECT(R5),RKDA(R2) ;LOAD SECTOR AND TRACK
 4805  036000  000431                          BR      8$              ;GO ISSUE COMMAND
 4806
 4807  036002  122765  000115  000001  5$:     CMPB    #OFFSET,P.CMND(R5) ;CHECK IF OFFSET
 4808  036010  001007                          BNE     6$              ;NO, CHECK FOR UNLOAD
 4809  036012  116565  000006  000032          MOVB    P.OFST(R5),P.ASOF(R5) ;STORE OFFSET
 4810  036020  016562  000032  000016          MOV     P.ASOF(R5),RKASOF(R2) ;LOAD OFFSET REGISTER
 4811  036026  000416                          BR      8$              ;GO ISSUE COMMAND
 4812
 4813  036030  122765  000111  000001  6$:     CMPB    #SRTSPL,P.CMND(R5) ;CHECK IF START SPINDLE
 4814  036036  001003                          BNE     7$              ;NO, CHECK IF RECAL
 4815  036040  013737  002620  002646          MOV     @.MIN,@.DRV     ;LOAD WATCH DOG TIME FOR 1 MINUTE
 4816  036046  122765  000113  000001  7$:     CMPB    #RECAL,P.CMND(R5) ;CHECK IF RECAL
 4817  036054  001003                          BNE     8$              ;NO, CONTINUE
```

```
 4818  036056  013737  002616  002646          MOV     W.BSEC,W.DRV     ;LOAD RECAL TIME FOR 8 SECONDS
 4819  036064  116565  000007  000017  8$:     MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 4820  036072  042765  165777  000016          BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT FORMAT
 4821                                                                  ;  AND DRIVE TYPE
 4822  036100  116565  000001  000016          MOVB    P.CMND(R5),P.CS1(R5) ;MOVE COMMAND INTO CS1
 4823  036106  042765  000200  000014          BIC     #W.WCK,P.PRST(R5) ;RESET WRITE FOR WRITE CHECK
 4824  036114  032765  000400  000014          BIT     #NOCHK,P.PRST(R5) ;CHECK IN NO CHECK MODE
 4825  036122  001533                          BEQ     30$              ;NO, SKIP CLEAR OF INTERRUPT ENABLE
 4826  036124  042765  000100  000016          BIC     #IE,P.CS1(R5)    ;CLEAR INTERRUPT ENABLE
 4827  036132  016562  000016  000000          MOV     P.CS1(R5),RKCS1(R2) ;ISSUE COMMAND
 4828  036140  004737  032102          10$:    JSR     PC,W.WTCH        ;CALL WATCH DOG TIMER
 4829  036144  016237  000000  002532          MOV     RKCS1(R2),T.CS1  ;STORE COMMAND AND STATUS REGISTER 1
 4830  036152  032737  000200  002532          BIT     #RDY,T.CS1       ;WAIT FOR READY
 4831  036160  001767                          BEQ     10$
 4832  036162  032737  100000  002532          BIT     #CERR,T.CS1      ;CHECK FOR ERROR
 4833  036170  001011                          BNE     15$              ;YES, GIVE NORMAL RETURN
 4834  036172  004737  032102          11$:    JSR     PC,W.WTCH        ;CALL WATCH DOG TIMER
 4835  036176  016237  000016  002546          MOV     RKASOF(R2),T.ASOF ;STORE ATTENTION SUMMARY
 4836  036204  133737  002633  002547          BITB    INTMSK,T.ASOF+1  ;CHECK IF INTERRUPT HAS OCCURRED
 4837  036212  001767                          BEQ     11$              ;WAIT FOR DRIVE INTERRUPT
 4838  036214  105037  002632          15$:    CLRB    W.TIME           ;RESET TIMING ON THIS DRIVE
 4839  036220  005037  002646                  CLR     W.DRV            ;CLEAR DRIVE TIMING COUNT
 4840  036224  004737  035476                  JSR     PC,R.NORM        ;INDICATE COMMAND IS FINISHED
 4841  036230  000137  037404                  JMP     C.RTRN           ;RESTORE REGISTERS
 4842
 4843  036234  016562  000010  000004  20$:    MOV     P.BALO(R5),RKBA(R2) ;LOAD BUS ADDRESS REGISTER
 4844  036242  016562  000012  000002          MOV     P.WC(R5),RKWC(R2) ;LOAD WORD COUNT REGISTER
 4845  036250  016562  000002  000020          MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS REGISTER
 4846  036256  016562  000004  000006          MOV     P.SECT(R5),RKDA(R2) ;LOAD SECTOR AND TRACK NUMBER
 4847  036264  122765  000131  000001          CMPB    #WRTCHK,P.CMND(R5) ;CHECK IF WRITE CHECK COMMAND
 4848  036272  001010                          BNE     25$              ;NO, GO ISSUE THE COMMAND
 4849  036274  032765  000200  000014          BIT     #W.WCK,P.PRST(R5) ;CHECK IF WRITE COMMAND SHOULD BE ISSUED
 4850  036302  001404                          BEQ     25$              ;NO, GO ISSUE THE COMMAND
 4851  036304  012765  000123  000016          MOV     #WRDATA,P.CS1(R5) ;ISSUE WRITE COMMAND
 4852  036312  000406                          BR      26$              ;GO ISSUE COMMAND
 4853
 4854  036314  116565  000001  000016  25$:    MOVB    P.CMND(R5),P.CS1(R5) ;MOVE COMMAND INTO CS1
 4855  036322  042765  000200  000014          BIC     #W.WCK,P.PRST(R5) ;RESET WRITE FOR WRITE CHECK
 4856  036330  116565  000007  000017  26$:    MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 4857  036336  142765  177750  000017          BICB    #^C<B.CFMT!B.CDT!B.BA16!B.BA17>,P.CS1+1(R5) ;CLEAR ALL BITS EXCEPT
 4858                                                                  ;  FORMAT, DRIVE TYPE, AND BUS ADDRESS
 4859                                                                  ;  BITS 16-17
 4860  036344  010537  002606                  MOV     R5,0.WAIT        ;LOAD WAITING FOR COMMAND
 4861  036350  032765  100000  000014          BIT     #DTBAII,P.PRST(R5) ;CHECK IF INHIBIT BUS ADDRESS INCREMENT
 4862  036356  001403                          BEQ     27$              ;NO, LOAD CS2
 4863  036360  052765  000020  000020          BIS     #BAI,P.CS2(R5)   ;SET INHIBIT BUS ADDRESS INCREMENT
 4864  036366  016562  000020  000010  27$:    MOV     P.CS2(R5),RKCS2(R2) ;LOAD CS2
 4865  036374  032765  000400  000014          BIT     #NOCHK,P.PRST(R5) ;CHECK IN NO CHECK MODE
 4866  036402  001403                          BEQ     30$              ;NO, SKIP CLEAR OF INTERRUPT ENABLE
 4867  036404  042765  000100  000016          BIC     #IE,P.CS1(R5)    ;CLEAR INTERRUPT ENABLE
 4868  036412  016562  000016  000000  30$:    MOV     P.CS1(R5),RKCS1(R2) ;ISSUE COMMAND
 4869  036420  000137  037404                  JMP     C.RTRN           ;RESTORE REGISTERS
 4870
 4871                                  .SBTTL   *SPECIAL COMMAND PROCESSING
 4872
 4873  036424  122765  000141  000001  C.SPEC: CMPB    #RDSTAT,P.CMND(R5) ;CHECK IF READ DRIVE STATUS
```

```
 4874  036432  001132                        BNE     10$                 ;NO, PROCESS OTHER COMMANDS
 4875  036434  016562  000020  000010        MOV     P.CS2(R5),RKCS2(R2) ;LOAD CS2 FOR COMMAND
 4876  036442  116565  000007  000017        MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 4877  036450  042765  165777  000016        BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT FORMAT
 4878                                                                ;  AND DRIVE TYPE
 4879  036456  112765  000001  000016        MOVB    #DR.SEL,P.CS1(R5) ;STORE COMMAND
 4880  036464  016562  000016  000000        MOV     P.CS1(R5),RKCS1(R2) ;ISSUE COMMAND
 4881  036472  004737  032102          2$:   JSR     PC,W.WTCH           ;CALL WATCH-DOG TIMER
 4882  036476  016265  000000  000016        MOV     RKCS1(R2),P.CS1(R5) ;STORE COMMAND AND STATUS REG. 1
 4883  036504  032765  000200  000016        BIT     #RDY,P.CS1(R5)  ;WAIT FOR READY
 4884  036512  001767                        BEQ     2$
 4885  036514  004737  035154                JSR     PC,I.CST1           ;STORE CONTROLLER REGISTERS
 4886  036520  016265  000034  000040        MOV     RKMR2(R2),P.AOO(R5) ;STORE STATUS BYTE OO MESSAGE A
 4887  036526  016265  000036  000042        MOV     RKMR3(R2),P.BOO(R5) ;STORE STATUS BYTE OO MESSAGE B
 4888  036534  032765  100000  000016        BIT     #CERR,P.CS1(R5) ;CHECK IF CONTROLLER ERROR
 4889  036542  001436                        BEQ     6$                  ;NO, GATHER DRIVE STATUS
 4890  036544  105037  002632                CLRB    W.TIME              ;RESET WATCH DOG TIMING ON THIS DRIVE
 4891  036550  005037  002646                CLR     W.DRV               ;CLEAR WATCH DOG COUNT
 4892  036554  032765  000400  000014        BIT     #NOCHK,P.PRST(R5) ;CHECK IF NO CHECK MODE
 4893  036562  001043                        BNE     8$                  ;YES, INDICATE NORMAL RETURN
 4894  036564  032765  001000  000020        BIT     #MDS,P.CS2(R5)  ;CHECK IF MULTIPLE DRIVE SELECT
 4895  036572  001043                        BNE     9$                  ;YES, INDICATE CONTROLLER ERROR
 4896  036574  004037  034500                JSR     R0,I.CCLR           ;CLEAR ERROR
 4897  036600  037404                        C.RTRN                      ;ERROR RETURN
 4898  036602  032765  010400  000020        BIT     #NED!UFE,P.CS2(R5) ;CHECK IF NON-EXISTENT DRIVE OR UNIT FIELD ERROR
 4899  036610  001007                        BNE     5$                  ;REPORT ERROR
 4900  036612  032765  000001  000036        BIT     #DRA,P.DS(R5)   ;CHECK IF DRIVE AVAILIABLE
 4901  036620  001003                        BNE     5$                  ;YES, REPORT ERROR
 4902  036622  052765  010000  000014        BIS     #DRVSZD,P.PRST(R5) ;INDICATE DRIVE IS SEIZED BY OTHER PORT
 4903  036630  004737  035464          5$:   JSR     PC,R.ABNL           ;INDICATE ABNORMAL RETURN
 4904  036634  000137  037404                JMP     C.RTRN              ;RESTORE REGISTERS
 4905
 4906  036640  004037  035236          6$:   JSR     R0,I.STAT           ;GATHER DRIVE STATUS
 4907  036644  037404                        C.RTRN                      ;ERROR RETURN
 4908  036646  105037  002632                CLRB    W.TIME              ;STOP WATCH-DOG TIMING ON DRIVE
 4909  036652  005037  002646                CLR     W.DRV               ;RESET WATCH-DOG TIME
 4910  036656  032765  000400  000014        BIT     #NOCHK,P.PRST(R5) ;CHECK IF NO CHECK MODE
 4911  036664  001402                        BEQ     8$                  ;NO, REPORT ERROR
 4912  036666  005062  000000                CLR     RKCS1(R2)           ;CLEAR INTERRUPT ENABLE
 4913  036672  004737  035476          8$:   JSR     PC,R.NORM           ;REPORT COMMAND COMPLETE
 4914  036676  000137  037404                JMP     C.RTRN              ;RESTORE REGISTERS
 4915
 4916  036702  052737  100000  002604  9$:   BIS     #E.MDS,E.CONT   ;SET MULTIPLE DRIVE SELECT
 4917  036710  004737  035510                JSR     PC,R.CONT           ;INDICATE CONTROLLER ERROR
 4918  036714  000137  037404                JMP     C.RTRN
 4919
 4920  036720  122765  000140  000001  10$:   CMPB    #RELEAS,P.CMND(R5) ;CHECK IF RELEASE COMMAND
 4921  036726  001040                        BNE     13$                 ;NO, CHECK IF READ ALL HEADERS
 4922  036730  010537  002606                MOV     R5,O.WAIT           ;STORE PARAMETER BLOCK ADDRESS IN
 4923                                                                ;  WAIT FOR COMMAND
 4924  036734  052765  000010  000020        BIS     #RLS,P.CS2(R5)  ;SET RELEASE BIT
 4925  036742  016562  000020  000010        MOV     P.CS2(R5),RKCS2(R2) ;LOAD CS2 FOR DESELECT
 4926  036750  112737  000001  002626        MOVB    #1,I.ISRL           ;SET FLAG FOR RELEASE COMMAND
 4927  036756  116565  000007  000017        MOVB    P.CS1H(R5),P.CS1+1(R5);STORE BITS 8-15 OF CS1
 4928  036764  042765  165777  000016        BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT FORMAT
 4929                                                                ;  AND DRIVE TYPE
```

