```
                    00001   ;*****************************************************************
                    00002   *
                    00003   *       PREP67 - H/Z-67 Disk Preparation Utility Program
                    00004   *
                    00005   *       This software was originally developed by Heath/Zenith
                    00006   *       Data Systems.  Its purpose is to initialize the "Winchester"
                    00007   *       disk, perform a media check on the disk surface, and
                    00008   *       initialize the disk tables for subsequent use with the
                    00009   *       partitioning utility PART. PREP67 is normally run only once on
                    00010   *       a new disk drive and of course it destroys all existing
                    00011   *       content on the drive.
                    00012   *
                    00013   *       The source code for this utility was never published.  This
                    00014   *       listing is the result of a disassembly of the code
                    00015   *       by Glenn Roberts in November 2011.  All comments and label
                    00016   *       naming conventions are based on a "best guess" given the
                    00017   *       description and observation of how the PREP67 utility functions.
                    00018   *
                    00019   *       The code has been verified to assemble to produce an executable
                    00020   *       that is operationally identical to the original Version
                    00021   *       of the file from Heath/Zenith.
                    00022   *
                    00023   *       28 November 2011
                    00024   *
                    00025   *       The following mod's have been made to the baseline code:
                    00026   *
                    00027   *       1. User can specify port on command line,e.g.:
                    00028   *               PREP67 170
                    00029   *       2. set for 6 tracks/cyl. (from default 4)
                    00030   *       3. fixed CTL-C handler bug
                    00031   *       4. ORG to USERFWA
                    00032   *
                    00033   ;*****************************************************************
                    00034   *
                    00035   *       Physical disk characteristics
                    00036   *
001.000             00037   SECSIZ  EQU     256             Bytes per sector
000.050             00038   SPTRK   EQU     40              Sectors per track
000.006             00039   TPCYL   EQU     6               Tracks per cylinder
000.360             00040   SPCYL   EQU     240             Sectors per cylinder (SPTRK*TPCYL)
000.364             00041   TOTCYL  EQU     244             Total cylinders in volume
344.300             00042   TOTSEC  EQU     58560           Total # sectors (SPCYL*TOTCYL)
                    00043   *
                    00044   *       Table parameters
                    00045   *
000.252             00046   MAXBS   EQU     170             Maximum # bad sectors in bad sector table
000.036             00047   SAT.UA  EQU     36Q             Sector Allocation Table (Unallocated sector flag)
000.037             00048   SAT.ET  EQU     37Q             Sector Allocation Table (End of table flag)
                    00049
000.170             00050   PDFLT   EQU     170Q            Default port to use (if none specified
                    00051   *
                    00052   *       Class 0 Command data structure offsets
                    00053   *
000.000             00054   C0.OPC  EQU     0               Op code
000.001             00055   C0.LUN  EQU     1               LUN (3 MSB bits)
000.001             00056   C0.AD2  EQU     1               adr2 (5 LSB bits)
000.002             00057   C0.AD1  EQU     2               adr1
```

```
000.003             00058   C0.AD0  EQU     3               adr0
000.004             00059   C0.NB   EQU     4               Number of blocks
000.005             00060   C0.CTL  EQU     5               Control
000.006             00061   C0.LEN  EQU     6               Length of command data block
                    00062
263.355             00063   MI.OTIR EQU     263355A         Z80 "OTIR" instruction
262.355             00064   MI.INIR EQU     262355A         Z80 "INIR" instruction
                    00065
042.200             00066           XTEXT   ASCII
                    00067X


                    00069X  **      ASCII CHARACTER EQUIVALENCES.
                    00070X
000.015             00071X  CR      EQU     13              CARRIAGE RETURN
000.012             00072X  LF      EQU     10              LINE FEED
000.200             00073X  NULL    EQU     200Q            PAD CHARACTER
000.000             00074X  NUL2    EQU     0
000.007             00075X  BELL    EQU     7               BELL CHARACTER
000.177             00076X  RUBOUT  EQU     177Q
000.010             00077X  BKSP    EQU     10Q             CTL-H
000.026             00078X  C.SYN   EQU     26Q             SYNC
000.002             00079X  C.STX   EQU     2               STX
000.047             00080X  QUOTE   EQU     47Q
000.011             00081X  TAB     EQU     11Q
000.033             00082X  ESC     EQU     33Q
000.012             00083X  NL      EQU     12Q             NEW LINE (HDOS SYSTEMS)
000.212             00084X  ENL     EQU     NL+200Q         NL + END-OF-LINE-FLAG
000.014             00085X  FF      EQU     14Q             FORM FEED
000.001             00086X  CTLA    EQU     01Q             CTL-A
000.002             00087X  CTLB    EQU     02Q             CTL-B
000.003             00088X  CTLC    EQU     03Q             CTL-C
000.004             00089X  CTLD    EQU     04Q             CTL-D