```
4930  036772  112765  000101  000016          MOVB    #SELDRV,P.CS1(R5) ;STORE COMMAND
4931  037000  032765  000400  000014          BIT     #NOCHK,P.PRST(R5) ;CHECK IF NO CHECK MODE
4932  037006  001403                          BEQ     11$               ;NO, DO NOT RESET INTERRUPT ENABLE
4933  037010  042765  000100  000016          BIC     #IE,P.CS1(R5)     ;RESET INTERRUPT ENABLE
4934  037016  016562  000016  000000  11$:    MOV     P.CS1(R5),RKCS1(R2) ;ISSUE COMMAND
4935  037024  000137  037404                  JMP     C.RTRN            ;RESTORE REGISTERS
4936
4937  037030  122765  000164  000001  13$:    CMPB    #RDALHD,P.CMND(R5) ;CHECK IF READ ALL HEADERS
4938  037036  001053                          BNE     30$               ;NO, CHECK IF CONTROLLER CLEAR
4939  037040  010537  002606                  MOV     R5,O.WAIT         ;SET WAITING FOR COMMAND COMMPLETION
4940  037044  016537  000010  002622          MOV     P.BALO(R5),HDR.AD ;LOAD HEADER ADDRESS
4941  037052  132765  000020  000007          BITB    #B.CFMT,P.CS1H(R5) ;CHECK IF 22 SECTOR FORMANT
4942  037060  001404                          BEQ     14$               ;YES, LOAD 22 IN HEADER COUNT
4943  037062  012737  000024  002624          MOV     #20.,HDR.CT       ;LOAD 20 IN SECTOR COUNT
4944  037070  000403                          BR      22$               ;GO ISSUE READ HEADER COMMAND
4945
4946  037072  012737  000026  002624  14$:    MOV     #22.,HDR.CT       ;LOAD 22 IN SECTOR COUNT
4947  037100  016562  000002  000020  22$:    MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS
4948  037106  016562  000004  000006          MOV     P.SECT(R5),RKDA(R2) ;LOAD TRACK NUMBER
4949  037114  016562  000020  000010          MOV     P.CS2(R5),RKCS2(R2) ;LOAD DRIVE NUMBER
4950  037122  116525  000007  000017          MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
4951  037130  042765  165777  000016          BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT DRIVE TYPE
4952                                                                  ;  AND FORMAT
4953  037136  112765  000125  000016          MOVB    #RDHEAD,P.CS1(R5) ;STORE READ HEADER COMMAND
4954  037144  032765  000400  000014          BIT     #NOCHK,P.PRST(R5) ;CHECK IF NO CHECK MODE
4955  037152  001027                          BNE     34$               ;YES, INDICATE ILLEGAL DRIVER COMMAND
4956  037154  016562  000016  000000          MOV     P.CS1(R5),RKCS1(R2) ;ISSUE READ HEADER
4957  037162  000137  037404                  JMP     C.RTRN            ;RESTORE REGISTERS
4958
4959  037166  122765  000176  000001  30$:    CMPB    #CONCLR,P.CMND(R5) ;CHECK IF CONTROLLER CLEAR
4960  037174  001012                          BNE     32$               ;NO, CHECK IF SUBSYSTEM CLEAR
4961  037176  004037  034500                  JSR     R0,I.CCLR         ;CLEAR CONTROLLER
4962  037202  037404                          C.RTRN                    ;ERROR RETURN
4963  037204  032765  000400  000014          BIT     #NOCHK,P.PRST(R5) ;CHECK IF NO CHECK MODE
4964  037212  001472                          BEQ     40$               ;NO, INDICATE NORMAL RETURN
4965  037214  005062  000000                  CLR     RKCS1(R2)         ;RESET INTERRUPT ENABLE
4966  037220  000467                          BR      40$               ;INDICATE NORMAL RETURN
4967
4968  037222  122765  000177  000001  32$:    CMPB    #SUBCLR,P.CMND(R5) ;CHECK IF SUBSYSTEM CLEAR
4969  037230  001406                          BEQ     36$               ;YES, CLEAR SUBSYSTEM
4970  037232  052737  000100  002604  34$:    BIS     #E.ILLD,E.CONT    ;SET ILLEGAL DRIVER COMMAND
4971  037240  004737  035510                  JSR     PC,R.CONT         ;REPORT ERROR
4972  037244  000457                          BR      C.RTRN            ;RESTORE REGISTERS
4973
4974  037246  012762  000040  000010  36$:    MOV     #SCLR,RKCS2(R2)   ;ISSUE SUBSYSTEM CLEAR
4975  037254  016265  000000  000016          MOV     RKCS1(R2),P.CS1(R5) ;STORE COMMAND AND STATUS REGISTER 1
4976  037262  032765  100000  000016          BIT     #CERR,P.CS1(R5)   ;CLEAR IF CONTROLLER ERROR RESET
4977  037270  001406                          BEQ     37$               ;NO, FINISH COMMAND
4978  037272  052737  000001  002604          BIS     #BIT0,E.CONT      ;SET CLEAR SUBSYSTEM DID NOT CLEAR
4979                                                                  ;  CONTROLLER ERROR
4980  037300  004737  035510                  JSR     PC,R.CONT         ;REPORT ERROR
4981  037304  000437                          BR      C.RTRN            ;RESTORE REGISTERS
4982
4983  037306  013746  002612          37$:    MOV     W.MILI,-(SP)      ;LOAD 16 MILI-SECOND COUNT FOR ATTENTION
4984                                                                  ;  TO DISAPPEAR
4985  037312  016265  000000  000016  38$:    MOV     RKCS1(R2),P.CS1(R5) ;STORE CS1
```

```
 4986  037320  032765  040000  000016          BIT     #DI.P.CS1(R5)   ;CHECK IF ATTENTIONS CLEARED
 4987  037326  001411                          BEQ     39$             ;YES, FINISH COMMAND
 4988  037330  005316                          DEC     (SP)            ;DECREMENT 16 MILISECOND COUNT
 4989  037332  001367                          BNE     38$             ;CHECK DRIVE INTERRUPT AGAIN
 4990  037334  005726                          TST     (SP)+           ;ADJUST STACK
 4991  037336  052737  000040  002604          BIS     #E.SCLR,E.CONT  ;SET SUBSYSTEM CLEAR DID NOT CLEAR
 4992                                                                  ;  DRIVE ATTENTIONS
 4993  037344  004737  035510                  JSR     PC,R.CONT       ;REPORT ERROR
 4994  037350  000415                          BR      C.RTRN          ;RESTORE REGISTER
 4995
 4996  037352  005726                  39$:    TST     (SP)+           ;ADJUST STACK
 4997  037354  032765  000400  000014          BIT     #NOCHK,P.PRST(R5) ;CHECK IF NO CHECK MODE
 4998  037362  001010                          BNE     C.RTRN          ;YES, RESTORE REGISTERS
 4999  037364  112737  177777  002626          MOVB    #-1,I.ISRL      ;SET INTERRUPT ENABLE SET
 5000  037372  012762  000100  000000          MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
 5001  037400  004737  035476          40$:    JSR     PC,R.NORM       ;INDICATE NORMAL TERMINATION
 5002
 5003  037404  012637  177776          C.RTRN: MOV     (SP)+,PS        ;RESTORE PSW
 5004  037410  012600                          MOV     (SP)+,R0        ;RESTORE R0
 5005  037412  012601                          MOV     (SP)+,R1        ;RESTORE R1
 5006  037414  012602                          MOV     (SP)+,R2        ;RESTORE R2
 5007  037416  012603                          MOV     (SP)+,R3        ;RESTORE R3
 5008  037420  012604                          MOV     (SP)+,R4        ;RESTORE R4
 5009  037422  012605                          MOV     (SP)+,R5        ;RESTORE R5
 5010  037424  000207                          RTS     PC              ;RETURN
 5011                                  .SBTTL  OCTAL TO BINARY CONVERSION ROUTINE
 5012
 5013                                  ;;**************************************************************
 5014                                  ;*
 5015                                  ;*      THIS ROUTINE WILL CHECK A STRING OF ASCII CHARACTERS TERMINATED
 5016                                  ;*      WITH A NULL <000> OR COMMA.  IF THE CHARACTERS ARE LEGAL
 5017                                  ;*      IT  WILL GENERATE TWO BINARY WORDS PLACING THE LOW 16 BITS
 5018                                  ;*      ON THE STACK AND THE HIGH 16 BITS IN LOCATION $HIOCT.
 5019                                  ;*
 5020                                  ;*CALL
 5021                                  ;*      MOV     <ADDRESS OF ASCII STRING>,-(SP)
 5022                                  ;*      JSR     PC,OCTBIN
 5023                                  ;*      <ADDRESS OF ERROR RETURN>
 5024                                  ;*      RETURN
 5025                                  ;*
 5026                                  ;;**************************************************************
 5027
 5028  037426  010046                  OCTBIN: MOV     R0,-(SP)        ;SAVE R0
 5029  037430  010146                          MOV     R1,-(SP)        ;SAVE R1
 5030  037432  010246                          MOV     R2,-(SP)        ;SAVE R2
 5031  037434  016600  000010                  MOV     10(SP),R0       ;GET ADDRESS OF ASCII STRING
 5032  037440  005001                          CLR     R1              ;CLEAR DATA WORDS
 5033  037442  005002                          CLR     R2
 5034  037444  112046                  2$:     MOVB    (R0)+,-(SP)     ;PICK THIS CHARACTER
 5035  037446  001423                          BEQ     3$              ;IF ZERO GET OUT
 5036  037450  121627  000054                  CMPB    (SP),#',        ;CHECK IF COMMA
 5037  037454  001420                          BEQ     3$              ;IF COMMA GET OUT
 5038  037456  122716  000060                  CMPB    #'0,(SP)        ;MAKE SURE THIS CHARACTER IS
 5039  037462  003030                          BGT     4$              ;  AN OCTAL DIGIT
 5040  037464  122716  000067                  CMPB    #'7,(SP)
 5041  037470  002425                          BLT     4$
```

```
5042  037472  006301                          ASL     R1              ; *2
5043  037474  006102                          ROL     R2
5044  037476  006301                          ASL     R1              ; *4
5045  037500  006102                          ROL     R2
5046  037502  006301                          ASL     R1              ; *8
5047  037504  006102                          ROL     R2
5048  037506  042716  177770                  BIC     #^C7,(SP)       ;STRIP THE ASCII JUNK
5049  037512  062601                          ADD     (SP)+,R1        ;ADD THIS DIGIT
5050  037514  000753                          BR      2$              ;LOOP
5051  037516  005726                  3$:     TST     (SP)+           ;CLEAN PARTIAL FROM STACK
5052  037520  010166  000010                  MOV     R1,10(SP)       ;SAVE RESULT
5053  037524  010237  037560                  MOV     R2,$HIOCT
5054  037530  012602                          MOV     (SP)+,R2        ;RESTORE R2
5055  037532  012601                          MOV     (SP)+,R1        ;RESTORE R1
5056  037534  012600                          MOV     (SP)+,R0        ;RESTORE R0
5057  037536  062716  000002                  ADD     #2,(SP)         ;ADJUST RETURN
5058  037542  000207                          RTS     PC              ;RETURN
5059
5060  037544  005726                  4$:     TST     (SP)+           ;CLEAN UP PARTIAL FROM STACK
5061  037546  012602                          MOV     (SP)+,R2        ;RESTORE R2
5062  037550  012601                          MOV     (SP)+,R1        ;RESTORE R1
5063  037552  012600                          MOV     (SP)+,R0        ;RESTORE R0
5064  037554  013616                          MOV     @(SP)+,(SP)     ;PUT ADDRESS OF ERROR ROUTINE ON STACK
5065  037556  000207                          RTS     PC              ;GO PROCESS ERROR
5066  037560  000000          $HIOCT: .WORD   0               ;HIGH ORDER BITS GO HERE
5067                          .SBTTL  TYPE ROUTINE
5068
5069                          ;;*****************************************************************
5070                          ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
5071                          ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
5072                          ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
5073                          ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
5074                          ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
5075                          ;*
5076                          ;*CALL:
5077                          ;*1) USING A TRAP INSTRUCTION
5078                          ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
5079                          ;*OR
5080                          ;*      TYPE
5081                          ;*      MESADR
5082                          ;*
5083
5084  037562  105737  001157  $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
5085  037566  100002                  BPL     1$              ;;BR IF YES
5086  037570  000000                  HALT                    ;;HALT HERE IF NO TERMINAL
5087  037572  000407                  BR      3$              ;;LEAVE
5088  037574  010046          1$:     MOV     R0,-(SP)        ;;SAVE R0
5089  037576  017600  000002          MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
5090  037602  112046          2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
5091  037604  001005                  BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
5092  037606  005726                  TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
5093  037610  012600          60$:    MOV     (SP)+,R0        ;;RESTORE R0
5094  037612  062716  000002  3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
5095  037616  000002                  RTI                     ;;RETURN
5096  037620  122716  000011  4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
5097  037624  001430                  BEQ     8$
```

```
5098  037626  122716  000200          CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
5099  037632  001006                  BNE     5$
5100  037634  005726                  TST     (SP)+           ;;POP  <CR><LF> EQUIV
5101  037636  104401                  TYPE                    ;;TYPE A CR AND LF
5102  037640  001267                  $CRLF
5103  037642  105037  037776          CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
5104  037646  000755                  BR      2$              ;;GET NEXT CHARACTER
5105  037650  004737  037732  5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
5106  037654  123726  001156  6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
5107  037660  001350                  BNE     2$              ;;IF NO GO GET NEXT CHAR.
5108  037662  013746  001154          MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
5109                                                          ;;AND THE NULL CHAR.
5110  037666  105366  000001  7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
5111  037672  002770                  BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
5112  037674  004737  037732          JSR     PC,$TYPEC       ;;GO TYPE A NULL
5113  037700  105337  037776          DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
5114  037704  000770                  BR      7$              ;;LOOP
5115
5116                                  ;HORIZONTAL TAB PROCESSOR
5117
5118  037706  112716  000040  8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
5119  037712  004737  037732  9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
5120  037716  132737  000007  037776  BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
5121  037724  001372                  BNE     9$              ;;TAB STOP
5122  037726  005726                  TST     (SP)+           ;;POP SPACE OFF STACK
5123  037730  000724                  BR      2$              ;;GET NEXT CHARACTER
5124  037732  105777  141212  $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
5125  037736  100375                  BPL     $TYPEC
5126  037740  116677  000002  141204  MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
5127  037746  122766  000015  000002  CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
5128  037754  001003                  BNE     1$              ;;BRANCH IF NO
5129  037756  105037  037776          CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
5130  037762  000406                  BR      $TYPEX          ;;EXIT
5131  037764  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
5132  037772  001402                  BEQ     $TYPEX          ;;BRANCH IF YES
5133  037774  105227                  INCB    (PC)+           ;;COUNT THE CHARACTER
5134  037776  000000  $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
5135  040000  000207  $TYPEX: RTS     PC
5136
5137                                  .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
5138
5139                                  ;;******************************************************************
5140                                  ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
5141                                  ;*UNSIGNED OCTAL ASCIZ NUMBER.
5142                                  ;*CALL
5143                                  ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
5144                                  ;*      JSR     PC,@#$DB20      ;;CALL THE ROUTINE
5145                                  ;*      RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK
5146
5147
5148  040002  104411  $DB20:  SAVREG                  ;;SAVE ALL REGISTERS
5149  040004  016601  000002          MOV     2(SP),R1        ;;PICKUP THE POINTER TO LOW WORD
5150  040010  012705  040121          MOV     #$OCTVL+13.,R5  ;;POINTER TO DATA TABLE
5151  040014  012704  000014          MOV     #12.,R4         ;;DO ELEVEN CHARACTERS
5152  040020  012703  177770          MOV     #^C7,R3         ;;MASK
5153  040024  012100                  MOV     (R1)+,R0        ;;LOWER WORD
```

```
                        DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE

 5154  040026  012101                        MOV     (R1)+,R1        ;;HIGH WORD
 5155  040030  005002                        CLR     R2              ;;TERMINATOR
 5156  040032  110245                1$:     MOVB    R2,-(R5)        ;;PUT CHARACTER IN DATA TABLE
 5157  040034  010002                        MOV     R0,R2           ;;GET THIS DIGIT
 5158  040036  005304                        DEC     R4              ;;COUNT THIS CHARACTER
 5159  040040  003007                        BGT     3$              ;;BR IF NOT THE LAST DIGIT
 5160  040042  001405                        BEQ     2$              ;;BR IF IT IS THE LAST DIGIT
 5161  040044  005205                        INC     R5              ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
 5162  040046  010566  000002                MOV     R5,2(SP)        ;;ASCIZ CHAR. & PUT IT ON THE STACK
 5163  040052  104412                        RESREG                  ;;RESTORE ALL REGISTERS
 5164  040054  000207                        RTS     PC              ;;RETURN TO USER
 5165  040056  006203                2$:     ASR     R3              ;;POSITION THE MASK FOR THE LAST DIGIT
 5166  040060  006001                3$:     ROR     R1              ;;POSITION THE BINARY NUMBER FOR
 5167  040062  006000                        ROR     R0              ;;      THE NEXT OCTAL DIGIT
 5168  040064  006001                        ROR     R1
 5169  040066  006000                        ROR     R0
 5170  040070  006001                        ROR     R1
 5171  040072  006000                        ROR     R0
 5172  040074  040302                        BIC     R3,R2           ;;MASK OUT ALL JUNK
 5173  040076  062702  000060                ADD     #'0,R2          ;;MAKE THIS CHAR. ASCII
 5174  040102  000753                        BR      1$              ;;GO PUT IT IN THE DATA TABLE
 5175  040104  000016                $OCTVL: .BLKB   14.             ;;RESERVE DATA TABLE
 5176                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 5177
 5178                                  ;;**************************************************************
 5179                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 5180                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 5181                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 5182                                  ;*CALL:
 5183                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 5184                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
 5185                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
 5186                                  ;*      .BYTE   M               ;;M=1 OR 0
 5187                                  ;*                                      ;;1=TYPE LEADING ZEROS
 5188                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
 5189                                  ;*
 5190                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
 5191                                  ;*$TYPOS OR $TYPOC
 5192                                  ;*CALL:
 5193                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 5194                                  ;*      TYPON                   ;;CALL FOR TYPEOUT
 5195                                  ;*
 5196                                  ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
 5197                                  ;*CALL:
 5198                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 5199                                  ;*      TYPOC                   ;;CALL FOR TYPEOUT
 5200
 5201  040122  017646  000000        $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 5202  040126  116637  000001  040345        MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
 5203  040134  112637  040347                MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
 5204  040140  062716  000002                ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 5205  040144  000406                        BR      $TYPON
 5206  040146  112737  000001  040345 $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
 5207  040154  112737  000006  040347        MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
 5208  040162  112737  000005  040344 $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 5209  040170  010346                        MOV     R3,-(SP)        ;;SAVE R3
```

```
5210  040172  010446                        MOV     R4,-(SP)        ;;SAVE R4
5211  040174  010546                        MOV     R5,-(SP)        ;;SAVE R5
5212  040176  113704  040347                MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
5213  040202  005404                        NEG     R4
5214  040204  062704  000006                ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
5215  040210  110437  040346                MOVB    R4,$OMODE       ;;SAVE IT FOR USE
5216  040214  113704  040345                MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
5217  040220  016605  000012                MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
5218  040224  005003                        CLR     R3              ;;CLEAR THE OUTPUT WORD
5219  040226  006105                1$:     ROL     R5              ;;ROTATE MSB INTO "C"
5220  040230  000404                        BR      3$              ;;GO DO MSB
5221  040232  006105                2$:     ROL     R5              ;;FORM THIS DIGIT
5222  040234  006105                        ROL     R5
5223  040236  006105                        ROL     R5
5224  040240  010503                        MOV     R5,R3
5225  040242  006103                3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
5226  040244  105337  040346                DECB    $OMODE          ;;TYPE THIS DIGIT?
5227  040250  100016                        BPL     7$              ;;BR IF NO
5228  040252  042703  177770                BIC     #177770,R3      ;;GET RID OF JUNK
5229  040256  001002                        BNE     4$              ;;TEST FOR 0
5230  040260  005704                        TST     R4              ;;SUPPRESS THIS 0?
5231  040262  001403                        BEQ     5$              ;;BR IF YES
5232  040264  005204                4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
5233  040266  052703  000060                BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
5234  040272  052703  000040        5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
5235  040276  110337  040342                MOVB    R3,8$           ;;SAVE FOR TYPING
5236  040302  104401  040342                TYPE    ,8$             ;;GO TYPE THIS DIGIT
5237  040306  105337  040344        7$:     DECB    $OCNT           ;;COUNT BY 1
5238  040312  003347                        BGT     2$              ;;BR IF MORE TO DO
5239  040314  002402                        BLT     6$              ;;BR IF DONE
5240  040316  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
5241  040320  000744                        BR      2$              ;;GO DO THE LAST DIGIT
5242  040322  012605                6$:     MOV     (SP)+,R5        ;;RESTORE R5
5243  040324  012604                        MOV     (SP)+,R4        ;;RESTORE R4
5244  040326  012603                        MOV     (SP)+,R3        ;;RESTORE R3
5245  040330  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
5246  040336  012616                        MOV     (SP)+,(SP)
5247  040340  000002                        RTI                     ;;RETURN
5248  040342     000                8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
5249  040343     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
5250  040344     000                $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
5251  040345     000                $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
5252  040346  000000                $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
5253                                .SBTTL  ERROR HANDLER ROUTINE
5254
5255                                ;;**************************************************************
5256                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
5257                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
5258                                ;*AND GO TO TYPERR ON ERROR
5259                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
5260                                ;*SW15=1         HALT ON ERROR
5261                                ;*SW13=1         INHIBIT ERROR TYPEOUTS
5262                                ;*SW10=1         BELL ON ERROR
5263                                ;*SW09=1         LOOP ON ERROR
5264                                ;*CALL
5265                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
```

```
                                        $ERROR:
                                                CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
040350  104406                          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
040352  105237  001103                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
040356  001775                                  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
040360  013777  001102  140554                  BIT     #BIT10,@SWR     ;;BELL ON ERROR?
040366  032777  002000  140544                  BEQ     1$              ;;NO - SKIP
040374  001402                                  TYPE    ,$BELL          ;;RING BELL
040376  104401  001262                  1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
040402  005237  001112                          MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
040406  011637  001116                          SUB     #2,$ERRPC
040412  162737  000002  001116                  MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
040420  117737  140472  001114                  BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
040426  032777  020000  140504                  BNE     20$             ;;SKIP TYPEOUTS
040434  001004                                  JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
040436  004737  025362                          TYPE    ,$CRLF
040442  104401  001267
040446                                  20$:
040446  005777  140466                  2$:     TST     @SWR            ;;HALT ON ERROR
040452  100002                                  BPL     3$              ;;SKIP IF CONTINUE
040454  000000                                  HALT                    ;;HALT ON ERROR!
040456  104406                                  CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
040460  032777  001000  140452          3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
040466  001402                                  BEQ     4$              ;;BR IF NO
040470  013716  001110                          MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
040474  005737  001260                  4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
040500  001402                                  BEQ     5$              ;;BR IF NONE
040502  013716  001260                          MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
040506                                  5$:
040506  000002                                  RTI                     ;;RETURN
                                        .SBTTL  TTY INPUT ROUTINE

                                        ;;**************************************************************
                                        .ENABL  LSB
040510  000000                        $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
040512  000000                        $TKQIN: .WORD   0               ;;INPUT POINTER
040514  000000                        $TKQOUT: .WORD  0               ;;OUTPUT POINTER
040516  000001                        $TKQSRT: .BLKB  1               ;;TTY KEYBOARD QUEUE
        040517                          $TKQEND=.
        040520                          .EVEN

                                        ;*TK INITIALIZE ROUTINE
                                        ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
                                        ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
                                        ;
                                        ;*CALL:
                                        ;*      JSR     PC,$TKINT
                                        ;*      RETURN
                                        ;
040520  005037  040510          $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
040524  012737  040516  040512          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
040532  013737  040512  040514          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
040540  012737  040570  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
040546  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
040554  005777  140366                  TST     @$TKB           ;;CLEAR DONE FLAG
040560  012777  000100  140356          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
```

```
5322  040566  000207                        RTS     PC              ;;RETURN TO CALLER
5323
5324                                        ;*TK SERVICE ROUTINE
5325                                        ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
5326                                        ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
5327                                        ;*IT IN THE QUEUE.
5328                                        ;*IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
5329                                        ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (START)
5330                                        ;
5331  040570  117746  140352           $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
5332  040574  042716  177600                   BIC     #↑C177,(SP)     ;;STRIP THE JUNK
5333  040600  021627  000003                   CMP     (SP),#3         ;;IS IT A CONTROL C?
5334  040604  001007                           BNE     1$              ;;BRANCH IF NO
5335  040606  104401  041766                   TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (↑C)
5336  040612  004737  040520                   JSR     PC,$TKINT       ;;INIT THE KEYBOARD
5337  040616  005726                           TST     (SP)+           ;;CLEAN UP STACK
5338  040620  000137  016100                   JMP     START           ;;CONTROL C RESTART
5339  040624  021627  000007           1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
5340  040630  001004                           BNE     2$              ;;BRANCH IF NO
5341  040632  022737  000176  001140           CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
5342  040640  001500                           BEQ     6$              ;;GO TO SWR CHANGE
5343
5344  040642                           2$:
5345  040642  022737  000001  040510           CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
5346  040650  001004                           BNE     3$              ;;BRANCH IF NO
5347  040652  104401  001262                   TYPE    ,$BELL          ;;RING THE TTY BELL
5348  040656  005726                           TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
5349  040660  000451                           BR      5$              ;;EXIT
5350  040662  021627  000023           3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
5351  040666  001021                           BNE     32$             ;;BRANCH IF NO
5352  040670  005077  140250                   CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
5353  040674  005726                           TST     (SP)+           ;;CLEAN CHAR OFF STACK
5354  040676  105777  140242           31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
5355  040702  100375                           BPL     31$             ;;LOOP UNTIL ITS THERE
5356  040704  117746  140236                   MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
5357  040710  042716  177600                   BIC     #↑C177,(SP)     ;;MAKE IT 7-BIT ASCII
5358  040714  022627  000021                   CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
5359  040720  001366                           BNE     31$             ;;BRANCH IF NO
5360  040722  012777  000100  140214           MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
5361  040730  000002                           RTI                     ;;RETURN
5362  040732  005237  040510           32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
5363  040736  021627  000140                   CMP     (SP),#140       ;;IS IT UPPER CASE?
5364  040742  002405                           BLT     4$              ;;BRANCH IF YES
5365  040744  021627  000175                   CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
5366  040750  003002                           BGT     4$              ;;BRANCH IF YES
5367  040752  042716  000040                   BIC     #40,(SP)        ;;MAKE IT UPPER CASE
5368  040756  112677  177530           4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
5369  040762  005237  040512                   INC     $TKQIN          ;;UPDATE THE POINTER
5370  040766  023727  040512  040517           CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
5371  040774  001003                           BNE     5$              ;;BRANCH IF NO
5372  040776  012737  040516  040512           MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
5373  041004  000002                   5$:     RTI                     ;;RETURN
5374
5375                                        ;;**************************************************************
5376                                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
5377                                        ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
```

```
5378                                    :*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
5379                                    :*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
5380  041006  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
5381  041014  001124                          BNE     15$             ;;EXIT IF NOT
5382  041016  105777  140122                  TSTB    @$TKS           ;;IS A CHAR WAITING?
5383  041022  100121                          BPL     15$             ;;IF NOT, EXIT
5384  041024  117746  140116                  MOVB    @$TKB,-(SP)     ;;YES
5385  041030  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
5386  041034  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
5387  041040  001300                          BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
5388                                                                  ;;AND EXIT
5389
5390                                    ;;**************************************************************
5391                                    ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
5392                                    ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
5393                                    ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
5394  041042  123727  001134  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
5395  041050  001674                          BEQ     2$              ;;BRANCH IF YES
5396  041052  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
5397  041054  004737  040520                  JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
5398  041060  005077  140060                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
5399  041064  112737  000001  001135          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
5400
5401  041072  104401  042000                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
5402  041076  104401  042005          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
5403  041102  013746  000176                  MOV     $WREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
5404  041106  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
5405  041110  104401  042016                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
5406  041114  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
5407  041116  005046                          CLR     -(SP)           ;;THE NEW SWR
5408  041120  105777  140020          7$:     TSTB    @$TKS           ;;CHAR THERE?
5409  041124  100375                          BPL     7$              ;;IF NOT TRY AGAIN
5410
5411  041126  117746  140014                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
5412  041132  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
5413
5414  041136  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
5415  041142  001015                          BNE     9$              ;;BRANCH IF NOT
5416  041144  104401  041766                  TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
5417  041150  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
5418  041154  123727  001135  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
5419  041162  001003                          BNE     8$              ;;BRANCH IF NO
5420  041164  012777  000100  137752          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
5421  041172  000137  016100          8$:     JMP     START           ;;CONTROL-C RESTART
5422
5423
5424  041176  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
5425  041202  001005                          BNE     10$             ;;BRANCH IF NOT
5426  041204  104401  041773                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
5427  041210  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
5428  041214  000737                          BR      19$             ;;LET'S TRY IT AGAIN
5429
5430
5431  041216  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
5432  041222  001022                          BNE     16$             ;;BRANCH IF NO
5433  041224  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
```

```
                                        BEQ     11$             ;;BRANCH IF YES
                                        MOV     2(SP),@SWR      ;;SAVE NEW SWR
                                11$:    ADD     #6,SP           ;;CLEAR UP STACK
                                14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
                                        CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
                                        BNE     15$             ;;BRANCH IF NOT
                                        MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
                                15$:    RTI                     ;;RETURN
                                16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
                                        CMP     (SP),#60        ;;CHAR < 0?
                                        BLT     18$             ;;BRANCH IF YES
                                        CMP     (SP),#67        ;;CHAR > 7?
                                        BGT     18$             ;;BRANCH IF YES
                                        BIC     #60,(SP)+       ;;STRIP-OFF ASCII
                                        TST     2(SP)           ;;IS THIS THE FIRST CHAR
                                        BEQ     17$             ;;BRANCH IF YES
                                        ASL     (SP)            ;;NO, SHIFT PRESENT
                                        ASL     (SP)            ;;   CHAR OVER TO MAKE
                                        ASL     (SP)            ;;   ROOM FOR NEW ONE.
                                17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
                                        BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
                                        BR      7$              ;;GET THE NEXT ONE
                                18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
                                        BR      20$             ;;SIMULATE CONTROL-U
                                .DSABL  LSB

;;*****************************************************************
;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
;*CALL:
;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
;*                              ;;WITH PARITY BIT STRIPPED OFF
;

041350  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
041352  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
041360  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
041364  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
041366  012746  041374                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
041372  000002                          RTI                     ;;POP NEW PC AND PS
041374                          64$:
041374  005737  040510          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
041400  001775                          BEQ     1$
041402  005337  040510                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
041406  117766  177102  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
041414  005237  040514                  INC     $TKQOUT         ;;UPDATE THE POINTER
041420  023727  040514  040517          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
041426  001003                          BNE     2$              ;;BRANCH IF NO
041430  012737  040516  040514          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
041436  000002                  2$:     RTI                     ;;RETURN
;;*****************************************************************
;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
;*CALL:
;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
```

```
5490                                    :*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
5491
5492  041440  010346                    $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
5493  041442  005046                            CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
5494  041444  012703  041674            1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
5495  041450  022703  041766            2$:     CMP     #$TTYIN+72,R3   ;;BUFFER FULL?
5496  041454  101456                            BLOS    4$              ;;BR IF YES
5497  041456  104407                            RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
5498  041460  112613                            MOVB    (SP)+,(R3)      ;;GET CHARACTER
5499  041462  122713  000177            10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
5500  041466  001022                            BNE     5$              ;;BR IF NO
5501  041470  005716                            TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
5502  041472  001007                            BNE     6$              ;;BR IF NO
5503  041474  112737  000134  041672          MOVB    #'\,9$          ;;TYPE A BACK SLASH
5504  041502  104401  041672                    TYPE    ,9$
5505  041506  012716  177777                    MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
5506  041512  005303                    6$:     DEC     R3              ;;BACKUP BY ONE
5507  041514  020327  041674                    CMP     R3,#$TTYIN      ;;STACK EMPTY?
5508  041520  103434                            BLO     4$              ;;BR IF YES
5509  041522  111337  041672                    MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
5510  041526  104401  041672                    TYPE    ,9$             ;;GO TYPE
5511  041532  000746                            BR      2$              ;;GO READ ANOTHER CHAR.
5512  041534  005716                    5$:     TST     (SP)            ;;RUBOUT KEY SET?
5513  041536  001406                            BEQ     7$              ;;BR IF NO
5514  041540  112737  000134  041672          MOVB    #'\,9$          ;;TYPE A BACK SLASH
5515  041546  104401  041672                    TYPE    ,9$
5516  041552  005016                            CLR     (SP)            ;;CLEAR THE RUBOUT KEY
5517  041554  122713  000025            7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
5518  041560  001003                            BNE     8$              ;;BR IF NO
5519  041562  104401  041773                    TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
5520  041566  000726                            BR      1$              ;;GO START OVER
5521  041570  122713  000022            8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "^R"?
5522  041574  001011                            BNE     3$              ;;BRANCH IF NO
5523  041576  105013                            CLRB    (R3)            ;;CLEAR THE CHARACTER
5524  041600  104401  001267                    TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
5525  041604  104401  041674                    TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
5526  041610  000717                            BR      2$              ;;GO PICKUP ANOTHER CHACTER
5527  041612  104401  001266            4$:     TYPE    ,$QUES          ;;TYPE A '?'
5528  041616  000712                            BR      1$              ;;CLEAR THE BUFFER AND LOOP
5529  041620  111337  041672            3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
5530  041624  104401  041672                    TYPE    ,9$
5531  041630  122723  000015                    CMPB    #15,(R3)+       ;;CHECK FOR RETURN
5532  041634  001305                            BNE     2$              ;;LOOP IF NOT RETURN
5533  041636  105063  177777                    CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
5534  041642  104401  001270                    TYPE    ,$LF            ;;TYPE A LINE FEED
5535  041646  005726                            TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
5536  041650  012603                            MOV     (SP)+,R3        ;;RESTORE R3
5537  041652  011646                            MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
5538  041654  016666  000004  000002            MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
5539  041662  012766  041674  000004            MOV     #$TTYIN,4(SP)
5540  041670  000002                            RTI                     ;;RETURN
5541  041672     000                    9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
5542  041673     000                            .BYTE   0               ;;TERMINATOR
5543  041674  000072                    $TTYIN: .BLKB   72              ;;RESERVE 72 BYTES FOR TTY INPUT
5544  041766  041536  005015     000    $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL "C"
5545  041773     136  006525  000012    $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
```

```
5546  042000  043536  005015     000  $CNTLG: .RSCIZ  /^G/<15><12>     ;;CONTROL "G"
5547  042005     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
5548  042012  036440  000040
5549  042016  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
5550  042024  020075     000
5551          042030                  .EVEN
5552                                  .SBTTL  POWER DOWN AND UP ROUTINES
5553
5554                                  ;;*****************************************************************
5555                                  ;POWER DOWN ROUTINE
5556  042030  012737  042170  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
5557  042036  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
5558  042044  010046                  MOV     R0,-(SP)         ;;PUSH R0 ON STACK
5559  042046  010146                  MOV     R1,-(SP)         ;;PUSH R1 ON STACK
5560  042050  010246                  MOV     R2,-(SP)         ;;PUSH R2 ON STACK
5561  042052  010346                  MOV     R3,-(SP)         ;;PUSH R3 ON STACK
5562  042054  010446                  MOV     R4,-(SP)         ;;PUSH R4 ON STACK
5563  042056  010546                  MOV     R5,-(SP)         ;;PUSH R5 ON STACK
5564  042060  017746  137054          MOV     @SWR,-(SP)       ;;PUSH @SWR ON STACK
5565  042064  010637  042174          MOV     SP,$SAVR6        ;;SAVE SP
5566  042070  012737  042102  000024  MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
5567  042076  000000                  HALT
5568  042100  000776                  BR      .-2              ;;HANG UP
5569
5570                                  ;;*****************************************************************
5571                                  ;POWER UP ROUTINE
5572  042102  012737  042170  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
5573  042110  013706  042174          MOV     $SAVR6,SP        ;;GET SP
5574  042114  005037  042174          CLR     $SAVR6           ;;WAIT LOOP FOR THE TTY
5575  042120  005237  042174  1$:     INC     $SAVR6           ;;WAIT FOR THE INC
5576  042124  001375                  BNE     1$               ;;OF  WORD
5577  042126  012677  137006          MOV     (SP)+,@SWR       ;;POP STACK INTO @SWR
5578  042132  012605                  MOV     (SP)+,R5         ;;POP STACK INTO R5
5579  042134  012604                  MOV     (SP)+,R4         ;;POP STACK INTO R4
5580  042136  012603                  MOV     (SP)+,R3         ;;POP STACK INTO R3
5581  042140  012602                  MOV     (SP)+,R2         ;;POP STACK INTO R2
5582  042142  012601                  MOV     (SP)+,R1         ;;POP STACK INTO R1
5583  042144  012600                  MOV     (SP)+,R0         ;;POP STACK INTO R0
5584  042146  012737  042030  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
5585  042154  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
5586  042162  104401                  TYPE                     ;REPORT THE POWER FAILURE
5587  042164  042176          $PWRMG: .WORD   $POWER           ;;POWER FAIL MESSAGE POINTER
5588  042166  000002                  RTI
5589  042170  000000          $ILLUP: HALT                     ;;THE POWER UP SEQUENCE WAS STARTED
5590  042172  000776                  BR      .-2              ;; BEFORE THE POWER DOWN WAS COMPLETE
5591  042174  000000          $SAVR6: 0                        ;;PUT THE SP HERE
5592  042176  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
5593  042204  000122
5594                                  .EVEN
5595                                  .SBTTL  ROUTINE TO SIZE MEMORY
5596
5597                                  ;;*****************************************************************
5598                                  ;*CALL:
5599                                  ;*      JSR     PC,$SIZE
5600                                  ;*      RETURN
5601                                  ;*$LSTAD WILL CONTAIN THE LAST AVAILABLE MEMORY LOCATION
```

```

 042206  010046                        $SIZE:  MOV     R0,-(SP)               ;;SAVE R0 ON THE STACK
 042210  010146                                MOV     R1,-(SP)               ;;SAVE R1 ON THE STACK
 042212  013746  000004                        MOV     @#ERRVEC,-(SP)         ;;SAVE PRESENT ERROR VECTOR PS & PC
 042216  013746  000006                        MOV     @#ERRVEC+2,-(SP)
 042222  010600                                MOV     SP,R0                  ;;SAVE THE STACK POINTER
                                       ;;SET THE ERRVEC PS TO THE PRESENT PS
 042224  104400                                TRAP                           ;;PUSH OLD PSW AND PC ON STACK
 042226  012637  000006                        MOV     (SP)+,@#ERRVEC+2           ;;SAVE THE PSW IN @#ERRVEC+2
 042232  012737  042252  000004                MOV     #2$,@#ERRVEC           ;;SET FOR TIMEOUT
 042240  012701  020000                        MOV     #20000,R1              ;;FIRST ADDRESS
 042244  005711                        1$:     TST     (R1)                   ;;TEST THIS ADDRESS
 042246  005721                                TST     (R1)+                  ;;STEP TO NEXT ADDRESS
 042250  000775                                BR      1$                     ;;TRY ANOTHER
 042252  162701  000002                2$:     SUB     #2,R1                  ;;DROP BACK
 042256  010006                                MOV     R0,SP                  ;;RESTORE THE STACK
 042260  012637  000006                        MOV     (SP)+,@#ERRVEC+2 ;;RESTORE ERROR VECTOR
 042264  012637  000004                        MOV     (SP)+,@#ERRVEC
 042270  010137  042302                        MOV     R1,$LSTAD              ;;LAST ADDRESS
 042274  012601                                MOV     (SP)+,R1               ;;RESTORE R1
 042276  012600                                MOV     (SP)+,R0               ;;RESTORE R0
 042300  000207                                RTS     PC
 042302  000000                        $LSTAD: .WORD   0                      ;;CONTAINS THE LAST ADDRESS
                                       .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES

                                       ;;*******************************************************************
                                       ;*SAVE R0-R5
                                       ;*CALL:
                                       ;*       SAVREG
                                       ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
                                       ;*
                                       ;*TOP---(+16)
                                       ;* +2---(+18)
                                       ;* +4---R5
                                       ;* +6---R4
                                       ;* +8---R3
                                       ;*+10---R2
                                       ;*+12---R1
                                       ;*+14---R0

 042304                                $SAVREG:
 042304  010046                                MOV     R0,-(SP)               ;;PUSH R0 ON STACK
 042306  010146                                MOV     R1,-(SP)               ;;PUSH R1 ON STACK
 042310  010246                                MOV     R2,-(SP)               ;;PUSH R2 ON STACK
 042312  010346                                MOV     R3,-(SP)               ;;PUSH R3 ON STACK
 042314  010446                                MOV     R4,-(SP)               ;;PUSH R4 ON STACK
 042316  010546                                MOV     R5,-(SP)               ;;PUSH R5 ON STACK
 042320  016646  000022                        MOV     22(SP),-(SP)           ;;SAVE PS OF MAIN FLOW
 042324  016646  000022                        MOV     22(SP),-(SP)           ;;SAVE PC OF MAIN FLOW
 042330  016646  000022                        MOV     22(SP),-(SP)           ;;SAVE PS OF CALL
 042334  016646  000022                        MOV     22(SP),-(SP)           ;;SAVE PC OF CALL
 042340  000002                                RTI

                                       ;*RESTORE R0-R5
                                       ;*CALL:
                                       ;*       RESREG
```

```
5658  042342                          $RESREG:
5659  042342  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
5660  042346  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
5661  042352  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
5662  042356  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
5663  042362  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
5664  042364  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
5665  042366  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
5666  042370  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
5667  042372  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
5668  042374  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
5669  042376  000002                          RTI
5670                                  .SBTTL  TRAP DECODER
5671
5672                                  ;;**************************************************************
5673                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
5674                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
5675                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
5676                                  ;*GO TO THAT ROUTINE.
5677
5678  042400  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
5679  042402  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
5680  042406  005740                          TST     -(R0)           ;;BACKUP BY 2
5681  042410  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
5682  042412  006300                          ASL     R0              ;;POSITION FOR INDEXING
5683  042414  016000  042434                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
5684  042420  000200                          RTS     R0              ;;GO TO ROUTINE
5685
5686
5687                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
5688
5689  042422  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
5690  042424  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
5691  042432  000002                          RTI                     ;;RESTORE THE PSW
5692
5693                                  .SBTTL  TRAP TABLE
5694
5695                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
5696                                  ;*BY THE "TRAP" INSTRUCTION.
5697
5698                                  ;       ROUTINE
5699                                  ;       -------
5700  042434  042422                  $TRPAD: .WORD   $TRAP2
5701  042436  037562                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
5702  042440  040146                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
5703  042442  040122                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
5704  042444  040162                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
5705
5706  042446  041076                          $GTSWR  ;;CALL=GTSWR    TRAP+5(104405)  GET SOFT-SWR SETTING
5707
5708  042450  041006                          $CKSWR  ;;CALL=CKSWR    TRAP+6(104406)  TEST FOR CHANGE IN SOFT-SWR
5709  042452  041350                          $RDCHR  ;;CALL=RDCHR    TRAP+7(104407)  TTY TYPEIN CHARACTER ROUTINE
5710  042454  041440                          $RDLIN  ;;CALL=RDLIN    TRAP+10(104410) TTY TYPEIN STRING ROUTINE
5711  042456  042304                          $SAVREG ;;CALL=SAVREG   TRAP+11(104411) SAVE R0-R5 ROUTINE
5712  042460  042342                          $RESREG ;;CALL=RESREG   TRAP+12(104412) RESTORE R0-R5 ROUTINE
5713
```

```
5714  042462  025052  020040  047125  EM1:    .ASCIZ  /** UNIBUS PARITY ERROR/
5715  042470  041111  051525  050040
5716  042476  051101  052111  020131
5717  042504  051105  047522  000122
5718  042512  025052  020040  047516  EM2:    .ASCIZ  /** NON-EXISTANT MEMORY ERROR/
5719  042520  026516  054105  051511
5720  042526  040524  052116  046440
5721  042534  046505  051117  020131
5722  042542  051105  047522  000122
5723  042550  025052  020040  047516  EM3:    .ASCIZ  /** NON-EXISTANT DRIVE ERROR/
5724  042556  026516  054105  051511
5725  042564  040524  052116  042040
5726  042572  044522  042526  042440
5727  042600  051122  051117     000
5728  042605     052  020052  052440  EM4:    .ASCIZ  /** UNIT FIELD ERROR/
5729  042612  044516  020124  044506
5730  042620  046105  020104  051105
5731  042626  047522  000122
5732  042632  025052  020040  052523  EM5:    .ASCIZ  /** SUBSYSTEM TIMEOUT/
5733  042640  051502  051531  042524
5734  042646  020115  044524  042515
5735  042654  052517  000124
5736  042660  025052  020040  051104  EM6:    .ASCIZ  /** DRIVE BUS PARITY ERROR/
5737  042666  053111  020105  052502
5738  042674  020123  040520  044522
5739  042702  054524  042440  051122
5740  042710  051117     000
5741  042713     052  020052  042040  EM7:    .ASCIZ  /** DRIVE DETECTED PARITY ERROR/
5742  042720  044522  042526  042040
5743  042726  052105  041505  042524
5744  042734  020104  040520  044522
5745  042742  054524  042440  051122
5746  042750  051117     000
5747  042753     052  020052  040440  EM10:   .ASCIZ  /** AC LOW/
5748  042760  020103  047514  000127
5749  042766  025052  020040  050123  EM11:   .ASCIZ  /** SPEED LOSS/
5750  042774  042505  020104  047514
5751  043002  051523     000
5752  043005     052  020052  044440  EM12:   .ASCIZ  /** ILLEGAL FUNCTION ERROR/
5753  043012  046114  043505  046101
5754  043020  043040  047125  052103
5755  043026  047511  020116  051105
5756  043034  047522  000122
5757  043040  025052  020040  051120  EM13:   .ASCIZ  /** PROGRAMMING ERROR/
5758  043046  043517  040522  046515
5759  043054  047111  020107  051105
5760  043062  047522  000122
5761  043066  025052  020040  047516  EM14:   .ASCIZ  /** NON-EXISTANT FUNCTION ERROR/
5762  043074  026516  054105  051511
5763  043102  040524  052116  043040
5764  043110  047125  052103  047511
5765  043116  020116  051105  047522
5766  043124  000122
5767  043126  025052  020040  051104  EM15:   .ASCIZ  /** DRIVE TYPE ERROR/
5768  043134  053111  020105  054524
5769  043142  042520  042440  051122
```

```
5770  043150  051117     000
5771  043153     052  020052  043040  EM16:   .ASCIZ  /** FORMAT ERROR/
5772  043160  051117  040515  020124
5773  043166  051105  047522  000122
5774  043174  025052  020040  051127  EM17:   .ASCIZ  /** WRITE LOCK ERROR/
5775  043202  052111  020105  047514
5776  043210  045503  042440  051122
5777  043216  051117     000
5778  043221     052  020052  042040  EM20:   .ASCIZ  /** DRIVE UNSAFE ERROR/
5779  043226  044522  042526  052440
5780  043234  051516  043101  020105
5781  043242  051105  047522  000122
5782  043250  025052  020040  042523  EM21:   .ASCIZ  /** SEEK INCOMPLETE ERROR/
5783  043256  045505  044440  041516
5784  043264  046517  046120  052105
5785  043272  020105  051105  047522
5786  043300  000122
5787  043302  025052  020040  054503  EM22:   .ASCIZ  /** CYLINDER OVERFLOW ERROR/
5788  043310  044514  042116  051105
5789  043316  047440  042526  043122
5790  043324  047514  020127  051105
5791  043332  047522  000122
5792  043336  025052  020040  046111  EM23:   .ASCIZ  /** ILLEGAL CYLINDER ADDRESS ERROR/
5793  043344  042514  040507  020114
5794  043352  054503  044514  042116
5795  043360  051105  040440  042104
5796  043366  042522  051523  042440
5797  043374  051122  051117     000
5798  043401     052  020052  042040  EM24:   .ASCIZ  /** DRIVE OFF TRACK/
5799  043406  044522  042526  047440
5800  043414  043106  052040  040522
5801  043422  045503     000
5802  043425     052  020052  042040  EM25:   .ASCIZ  /** DRIVE TIMING ERROR/
5803  043432  044522  042526  052040
5804  043440  046511  047111  020107
5805  043446  051105  047522  000122
5806  043454  025052  020040  040504  EM26:   .ASCIZ  /** DATA LATE ERROR/
5807  043462  040524  046040  052101
5808  043470  020105  051105  047522
5809  043476  000122
5810  043500  025052  020040  047503  EM27:   .ASCIZ  /** CONTROLLER TIMEOUT ERROR/
5811  043506  052116  047522  046114
5812  043514  051105  052040  046511
5813  043522  047505  052125  042440
5814  043530  051122  051117     000
5815  043535     052  020052  047440  EM30:   .ASCIZ  /** OPERATION INCOMPLETE ERROR/
5816  043542  042520  040522  044524
5817  043550  047117  044440  041516
5818  043556  046517  046120  052105
5819  043564  020105  051105  047522
5820  043572  000122
5821  043574  025052  020040  042510  EM31:   .ASCIZ  /** HEADER VRC ERROR/
5822  043602  042101  051105  053040
5823  043610  041522  042440  051122
5824  043616  051117     000
5825  043621     052  020052  042040  EM32:   .ASCIZ  /** DATA CHECK ERROR/
```

```
5826  043626  052101  020101  044103
5827  043634  041505  020113  051105
5828  043642  047522  000122
5829  043646  025052  020040  051127  EM33:   .ASCIZ  /** WRITE CHECK ERROR/
5830  043654  052111  020105  044103
5831  043662  041505  020113  051105
5832  043670  047522  000122
5833  043674  025052  020040  040504  EM34:   .ASCIZ  /** DATA MISCOMPARE/
5834  043702  040524  046440  051511
5835  043710  047503  050115  051101
5836  043716  000105
5837  043720  025052  020040  047516  EM35:   .ASCIZ  /** NO DRIVE RESPONSE-UFE & NED/
5838  043726  042040  044522  042526
5839  043734  051040  051505  047520
5840  043742  051516  026505  043125
5841  043750  020105  020046  042516
5842  043756  000104
5843  043760  025052  020040  051104  EM36:   .ASCIZ  /** DRIVE ERROR WILL NOT CLEAR/
5844  043766  053111  020105  051105
5845  043774  047522  020122  044527
5846  044002  046114  047040  052117
5847  044010  041440  042514  051101
5848  044016     000
5849  044017     052  020052  042040  EM37:   .ASCIZ  /** DRIVE STATUS CHANGE WILL NOT CLEAR/
5850  044024  044522  042526  051440
5851  044032  040524  052524  020123
5852  044040  044103  047101  042507
5853  044046  053440  046111  020114
5854  044054  047516  020124  046105
5855  044062  040505  000122
5856  044066  025052  020040  052101  EM40:   .ASCIZ  /** ATTENTION BUT NO STATUS CHANGE OR FAULT/
5857  044074  042524  052116  047511
5858  044102  020116  052502  020124
5859  044110  047516  051440  040524
5860  044116  052524  020123  044103
5861  044124  047101  042507  047440
5862  044132  020122  040506  046125
5863  044140  000124
5864  044142  025052  020040  052101  EM41:   .ASCIZ  /** ATTENTION BUT DRIVE NOT AVAILABLE/
5865  044150  042524  052116  047511
5866  044156  020116  052502  020124
5867  044164  051104  053111  020105
5868  044172  047516  020124  053101
5869  044200  044501  040514  046102
5870  044206  000105
5871  044210  025052  020040  052101  EM42:   .ASCIZ  /** ATTENTION WHEN NOT EXPECTED/
5872  044216  042524  052116  047511
5873  044224  020116  044127  047105
5874  044232  047040  052117  042440
5875  044240  050130  041505  042524
5876  044246  000104
5877  044250  025052  020040  051105  EM43:   .ASCIZ  /** ERROR WHILE GATHERING DRIVE STATUS/
5878  044256  047522  020122  044127
5879  044264  046111  020105  040507
5880  044272  044124  051105  047111
5881  044300  020107  051104  053111
```

```
 5882  044306  020105  052123  052101
 5883  044314  051525     000
 5884  044317     052  020052  046440  EM52:   .ASCIZ  /** MULTIPLE DRIVE SELECT/
 5885  044324  046125  044524  046120
 5886  044332  020105  051104  053111
 5887  044340  020105  042523  042514
 5888  044346  052103     000
 5889  044351     052  020052  044040  EM53:   .ASCIZ  /** HEADER COMPARE ERROR/
 5890  044356  040505  042504  020122
 5891  044364  047503  050115  051101
 5892  044372  020105  051105  047522
 5893  044400  000122
 5894  044402  025052  020040  052523  EM56:   .ASCIZ  /** SUBSYSTEM TIMEOUT/
 5895  044410  051502  051531  042524
 5896  044416  020115  044524  042515
 5897  044424  052517  000124
 5898  044430  025052  020040  051105  EM60:   .ASCIZ  /** ERROR IN RECALIBRATE FOR RECOVERY/
 5899  044436  047522  020122  047111
 5900  044444  051040  041505  046101
 5901  044452  041111  040522  042524
 5902  044460  043040  051117  051040
 5903  044466  041505  053117  051105
 5904  044474  000131
 5905  044476  025052  020040  051120  EM61:   .ASCIZ  /** PROGRAM ABORTING FATAL ERROR IN RETRY/
 5906  044504  043517  040522  020115
 5907  044512  041101  051117  044524
 5908  044520  043516  043040  052101
 5909  044526  046101  042440  051122
 5910  044534  051117  044440  020116
 5911  044542  042522  051124  000131
 5912  044550  025052  020040  042510  EM62:   .ASCIZ  /** HEADER MISCOMPARE/
 5913  044556  042101  051105  046440
 5914  044564  051511  047503  050115
 5915  044572  051101  000105
 5916  044576  025052  020040  046103  EM63:   .ASCIZ  /** CLEAR CONTROLLER DID NOT CLEAR ERROR/
 5917  044604  040505  020122  047503
 5918  044612  052116  047522  046114
 5919  044620  051105  042040  042111
 5920  044626  047040  052117  041440
 5921  044634  042514  051101  042440
 5922  044642  051122  051117     000
 5923  044647     052  020052  047040  EM64:   .ASCIZ  /** NO ATTENTION IN ATTENTION SUMMARY REGISTER/
 5924  044654  020117  052101  042524
 5925  044662  052116  047511  020116
 5926  044670  047111  040440  052124
 5927  044676  047105  044524  047117
 5928  044704  051440  046525  040515
 5929  044712  054522  051040  043505
 5930  044720  051511  042524  000122
 5931  044726  025052  020040  047125  EM65:   .ASCIZ  /** UNSOLICITED ATTENTION/
 5932  044734  047523  044514  044503
 5933  044742  042524  020104  052101
 5934  044750  042524  052116  047511
 5935  044756  000116
 5936  044760  025052  020040  047125  EM66:   .ASCIZ  /** UNEXPECTED DATA TYPE ERROR/
 5937  044766  054105  042520  052103
```

```
 5938  044774  042105  042040  052101
 5939  045002  020101  054524  042520
 5940  045010  042440  051122  051117
 5941  045016     000
 5942  045017     052  020052  040440  EM67:   .ASCIZ  /** ATTENTION DID NOT RESET WITH CLEAR/
 5943  045024  052124  047105  044524
 5944  045032  047117  042040  042111
 5945  045040  047040  052117  051040
 5946  045046  051505  052105  053440
 5947  045054  052111  020110  046103
 5948  045062  040505  000122
 5949  045066  025052  020040  052523  EM70:   .ASCIZ  /** SUBSYSTEM CLEAR DID NOT CLEAR DRIVE ATTENTION/
 5950  045074  051502  051531  042524
 5951  045102  020115  046103  040505
 5952  045110  020122  044504  020104
 5953  045116  047516  020124  046103
 5954  045124  040505  020122  051104
 5955  045132  053111  020105  052101
 5956  045140  042524  052116  047511
 5957  045146  000116
 5958  045150  025052  020040  040504  EM71:   .ASCIZ  /** DATA LATE WHEN UNLOADING HEADER/
 5959  045156  040524  046040  052101
 5960  045164  020105  044127  047105
 5961  045172  052440  046116  040517
 5962  045200  044504  043516  044040
 5963  045206  040505  042504  000122
 5964  045214  025052  020040  047503  EM72:   .ASCIZ  /** CONTROLLER ERROR WHEN DRIVER SERVICING/
 5965  045222  052116  047522  046114
 5966  045230  051105  042440  051122
 5967  045236  051117  053440  042510
 5968  045244  020116  051104  053111
 5969  045252  051105  051440  051105
 5970  045260  044526  044503  043516
 5971  045266     000
 5972  045267     052  020052  042040  EM73:   .ASCIZ  /** DRIVE DETECTED PARITY ERROR/
 5973  045274  044522  042526  042040
 5974  045302  052105  041505  042524
 5975  045310  020104  040520  044522
 5976  045316  054524  042440  051122
 5977  045324  051117     000
 5978  045327     052  020052  052440  EM74:   .ASCIZ  /** UNDEFINED ERROR/
 5979  045334  042116  043105  047111
 5980  045342  042105  042440  051122
 5981  045350  051117     000
 5982  045353     052  020052  046440  EM75:   .ASCIZ  /** MARKING THIS SECTOR BAD/
 5983  045360  051101  044513  043516
 5984  045366  052040  044510  020123
 5985  045374  042523  052103  051117
 5986  045402  041040  042101     000
 5987  045407     052  020052  041040  EM76:   .ASCIZ  /** BAD DATA VERIFICATION WITH READ. ECC OF LAST RETRY IS:/
 5988  045414  042101  042040  052101
 5989  045422  020101  042526  044522
 5990  045430  044506  040503  044524
 5991  045436  047117  053440  052111
 5992  045444  020110  042522  042101
 5993  045452  020056  041505  020103
```

```
 5994  045460  043117  046040  051501
 5995  045466  020124  042522  051124
 5996  045474  020131  051511  000072
 5997  045502  025052  020040  042522  EM77:   .ASCIZ  /**  RETRY SUCCESSFUL/
 5998  045510  051124  020131  052523
 5999  045516  041503  051505  043123
 6000  045524  046125     000
 6001  045527     052  020052  051040  EM100:  .ASCIZ  /**  RETRY UNSUCCESSFUL/
 6002  045534  052105  054522  052440
 6003  045542  051516  041525  042503
 6004  045550  051523  052506  000114
 6005  045556  025052  020040  040503  EM101:  .ASCIZ  /**  CANNOT FIND A VALID HEADER IN TRACK JUST READ/
 6006  045564  047116  052117  043040
 6007  045572  047111  020104  020101
 6008  045600  040526  044514  020104
 6009  045606  042510  042101  051105
 6010  045614  044440  020116  051124
 6011  045622  041501  020113  052512
 6012  045630  052123  051040  040505
 6013  045636  000104
 6014  045640  025052  020040  040502  EM102:  .ASCIZ  /**  BAD SECTOR ERROR ON SECTOR NOT LISTED BAD/
 6015  045646  020104  042523  052103
 6016  045654  051117  042440  051122
 6017  045662  051117  047440  020116
 6018  045670  042523  052103  051117
 6019  045676  047040  052117  046040
 6020  045704  051511  042524  020104
 6021  045712  040502  000104
 6022  045716  025052  053440  051117  EM103:  .ASCIZ  /** WORD COUNT INCORRECT TO CONTINUE/
 6023  045724  020104  047503  047125
 6024  045732  020124  047111  047503
 6025  045740  051122  041505  020124
 6026  045746  047524  041440  047117
 6027  045754  044524  052516  000105
 6028  045762  025052  040503  052125  EM104:  .ASCIZ  /**CAUTION** DRIVE STATUS REPORTED MAY NOT BE CORRECT/
 6029  045770  047511  025116  020052
 6030  045776  051104  053111  020105
 6031  046004  052123  052101  051525
 6032  046012  051040  050105  051117
 6033  046020  042524  020104  040515
 6034  046026  020131  047516  020124
 6035  046034  042502  041440  051117
 6036  046042  042522  052103     000
 6037  046047     105  051122  051117  EM105:  .ASCII  /ERROR IN SECTOR ALREADY MARKED BAD/<15><12>
 6038  046054  044440  020116  042523
 6039  046062  052103  051117  040440
 6040  046070  051 4  040505  054504
 6041  046076  046440  051101  042513
 6042  046104  020104  040502  006504
 6043  046112     012
 6044  046113     123  051525  042520          .ASCII  /SUSPECT PHYSICAL DAMAGE IN HEADER AREA/<15><12>
 6045  046120  052103  050040  054510
 6046  046126  0 523  040503  020114
 6047  046134  0 504  040515  042507
 6048  046142  044440  020116  042510
 6049  046150  042101  051105  040440
```

```
6050  046156  042522  006501     012
6051  046163     106  051117  040515          .ASCIZ  /FORMAT ABORTING/
6052  046170  020124  041101  051117
6053  046176  044524  043516     000
6054  046203     103  046517  040515  DH100:  .ASCIZ  /COMMAND START VALUES/
6055  046210  042116  051440  040524
6056  046216  052122  053040  046101
6057  046224  042525  000123
6058  046230  045522  030466  020061  DH200:  .ASCIZ  /RK611 REGISTERS/
6059  046236  042522  044507  052123
6060  046244  051105  000123
6061  046250  042522  040515  047111  DH500:  .ASCIZ  /REMAINING REGISTERS NOT VALID/
6062  046256  047111  020107  042522
6063  046264  044507  052123  051105
6064  046272  020123  047516  020124
6065  046300  040526  044514  000104
6066  046306  044124  020105  047506  DH501:  .ASCIZ  /THE FOLLOWING REGISTER DATA MAY BE INCORRECT/
6067  046314  046114  053517  047111
6068  046322  020107  042522  044507
6069  046330  052123  051105  042040
6070  046336  052101  020101  040515
6071  046344  020131  042502  044440
6072  046352  041516  051117  042522
6073  046360  052103     000
6074  046363     105  051122  050040  DH101:  .ASCIZ  /ERR PC  DRIVE   CMND    CYLNDR  TRACK   SECTOR   WD CNT  BUFFER/
6075  046370  020103  042040  044522
6076  046376  042526  020040  041440
6077  046404  047115  020104  020040
6078  046412  041440  046131  042116
6079  046420  020122  052040  040522
6080  046426  045503  020040  051440
6081  046434  041505  047524  020122
6082  046442  020040  042127  041440
6083  046450  052116  020040  052502
6084  046456  043106  051105     000
6085  046463     122  041513  030523  DH201:  .ASCIZ  /RKCS1   RKCS2   RKDC    RKDA    RKWCR  RKBA    RKASOF/
6086  046470  020040  051040  041513
6087  046476  031123  020040  051040
6088  046504  042113  020103  020040
6089  046512  051040  042113  020101
6090  046520  020040  051040  053513
6091  046526  051103  020040  045522
6092  046534  040502  020040  020040
6093  046542  045522  051501  043117
6094  046550     000
6095  046551     122  042113  020123  DH202:  .ASCIZ  /RKDS    RKER/
6096  046556  020040  051040  042513
6097  046564  000122
6098  046566  030101  020060  020040  DH203:  .ASCIZ  /A00     B00     A01     B01     A02     B02     A03     B03/
6099  046574  020040  030102  020060
6100  046602  020040  020040  030101
6101  046610  020061  020040  020040
6102  046616  030102  020061  020040
6103  046624  020040  030101  020062
6104  046632  020040  020040  030102
6105  046640  020062  020040  020040
```

```
6106  046646  030101  020063  020040
6107  046654  020040  030102  000063
6108  046662  042510  042101  051105  DH601:  .ASCIZ  /HEADER SHOULD BE:/
6109  046670  051440  047510  046125
6110  046676  020104  042502  000072
6111  046704  042510  042101  051105  DH602:  .ASCIZ  /HEADER OF SECTOR 0 THIS CYLINDER IS:/
6112  046712  047440  020106  042523
6113  046720  052103  051117  030040
6114  046726  052040  044510  020123
6115  046734  054503  044514  042116
6116  046742  051105  044440  035123
6117  046750     000
6118  046751     123  040505  041522  DH603:  .ASCIZ  /SEARCHING FOR HEADEP:/
6119  046756  044510  043516  043040
6120  046764  051117  044040  040505
6121  046772  042504  035122     000
6122  046777     120  041501  020113  DH604:  .ASCIZ  /PACK ADDRESS OF ERROR IS:/
6123  047004  042101  051104  051505
6124  047012  020123  043117  042440
6125  047020  051122  051117  044440
6126  047026  035123     000
6127  047031     103  046131  042116  DH6041: .ASCIZ  /CYLNDR  TRACK   SECTOR/
6128  047036  020122  052040  040522
6129  047044  045503  020040  051440
6130  047052  041505  047524  000122
6131  047060  054503  047114  051104  DH6042: .ASCIZ  /CYLNDR  TRACK/
6132  047066  020040  051124  041501
6133  047074  000113
6134  047076  041505  020103  040504  DH605:  .ASCIZ  /ECC DATA IS:/
6135  047104  040524  044440  035123
6136  047112     000
6137  047113     120  051517  020040  DH6051: .ASCIZ  /POS     PAT     CORRECTABLE?/
6138  047120  020040  050040  052101
6139  047126  020040  020040  041440
6140  047134  051117  042522  052103
6141  047142  041101  042514  000077
6142  047150  047527  042122  020061  DH606:  .ASCIZ  /WORD1   WORD2   WORD3/
6143  047156  020040  047527  042122
6144  047164  020062  020040  047527
6145  047172  042122  000063
6146  047176  047507  042117  020040  DH701:  .ASCIZ  /GOOD    BAD     WORD NUM/
6147  047204  020040  040502  020104
6148  047212  020040  020040  047527
6149  047220  042122  047040  046525
6150  047226     000
6151  047227     102  042101  044040  DH607:  .ASCIZ  /BAD HEADER IS:/
6152  047234  040505  042504  020122
6153  047242  051511  000072
6154  047246  045522  051504  020040  DH204:  .ASCIZ  /RKDS    RKER    RKMR2   RKMR3/
6155  047254  020040  045522  051105
6156  047262  020040  020040  045522
6157  047270  051115  020062  020040
6158  047276  045522  051115  000063
6159  047304  051105  047522  020122  DH502:  .ASCIZ  /ERROR TRYING TO GET FAILURE INFORMATION/
6160  047312  051124  044531  043516
6161  047320  052040  020117  042507
```

```
6162  047326  020124  040506  046111
6163  047334  051125  020105  047111
6164  047342  047506  046522  052101
6165  047350  047511  000116
6166  047354  042523  047503  042116  D4503:  .ASCIZ   SECOND TIMEOUT OCCUPRED GETTING DRIVE STATUS/
6167  047362  052040  046511  047505
6168  047370  052125  047440  041503
6169  047376  051125  042522  020104
6170  047404  042507  052124  047111
6171  047412  020107  051104  053111
6172  047420  020105  052123  052101
6173  047426  051525     000
6174  047431     116  046525  042502  DH800:  .ASCIZ  /NUMBER OF RETRIES:/
6175  047436  020122  043117  051040
6176  047444  052105  044522  051505
6177  047452  000072
6178                                  .EVEN
6179  047454  042523  052103  051117  DH900:  .ASCIZ  /SECTOR  WC SB   WC IS/
6180  047462  020040  041527  051440
6181  047470  020102  020040  041527
6182  047476  044440  000123
6183  047502  001116  006340  006372  DT100:  .WORD   $ERRPC,DRINUS,ERRCOM,$REG0,$REG1,$REG2,$REG3,$REG4
6184  047510  001162  001164  001166
6185  047516  001170  001172
6186  047522  001174  001176  001200  DT201:  .WORD   $REG5,$REG6,$REG7,$REG10,$REG11,$REG12,$REG13
6187  047530  001202  001204  001206
6188  047536  001210
6189  047540  001212  001214          DT202:  .WORD   $REG14,$REG15
6190  047544  001216  001220  001222  DT203:  .WORD   $REG16,$REG17,$REG20,$REG21,$REG22,$REG23,$REG24,$REG25
6191  047552  001224  001226  001230
6192  047560  001232  001234
6193  047564  001174  001176  001200  DT601:  .WORD   $REG5,$REG6,$REG7
6194  047572  001202  001204  001206  DT602:  .WORD   $REG10,$REG11,$REG12
6195
6196  047600  000005                  DF01:   .WORD   5               ;NUMBER OF HEADER LINES
6197  047602     000                          .BYTE   0               ;NUMBER OF COL FOR FIRST HDR
6198  047603     000                          .BYTE   0               ;ALL CHARACTERS OCTAL
6199  047604  046363                          .WORD   DH101           ;SECOND HDR LINE
6200  047606     010     000                  .BYTE   10,0            ;NUM OF COL-ALL OCTAL
6201  047610  046230                          .WORD   DH200
6202  047612     000     000                  .BYTE   0,0
6203  047614  046463                          .WORD   DH201
6204  047616     007     000                  .BYTE   7,0
6205  047620  046250                          .WORD   DH500
6206  047622     000     000                  .BYTE   0,0
6207
6208  047624  000006                  DF02:   .WORD   6
6209  047626     000     000                  .BYTE   0,0
6210  047630  046363                          .WORD   DH101
6211  047632     010     000                  .BYTE   10,0
6212  047634  046230                          .WORD   DH200
6213  047636     000     000                  .BYTE   0,0
6214  047640  046463                          .WORD   DH201
6215  047642     007     000                  .BYTE   7,0
6216  047644  046551                          .WORD   DH202
6217  047646     002     000                  .BYTE   2,0
```

```
6218  047650  046566                      .WORD   DH203
6219  047652     010      000             .BYTE   10,0
6220
6221  047654  000007              DFC3:   .WORD   7
6222  047656     000      000             .BYTE   0,0
6223  047660  046363                      .WORD   DH101
6224  047662     010      000             .BYTE   10,0
6225  047664  046230                      .WORD   DH200
6226  047666     000      000             .BYTE   0,0
6227  047670  046463                      .WORD   DH201
6228  047672     007      000             .BYTE   7,0
6229  047674  046306                      .WORD   DH501          ;"THE FOLLOWING REG DATA MB INCORRECT"
6230  047676     000      000             .BYTE   0,0
6231  047700  046551                      .WORD   DH202
6232  047702     002      000             .BYTE   2,0
6233  047704  046566                      .WORD   DH203
6234  047706     010      000             .BYTE   10,0
6235
6236  047710  000006              DF05:   .WORD   6              ;OPERATION INCOMPLETE
6237  047712     000      000             .BYTE   0,0
6238  047714  046363                      .WORD   DH101
6239  047716     010      000             .BYTE   10,0
6240  047720  046751                      .WORD   DH603
6241  047722     000      000             .BYTE   0,0
6242  047724  047150                      .WORD   DH606
6243  047726     003      000             .BYTE   3,0
6244  047730  046704                      .WORD   DH602
6245  047732     000      000             .BYTE   0,0
6246  047734  047150                      .WORD   DH606
6247  047736     003      000             .BYTE   3,0
6248
6249  047740  000006              DF06:   .WORD   6              ;HEADER VRC ERROR
6250  047742     000      000             .BYTE   0,0
6251  047744  046363                      .WORD   DH101
6252  047746     010      000             .BYTE   10,0
6253  047750  046662                      .WORD   DH601
6254  047752     000      000             .BYTE   0,0
6255  047754  047150                      .WORD   DH606
6256  047756     003      000             .BYTE   3,0
6257                          ;           .WORD   DH607          ;THIS LINE WHEN SPECIFIC HEADER CAN BE READ
6258  047760  046704                      .WORD   DH602          ;THIS LINE THEN DELETES
6259  047762     000      000             .BYTE   0,0
6260  047764  047150                      .WORD   DH606
6261  047766     003      000             .BYTE   3,0
6262
6263  047770  000006              DF07:   .WORD   6
6264  047772     000      000             .BYTE   0,0
6265  047774  046363                      .WORD   DH101
6266  047776     010      000             .BYTE   10,0
6267  050000  046777                      .WORD   DH604
6268  050002     000      000             .BYTE   0,0
6269  050004  047031                      .WORD   DH6041
6270  050006     003      000             .BYTE   3,0
6271  050010  047076                      .WORD   DH605
6272  050012     000      000             .BYTE   0,0
6273  050014  047113                      .WORD   DH6051
```

```
6274  050016  003  000                .BYTE   3,0
6275
6276  050020  000004                DF10:   .WORD   4
6277  050022  000  000                        .BYTE   0,0
6278  050024  046363                        .WORD   DH101
6279  050026  010  000                        .BYTE   10,0
6280  050030  046777                        .WORD   DH604
6281  050032  000  000                        .BYTE   0,0
6282  050034  047031                        .WORD   DH6041
6283  050036  003  000                        .BYTE   3,0
6284
6285  050040  000003                DF11:   .WORD   3
6286  050042  000  000                        .BYTE   0,0
6287  050044  047031                        .WORD   DH6041
6288  050046  003  000                        .BYTE   3,0
6289  050050  047176                        .WORD   DH701
6290  050052  000  000                        .BYTE   0,0
6291
6292  050054  000005                DF12:   .WORD   5
6293  050056  000  000                        .BYTE   0,0
6294  050060  046363                        .WORD   DH101
6295  050062  010  000                        .BYTE   10,0
6296  050064  046230                        .WORD   DH200
6297  050066  000  000                        .BYTE   0,0
6298  050070  046463                        .WORD   DH201
6299  050072  007  000                        .BYTE   7,0
6300  050074  047246                        .WORD   DH204
6301  050076  004  000                        .BYTE   4,0
6302
6303  050100  000005                DF13:   .WORD   5               ;FORMAT FOR 2ND LEVEL ERROR
6304  050102  000  000                        .BYTE   0,0             ;IN HEADER COMPARE ERROR
6305  050104  046363                        .WORD   DH101           ;AND 2ND LEVEL HEADER
6306  050106  010  000                        .BYTE   10,0            ;VRC ERROR
6307  050110  046662                        .WORD   DH601
6308  050112  000  000                        .BYTE   0,0
6309  050114  047150                        .WORD   DH606
6310  050116  003  000                        .BYTE   3,0
6311  050120  047304                        .WORD   DH502
6312  050122  000  000                        .BYTE   0,0
6313
6314  050124  000005                DF14:   .WORD   5               ;FORMAT FOR 2ND LEVEL ERROR
6315  050126  000  000                        .BYTE   0,0             ;IN OPERATION INCOMPLETE ERROR
6316  050130  046363                        .WORD   DH101
6317  050132  010  000                        .BYTE   10,0
6318  050134  046751                        .WORD   DH603
6319  050136  000  000                        .BYTE   0,0
6320  050140  047150                        .WORD   DH606
6321  050142  003  000                        .BYTE   3,0
6322  050144  047304                        .WORD   DH502
6323  050146  000  000                        .BYTE   0,0
6324
6325  050150  000010                DF15:   .WORD   10
6326  050152  000  000                        .BYTE   0,0
6327  050154  046363                        .WORD   DH101
6328  050156  010  000                        .BYTE   10,0
6329  050160  046230                        .WORD   DH200
```

```
6330  050162  000     000                 .BYTE   0,0
6331  050164  046463                      .WORD   DH201
6332  050166  007     000                 .BYTE   7,0
6333  050170  046551                      .WORD   DH202
6334  050172  002     000                 .BYTE   2,0
6335  050174  047354                      .WORD   DH503
6336  050176  000     000                 .BYTE   0,0
6337  050200  046306                      .WORD   DH501
6338  050202  000     000                 .BYTE   0,0
6339  050204  046566                      .WORD   DH203
6340  050206  010     000                 .BYTE   10,0
6341
6342  050210  000010              DF16:   .WORD   10
6343  050212  000     000                 .BYTE   0,0
6344  050214  046363                      .WORD   DH101
6345  050216  010     000                 .BYTE   10,0
6346  050220  046230                      .WORD   DH200
6347  050222  000     000                 .BYTE   0,0
6348  050224  046463                      .WORD   DH201
6349  050226  007     000                 .BYTE   7,0
6350  050230  046551                      .WORD   DH202
6351  050232  002     000                 .BYTE   2,0
6352  050234  047304                      .WORD   DH502
6353  050236  000     000                 .BYTE   0,0
6354  050240  046306                      .WORD   DH501
6355  050242  000     000                 .BYTE   0,0
6356  050244  046566                      .WORD   DH203
6357  050246  010     000                 .BYTE   10,0
6358
6359  050250  000004              DF17:   .WORD   4
6360  050252  000     000                 .BYTE   0,0
6361  050254  047150                      .WORD   DH606
6362  050256  003     000                 .BYTE   3,0
6363  050260  047227                      .WORD   DH607
6364  050262  000     000                 .BYTE   0,0
6365  050264  047150                      .WORD   DH606
6366  050266  003     000                 .BYTE   3,0
6367
6368  050270  000002              DF21:   .WORD   2
6369  050272  000     000                 .BYTE   0,0
6370  050274  047113                      .WORD   DH6051
6371  050276  003     000                 .BYTE   3,0
6372
6373  050300  000001              DF23:   .WORD   1
6374  050302  001     000                 .BYTE   1,0
6375
6376  050304  000001              DF24:   .WORD   1
6377  050306  002     000                 .BYTE   2,0
6378
6379  050310  000001              DF25:   .WORD   1
6380  050312  003     000                 .BYTE   3,0
6381
6382  050314  000002              DF26:   .WORD   2
6383  050316  000     000                 .BYTE   0,0
6384  050320  047031                      .WORD   DH6041
6385  050322  003     000                 .BYTE   3,0
```

```
  6386          000001                  .END
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
ABORT = 000200          1201# 3385  3390  3465  3490  3575  3585  3590  3595  3599  3607  3613  3626
                        3656
ACLO  = 000010           791# 3427
ALLTRM  027764          3466  3597  3604#
ALNPAC  014763          1804# 2349
ANYDER= 100000          1206# 2322  2621  2645  2669  2886  2952  2967  3049  3075  3164  3646  3896
A.ABNL  002600          1072# 2271* 3361* 3605* 3619* 3621* 3726  3727* 3751* 3813* 4694
A.CONT  002602          1073# 4704
A.NORM  002576          1071# 2272* 3362* 3606* 3814* 4698
BADENT  013711          1703# 2127  2139
BADSEC= 001000          1203# 2550  2558  2677  3552  3649
BAD632  014427          1765# 2327
BAI   = 000020           750# 4863
BA16  = 000400           726#
BA17  = 001000           727#
BDSRCK  021476          2540# 2676  3551
BDSTWT  031452          2406  3879#
BGNRTY  030144          3616  3625  3644#
BIT0  = 000001           123#  723   740   766   788   936  1080  1153  2636  2876  3002  3265  4978
BIT00 = 000001           113#  123
BIT01 = 000002           112#  122
BIT02 = 000004           111#  121
BIT03 = 000010           110#  120
BIT04 = 000020           109#  119
BIT05 = 000040           108#  118
BIT06 = 000100           107#  117
BIT07 = 000200           106#  116
BIT08 = 000400           105#  115
BIT09 = 001000           104#  114  5288
BIT1  = 000002           122#  741   768   937  1081  1154  1195  3269
BIT10 = 002000           103#  728   757   779   814   828   841   856   870   949  1090  1204  3295
                        5272
BIT11 = 004000           102#  729   758   780   799   815   829   842   857   871   950  1205
BIT12 = 010000           101#  731   759   781   816   830   843   844   858   872   951
BIT13 = 020000           100#  732   760   782   800   817   831   845   859   873   952  5279
BIT14 = 040000            99#  733   761   783   801   818   832   846   860   874   953  1091  2485
BIT15 = 100000            98#  734   735   762   784   802   819   879   954  1092  1206  2467  3298
BIT2  = 000004           121#  742   769   770   790   938  1082  1155  1196  3273
BIT3  = 000010           120#  748   749   771   791   939  1083  1156  1197  3277
BIT4  = 000020           119#  743   750   772   792   793   808   850   864   940  1084  1157  1198
                        3281
BIT5  = 000040           118#  751   752   773   794   809   823   836   851   865   941  1085  1158
                        1199  3285  4775
BIT6  = 000100           117#  724   753   774   795   810   824   837   852   866   942  1087  1200
BIT7  = 000200           116#  725   754   775   796   797   811   825   838   853   867   944  1201
BIT8  = 000400           115#  726   755   776   777   798   812   826   839   854   868   945  1088
                        1202  3289
BIT9  = 001000           114#  727   756   778   813   827   840   855   869   946  1089  1203  2453
                        3292
BLDEXH  031324          3506  3531  3851#
BLDHDR  021120          2441# 2592  2686  2747  2983
BPTVEC= 000014           130#
BSE   = 000200           775# 3546  4128  4294  4530
BSERR = 000002          1195# 2648  2659  2894  2903  3056  3548  3555  3644
BSFACT  004660          1163# 2310  2341  2347  2378  2379  2383  2468  2541  3930
BSHEAD  015547          1871# 3920
```

```
BSREPT  031652          2409    3917#
BSSINS  021602          2566#   2685    2982
BSSOFT  003660          1162#   2085    2336    2355    2377    2483    2486    2546    2548    2567    3889    3951
BSTAIL  015617          1878#   3928
BSWERR  015402          1854#   3905
BSWTFI  015512          1866#   3911
BUFFO   005674          1165#   2444    2479*   2480*   2612    2845    2847    2849    2856    2857    2858
BUFFI   006100          1166#   2822*   2823*   2824*   2845    2847    2849    2859    2860    2861
BUFPRT  006304          1167#
B.BA16= 000001           740#   4857
B.BA17= 000002           741#   4857
B.CDT = 000004           742#   2008    4511    4857
B.CFMT= 000020           743#   2046    2316    2317    2451    2615    3769    3771    3773    3780    3885    3888    3923
                        4511    4857    4941
CCINUS  006370          1193#   2587*   2741*   3811
CCLR  = 100000           735#   3718    4464
CDT   = 002000           728#   4270    4820    4877    4928    4951
CERR  = 100000           734#   3515    3540    3729    3741    4166    4225    4284    4466    4517    4832    4888    4976
CERTY   030174          3306    3651#
CFMT  = 010000           731#   3833    3863    4270    4820    4877    4928    4951
CKPRES  017526          2168    2175#
CKSWR = 104406          5268    5287    5708#
CLEAR = 000105           694#
CLR   = 000040           751#
CMDTO = 000100           942#   3408    3737    4027
COE   = 001000           778#   3472    4128    4294    4530
COLHD   015766          1898#   3931
COMSTR  005660          1164#   3140    3664
CONCLR= 000176           712#   4959
CONERR  025616          1073    3257#
CONREQ  014373          1760#   2295    2302
CORABL  015233          1834#   2974
CR    = 000015            38#   5127    5137
CRLF  = 000200            39#   5098    5137
CTINUS  006366          1192#   2588*   2742*   3812
CTO   = 004000           729#   3435    4526
CYLLIM  013254          1653#   1990
C.INIT  035532          3150    4737#
C.RTRN  037404          4841    4869    4897    4904    4907    4914    4918    4935    4957    4962    4972    4981    4994
                        4998    5003#
C.SPEC  036424          4777    4873#
DACOMP  024350          2971    2998#
DCK   = 100000           784#   3557    4128    4294    4530
DCKERR= 000020          1198#   2896    2969    2976    3058    3082    3559    3644
DCLO  = 000020           793#
DDISP = 177570            45#    179    1935
DDT   = 000400           798#
DERCNT  002650          1146#   2632*   2661*   2671    2674    2690    2726*   2872*   2909*   2943*   2954    2979    2990
                        3040*   3065*   3077    3080    3112*   3115    3499    3524    3566    3577    3881*   3902*   3903
DESL  = 000010           748#
DF01    047600           247     253     259     265     307     415    6196#
DF02    047624           271     289     295     301     313     319     325     331     337     343     349     355     361
                         367     373     379     433     439     445     553     559     565     571     577     583     589
                         595     601     607     649    6208#
DF03    047654           277     283    6221#
DF05    047710           385    6236#
```

```
DF06    047740          391   6249#
DF07    047770          397   6263#
DF10    050020          403   6276#
DF11    050040          409   6285#
DF12    050054          451    457    463    469    475    481    487    493   6292#
DF13    050100          499    511   6303#
DF14    050124          505   6314#
DF15    050150          517   6325#
DF16    050210          523   6342#
DF17    050250          541   6359#
DF21    050270          619   6368#
DF23    050300          625    631    637   6373#
DF24    050304          643   6376#
DF25    050310          547    655   6379#
DF26    050314          667   6382#
DH100   046203          245    251    257    263    269    275    281    287    293    299    305    311    317
                        323    329    335    341    347    353    359    365    371    377    383    389    395
                        401    413    431    437    443    449    455    461    467    473    479    485    491
                        497    503    509    515    521    551    557    563    569    575    581    587    593
                        599    605    647   6054#
DH101   046363         6074#  6199   6210   6223   6238   6251   6265   6278   6294   6305   6316   6327   6344
DH200   046230         6058#  6201   6212   6225   6296   6329   6346
DH201   046463         6085#  6203   6214   6227   6298   6331   6348
DH202   046551         6095#  6216   6231   6333   6350
DH203   046566         6098#  6218   6233   6339   6356
DH204   047246         6154#  6300
DH500   046250         6061#  6205
DH501   046306         6066#  6229   6337   6354
DH502   047304         6159#  6311   6322   6352
DH503   047354         6166#  6335
DH601   046662          539   6108#  6253   6307
DH602   046704         6111#  6244   6258
DH603   046751         6118#  6240   6318
DH604   046777          407    665   6122#  6267   6280
DH6041  047031         6127#  6269   6282   6287   6384
DH6042  047060          641   6131#
DH605   047076          617   6134#  6271
DH6051  047113         6137#  6273   6370
DH606   047150         6142#  6242   6246   6255   6260   6309   6320   6361   6365
DH607   047227         6151#  6363
DH701   047176         6146#  6289
DH800   047431          623    629    635   6174#
DH900   047454          653   6179#
DI    = 040000          733#  4314   4986
DISPLA  001142          179#  1935*  1943*  5271*
DISPRE  000174          145#  1943
DLT   = 100000          762#  3492   4122   4252   4288   4528
DMD   = 000040          809#
DOL1    030230         3652   3658#
DOL2    030304         3647   3650   3673#
DOL3    030164         3645   3648#
DONE    002652         1148#  3137*  3153   3163*  3258*  3355*  3675*  3678*
DONEPR  016462         1994#  2237
DRA   = 000001          788#  4117   4900
DRDY  = 000200          797#  3611
DRINIT  017776          150   2099   2169   2190   2268#  2270   2298   2305   2413
```

```
DRINUS  006340          1181#   2015    2276    2283    3728    6183
DROT   = 000040           794#  3482
DRPAR  = 000010           771#  3422
DRPDRV= 002000            949#  4762
DRVCAL  025220           2282    2285    2287    2289    2309    2321    2620    2644    2668    2780    2885    2951    2966
                         3048    3074    3137#   3620    3672    3779    3797    3802    3895
DRVDSC= 000040            941#  3587    4202    4368    4430
DRVHRD= 000020            940#  3582    4209    4404    4535
DRVMSK= 000007            747#  4111    4152    4303    4748
DRVPDT= 000004            938#  4310    4378
DRVPOS= 000002            937#  4310    4337    4799
DRVQ    013122           1635#  1978
DRVSZD= 010000            951#  4140    4902
DRVUSE= 000001            936#  4762
DRY    = 000200           796#
DR.CLR= 000005            719#  4190    4353    4395    4419    4533
DR.SEL= 000001            718#  4260    4339    4654    4661    4668    4675    4879
DSC    = 040000           801#
DSPTCH  020720           2391    2395#
DSWR   = 177570            44#   178    1934
DTBAII= 100000            954#  2643    2667    2883    2950    3046    3073    4762    4861
DTE    = 010000           781#  3487    4530
DTYE   = 000040           773#  3450    4530
DT100   047502            246    252     258     264     270     276     282     288     294     300     306     312     318
                          324    330     336     342     348     354     360     366     372     378     384     390     396
                          402    414     432     438     444     450     456     462     468     474     480     486     492
                          498    504     510     516     522     552     558     564     570     576     582     588     594
                          600    606     648    6183#
DT201   047522           6186#
DT202   047540           6189#
DT203   047544           6190#
DT601   047564            408     540     618     624     630     636     642     666    6193#
DT602   047572            546     654    6194#
ECCNC  = 000040           1199#  2972    3087    3562
ECCW   = 020000            817#
ECH    = 000100            774#  3560    4128    4294    4530
ECINUS  006360           1189#  1999    2003    2114    2244    2433
EMTVEC= 000030             133#  1922*   1923*
EM1     042462             244   5714#
EM10    042753             286   5747#
EM100   045527             634   6001#
EM101   045556             640   6005#
EM102   045640             646   6014#
EM103   045716             652   6022#
EM104   045762             658   6028#
EM105   046047             664   6037#
EM11    042766             292   5749#
EM12    043005             298   5752#
EM13    043040             304   5757#
EM14    043066             310   5761#
EM15    043126             316   5767#
EM16    043153             322   5771#
EM17    043174             328   5774#
EM2     042512             250   5718#
EM20    043221             334   5778#
EM21    043250             340   5782#
```

```
EM22    043302            346    5787#
EM23    043336            352    5792#
EM24    043401            358    5798#
EM25    043425            364    5802#
EM26    043454            370    5806#
EM27    043500            376    5810#
EM3     042550            256    5723#
EM30    043535            382     502    5815#
EM31    043574            388     508    5821#
EM32    043621            394    5825#
EM33    043646            400    5829#
EM34    043674            406    5833#
EM35    043720            412    5837#
EM36    043760            418    5843#
EM37    044017            424    5849#
EM4     042605            262    5728#
EM40    044066            430    5856#
EM41    044142            436    5864#
EM42    044210            442    5871#
EM43    044250            448    5877#
EM5     042632            268    5732#
EM52    044317            490    5884#
EM53    044351            496    5889#
EM56    044402            514     520    5894#
EM6     042660            274    5736#
EM60    044430            526    5898#
EM61    044476            532    5905#
EM62    044550            538    5912#
EM63    044576            454     550    5916#
EM64    044647            460     556    5923#
EM65    044726            466     562    5931#
EM66    044760            472     568    5936#
EM67    045017            478     574    5942#
EM7     042713            280    5741#
EM70    045066            484     580    5949#
EM71    045150            586    5958#
EM72    045214            592    5964#
EM73    045267            598    5972#
EM74    045327            604    5978#
EM75    045353            610    5982#
EM76    045407            616    5987#
EM77    045502            622     628    5997#
ERRCNT  002654   1150#    2273#  3166#  3168   3171   3173#  3259#  3356#  3651   3654   3809#
ERRCOM  006372   1194#    3314#  3315#  3689#  3690#  6183
ERRFRE  025302   1071     2272   3163#  3606   3814
ERRHDL  026240   1072     2271   3354#  3605   3621   3813
ERRLMT  002655   1151#    3651
ERRVEC= 000004    126#    1932   1933#  1944#  5605   5606   5610#  5611#  5618#  5619#
ER2ENT  026260   3358#    3520   3545   3628
ETINUS  006354   1187#    2001   2009   2012   2146   2242   2429
E.CCLR= 000001   1080#    4469
E.CERR= 001000   1089#    4543
E.CLAT= 000020   1084#    4195   4359   4400   4424
E.CMTO= 040000   1091#
E.CONT  002604   1074#    3265   3269   3273   3277   3281   3285   3289   3292   3295   3298   3305#  4094#
                 4132#    4195#  4205#  4276#  4298#  4321#  4327#  4359#  4400#  4424#  4469#  4521#  4543#
```

```
                    4682*   4916*   4970*   4978*   4991*
E.DLT = 000400       1088*   4276
E.DPAR= 002000       1090*   4205    4682
E.ILLD= 000100       1087*   4970
E.MDS = 100000       1092*   4094    4521    4916
E.NOAT= 000002       1081*   4321
E.SCLR= 000040       1085*   4991
E.UATT= 000004       1082*   4327
E.UDAT= 000010       1083*   4132    4298
E.UNLD= 020000        952*   4213    4333    4408    4418
FBSLAB  015645       1882*   3929
FMTE  = 000020        772*   3454    4530
GETANS  017610       1977    1979    1981    1983    1985    1987    1989    1991    1993    2206*
GETDRS  030466       3710    3716*
GETOUT  016734       2021    2042*
GETUBS  017136       2073    2082*
GNS   = ****** U      144    5701    5702    5703    5704    5706    5708    5709    5710    5711    5712
GO    = 000001        723*
GTPKAD  031200       3568    3579    3823*
GTSWR = 104405       5706*
HCRC  = 000400        776*
HDRCMP  023354       2748    2841*
HDR.AD  002622       1114*   4246    4250*   4940*
HDR.CT  002624       1115*   4254*   4943*   4946*
HELPFL  006377       1209*   1960
HELPQ   013036       1626*   1955
HLTPRG  030136       3634*   3657
HT    = 000011         36*   5096    5137
HVRC  = 000400        777*   3521    4128    4294    4530
HVRCER= 000004       1196*   2648    2894    3523    3644
H.HEAD  002627       1118*
IDAE  = 002000        779*   3477
IE    = 000100        724*   2296    2303    3511    3536    4474    4538    4826    4867    4933    5000
ILC   = 000001        766*   3438    4530
ILF   = 000004        769*
ILLSEC  013731       1706*   2155
IMPERR  015160       1826*   2728
IMPER2  015066       1816*   2987
INITDR  006312       1170*
INITEC  006332       1178*
INITET  006326       1176*
INITMD  006316       1172*   2035
INITOF  006334       1179*   2039    2051
INITSC  006330       1177*
INITSE  00631'       1171*
INITST  006324       1175*   2027
INITTP  006310       1169*   1974
INITWE  006320       1173*
INITWO  006322       1174*
INTMSK  002633       1123*   4169    4193    4214    4239    4325    4357    4398    4409    4422    4693*   4697*   4701*
                     4750*   4836
INTR  = 000300        714*
INVDCY  014025       1717*   2024    2118
INVDDR  014102       1726*   2030
INVDMD  014150       1733*   2034
INVDOF  014211       1739*   2038
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
INVDSE  013756          1710*   2022
INVDTK  014055          1722*   2026    2150
INVDTP  014126          1730*   2032
IOTVEC= 000020           131*
IR    = 000100           753*
IVDCYL  016640          2004    2024*
IVDDRV  016664          2016    2030*
IVDMOD  016700          2018    2034*
IVDOF   016716          2020    2038*
IVDTP   016672          1998    2032*
IVDTRK  016646          2010    2013    2026*
I.AERR  034114          4345    4351    4369    4378*
I.ATTN  033444          4105    4149    4282*
I.CCLR  034500          4183    4233    4464*   4536    4896    4961
I.CSTS  035132          4176    4611*
I.CST1  035154          4180    4232    4615*   4885
I.DRV   002634          1127*   4750    4751
I.ERR   032666          4177    4181*
I.ERRA  032646          4176*   4244
I.ERRC  032654          4114    4167    4179*   4312
I.HDAL  033112          4160    4223*
I.INTR  032232          1948    4084*
I.ISRL  002626          1117*   4098    4101*   4104*   4475*   4772*   4926*   4999*
I.ISSU  034562          4191    4261    4340    4354    4396    4420    4506*   4655    4662    4669    4676
I.IOO   032512          4102    4119    4146*
I.RTRN  034462          4096    4134    4144    4172    4184    4189    4192    4198    4207    4216    4219    4234    4236
                        4262    4265    4274    4278    4300    4316    4323    4329    4341    4355    4362    4374    4394
                        4397    4402    4411    4414    4421    4426    4435*
I.STAT  035236          4188    4393    4652*   4906
I.STOR  035004          4516    4566*
I.UNLD  034362          4335    4418*
KIPARD= ****** U        5601
LEV2ER= 000400          1202*   3302    3358    3411    3502    3508    3527    3533    3622    3676    3749
LF    = 000012            37*   5131    5137
MCLK  = 000400           812*
MDINUS  006344          1183*   2017    2036    2054    2058    2062    2067    2290
MDS   = 001000           756*   4092    4519    4894
MERD  = 001000           813*
MESMSK= 000017           806*
MEWD  = 002000           814*
MIND  = 000200           811*
MODEQ   013153          1640*   1982
MSP   = 000100           810*
M.CADD= 017760           882*
M.CDIF= 017760           881*
M.DRV = 000007           878*
M.HEAD= 007000           885*
  .ID = 000003           880*
m.PAR = 100000           879*
M.SECT= 000760           884*
M.SER = 077770           883*
NED   = 010000           759*   3392    3402    4107    4185    4528    4898
NEM   = 004000           758*   3387    4122    4288    4528
NEXTQ   013644          1696*   2106    2160
NOCHK = 000400           945*   4762    4824    4865    4892    4910    4931    4954    4963    4997
NODSC = 004000           950*   3592    4350
```

```
NONE    015751            1894*  3955   3965
NPRSVD  013452            1675*  2189
NXF   = 000004             770*  3446
OCTBIN  037426            2112   2131   2143   2216   5028*
OFFSET= 000115             698*  3798   4807
OFINUS  006362            1190*  2019   2040   2051   2235   3799
OFSETQ  013275            1656*  1992
OFST  = 000004             790*
OPCOMP  002651            1147*  2426*  2583*  2602   2740*  2763
OPCONT  002656            1152*  2042*  2053*  2056*  2060*  2064*  2069*  2072*  2082   2186*  2330   2332   2353
                          2357   2396   2401   2404   2749   2753   3793
OPI   = 020000             782*  3496   4128   4294   4530
OPIERR= 000010            1197*  2648   2894   3498   3644
OR    = 000200             754*
OREQSW= 000020            1157*  2053   3793
O.WAIT  002606            1094*  4028   4032*  4142*  4146   4162*  4228*  4241*  4256*  4800*  4860*  4922*  4939*
PACK  = 000103             693*  2286
PARAM3  016402            1982*  2037
PARAM5  016432            1988*  2029
PARM0   002362             983*  1138   2279   3604   3624   3671   3782
PARM1   002446            1015*  2275   3360   3617   3766
PARM6   016452            1992*  2041
PAT   = 000020             808*
PBLKT   002644            1138*  4025   4112   4138   4306   4331   4749*
PBSVAL= 001000             946*  3260   4687
PCA   = 004000             815*
PCD   = 010000             816*
PGE   = 002000             757*  3442   4528
PIP   = 020000             800*
PIRQ  = 177772              43*
PIRQVE= 000240             137*
POSOFF  031040            2773   2874   2944   2958   3042   3066   3792*
PREPLP  031126            2599   2760   3809*
PRGRS1  015255            1838*  2399
PRGRS2  015322            1845*  2403
PROGID  012747            1616*  1954
PRSVD   013436            1673*  2187
PR0   = 000000              60*
PR1   = 000040              61*
PR2   = 000100              62*
PR3   = 000140              63*
PR4   = 000200              64*
PR5   = 000240              65*  1070
PR6   = 000300              66*
PR7   = 000340              67*  1949
PS    = 177776              40*    41   4014   4020*  4034*  4743   4744*  5003*
PSDBSF= 000040            1158*  2186   2332   2357
PSDQ    013305            1658*  2175
PSW   = 177776              41*
PWRVEC= 000024             132*  1926*  1927*  5556*  5557*  5566*  5572*  5584*  5585*
P.ASOF= 000032             967*  3331   3800*  3801*  4356*  4357   4390*  4621*  4809*  4810
P.A00 = 000040             970*  2292   2299   3334   4347*  4348   4364*  4365   4678*  4886*
P.A01 = 000044             972*  3336   4657*
P.A10 = 000050             974*  3338   4664*
P.A11 = 000054             976*  3340   4671*
P.BAHI= 000007             929*
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
P.BALO= 000010   330#  2310#  2336#  2612#  2635#  2638#  2774#  2875#  2878#  2965#  3072#  3323  3697
                3889#  4845   4940
P.BAR = 000024   964#  3330   4387#  4617#
P.B00 = 000042   971#  3335   4342#  4343   4679#  4887#
P.B01 = 000046   973#  3337   4658#
P.B10 = 000052   975#  3339   3760   4665#
P.B11 = 000056   977#  3341   4263#  4672#
P.CMND= 000001   923#  2281#  2284#  2286#  2288#  2306#  2313#  2611#  2641#  2662#  2779#  2884#  2945#
                2959#  3047#  3067#  3315   3618#  3690   3774#  3795#  3798#  3890#  4158   4775   4779
                4791   4801   4807   4813   4816   4822   4847   4854   4873   4920   4937   4959   4968
P.CS1 = 000016   961#  3324   3417   3435   3833   3863   4179#  4231#  4269#  4270#  4272#  4273   4382#
                4384#  4611#  4614#  4765   4819#  4820#  4822#  4826#  4827   4851#  4854#  4856#  4857#
                4867#  4868   4876#  4877#  4879#  4880   4882#  4883   4888   4927#  4928#  4930#  4933#
                4934   4950#  4951#  4953#  4956   4975#  4976   4985#  4986
P.CS1H= 000007   928#  2277#  2316#  2352#  3771#  3773#  3780#  3781#  3888#  4269   4510   4819   4956
                4876   4927   4941   4950
P.CS2 = 000020   962#  3325   3382   3387   3392   3394   3400   3402   3442   3492   3573   4185   4385#
                4615#  4773#  4798   4863#  4864   4875   4894   4898   4924#  4925   4949
P.CYLN= 000002   924#  2307#  2613#  2639#  2664#  2776#  2880#  2946#  2961#  3043#  3068#  3317   3691
                3768#  3796#  3891#  4267   4803   4845   4947
P.DCYL= 000030   966#  3326   3823   3852   4389#  4623#
P.DRVN= 000000   922#  2276#  2283#  4155   4332   4508   4747
P.DS  = 000036   969#  3332   3427   3431   3482   4392#  4619#  4900
P.DTS = 000026   965#  2651   2657   2890   3053   3327   3549   3759   3826   3827   3853   3859   4388#
                4618#
P.EPAT= 000062   979#  3085   3570   4181#  4767
P.EPOS= 000060   978#  3084   3569   4182#
P.ER  = 000034   968#  3333   3381   4391#  4620#
P.OFST= 000006   927#  3799#  4809
P.PRST= 000014   932#  2280#  2610#  2642#  2643#  2667#  2775#  2882#  2883#  2950#  2964#  3045#  3046#
                3073#  3260   3408   3582   3587   3592   3596   3737   3884#  4027#  4140#  4174#  4202#
                4209   4213#  4310#  4333   4337#  4350#  4368#  4378#  4404   4408#  4418#  4430#  4535#
                4687#  4762#  4799#  4823#  4824   4849   4855#  4861   4865   4892   4902#  4910   4931
                4954   4963   4997
P.SECT= 000004   925#  2320#  2634#  2666#  2695#  2713   2873#  2910#  2929   2948#  2963#  3039#  3070#
                3094#  3102   3320   3694   3767#  3893#  4268   4804   4846   4948
P.TRCK= 000005   926#  2308#  2619#  2640#  2665#  2777#  2881#  2947#  2962#  3044#  3069#  3318   3692
                3892#
P.WC  = 000012   931#  2314#  2315#  2614#  2617#  2618#  2633#  2663#  2725#  2871#  2941#  2949#  2960#
                3041#  3071#  3111#  3322   3696   3894#  4844
P.WCR = 000022   963#  2654   2889   3052   3328   4386#  4616#
QNEWSW= ****** U  682   3800
QUES    006374  1208#  2162   2239
Q.INIT= 040000   953#
RCDASW= 000010  1156#  2056   2401   2749
RCLREQ= 004000  1205#  3420   3425   3470   3475   3480   3485   3615
RDALHD= 000164   711#  2779   4158   4937
RDBUF   002660  1161#  2089   2104   2367   2774   2787   2789   2798   2799   2800   2801   2803   2812
                2822   2823   2824   2965   3005
RDCHR = 104407  5497   5709#
RDDATA= 000121   700#  2313   2959   3047   3067
RDGATE= 100000   819#
RDHDRS  023016  2746   2771#
RDHDD   030700  3507   3532   3757#
RDHEAD= 000125   702#  3774   4272   4953
RDLIN = 104410  1956   2095   2107   2176   2207   5710#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
RDOP    024516          2751    3037#
RDSTAT= 000141           710#   2288    4873
RDY   = 000200           725#   3734    4830    4883
RECAL = 000113           697#   2284    3618    4816
RECODE  006264          1191#   2274#   2322    2550#   2558#   2621    2645    2648    2659    2669    2677    2886    2894
                        2896    2903    2952    2956    2967    2969    2972    2976    3049    3056    3058    3075    3082
                        3087    3164    3174#   3302    3357#   3358    3385#   3390#   3411    3420#   3425#   3465#   3470#
                        3475#   3480#   3485#   3490#   3498#   3502    3508    3523#   3527    3533    3548#   3552    3555#
                        3559#   3562#   3575#   3576#   3585#   3590#   3595#   3599#   3607    3613#   3615    3622    3626#
                        3644    3646#   3649    3656#   3676#   3749#   3810#   3896
RELEAS= 000140           709#   4920
REPSUP  026060          3262    3312#   3363
RESREG= 104412          2466    2551    2578    2625    2731    2836    2864    2993    3017    3118    3244    3342    3673
                        3711    3752    3783    3803    3871    3912    3966    5163    5712#
RESVEC= 000010           127#
RETANL  030310          3361    3619    3675#   3727
RETNML  030326          3362    3678#
RKASOF= 000016           678#   3704    4168    4238    4318    4573    4621    4810#   4835
RKBA  = 000004           673#   3703    4569    4617    4843#
RKBAS   002570          1068#   2278    3510    3535    3610    3658    3687    4021    4090    4754
RKCS1 = 000000           671#   2296#   2303#   3511#   3515    3536#   3540    3698    3718#   3729    3732#   3733    3740
                        4113    4165    4223    4273#   4282    4464#   4465    4474#   4513#   4514    4538#   4566    4827#
                        4829    4868#   4880#   4882    4912#   4934#   4956#   4965#   4975    4985    5000#
RKCS2 = 000010           675#   3659#   3699    3728#   3750#   4091    4251    4509#   4567    4615    4798#   4864#   4875#
                        4925#   4949#   4974#
RKDA  = 000006           674#   3701    4268#   4570    4618    4804#   4846#   4948#
RKDB  = 000024           681#   3512    3513    3514    3537    3538    3539    4247    4248    4249
RKDC  = 000020           679#   3700
RKDCYL= 000020           680#   4267#   4574    4623    4803#   4845#   4947#
RKDS  = 000012           676#   3611    3705    4116    4571    4619
RKECPS= 000030           686#   4182    4578
RKECPT= 000032           688#   4181    4579
RKER  = 000014           677#   3706    4125    4291    4572    4620
RKMR1 = 000022           682#   3731#   4259#   4338#   4575    4652#   4659#   4666#   4673#   4774#
RKMR2 = 000034           683#   3743    4576    4886
RKMR3 = 000036           684#   3744    4577    4887
RKPAT = 000032           687#
RKPOS = 000030           685#
RKPRI   002574          1070#   4020    4744
RKVEC   002572          1069#   1947
RKWC  = 000002           672#   3702    4568    4616    4844#
RLS   = 000010           749#   4924
RSTADD  016632          2000    2002    2022#
RWLREQ  014313          1751#   2301
R.ABNL  035464          4033    4143    4218    4235    4413    4433    4693#   4903
R.CONT  035510          4095    4133    4197    4206    4277    4299    4322    4328    4361    4401    4425    4470    4522
                        4545    4683    4701#   4917    4971    4980    4993
R.NORM  035476          4171    4264    4373    4697#   4840    4913    5001
SAVREG= 104411          2442    2540    2566    2609    2631    2772    2842    2870    2999    3038    3186    3257    3313
                        3354    3686    3716    3758    3792    3851    3880    3918    5148    5711#
SBSLAB  015706          1888#   3956
SCINUS  006356          1188#   1999    2116    2221    2585    2738
SCLR  = 000040           752#   3659    3750    4974
SDBSFP  013372          1667#   2178
SECINE  015470          1863#   3906
SECTO   013135          1637#   1980
```

D11


```
SEEK  = 000117          699#  2306  3795  4791  4801
SEINUS  006342         1182#  2043  2045*  2153  2693  2708  2783  2814  2816  2844  2901  2906  2924
                       3062  3092
SELDRV= 000101          692#  4930
SERNUM  014723         1798#  2340
SETRS   016652         2023  2025  2027#
SKI   = 000002          768#  3467
SPACE2  012744         1614#  3222  3231  3242  3936  3940
SPACE3  016030         1904#
SPACE4  016023         1903#  3932
SPAR  = 020000          732#  3417  4526
SPCLSS= 000020          792#  3431
SRHTBS  021334         2469  2501#  2542
SRTSPL= 000111          696#  4813
STACK = 001100           31#  1920  2269  2410  3815  3818
STARS   015370         1852#  2346
START   016100          148  1913#  2351  2365  2376  2390  2683  2730  2811  3635  5338  5421
STINUS  006352         1186#  2001  2028  2148  2219  2431  2586  2739
STKLMT= 177774           42#
STPARM  016342         1959  1974#  2031  2033  2232  2414
SUBCLR= 000177          713#  2281  4966
SVAL  = 100000          802#
SWLREQ  014235         1743#  2294
SWR     001140          178#  1918  1934*  1936  1942*  2407  2411  2420  2589  2595  2743  2756  3028
                       3187  5272  5279  5284  5288  5341  5380  5435*  5564  5577*
SWREG   000176          146#  1942  5341  5380  5403
SW0   = 000001           95#  2411
SW00  = 000001           85#    95
SW01  = 000002           84#    94
SW02  = 000004           83#    93
SW03  = 000010           82#    92
SW04  = 000020           81#    91
SW05  = 000040           80#    90
SW06  = 000100           79#    89
SW07  = 000200           78#    88
SW08  = 000400           77#    87
SW09  = 001000           76#    86
SW1   = 000002           94#  3028
SW10  = 002000           75#
SW11  = 004000           74#
SW12  = 010000           73#
SW13  = 020000           72#  3187
SW14  = 040000           71#  2420
SW15  = 100000           70#
SW2   = 000004           93#
SW3   = 000010           92#
SW4   = 000020           91#
SW5   = 000040           90#
SW6   = 000100           89#
SW7   = 000200           88#  2407
SW8   = 000400           87#
SW9   = 001000           86#  2589  2595  2743  2756
S.ACLO= 000100          837#
S.BRHM= 000100          852#
S.BRKE= 040000          874#
S.CART= 000400          854#
```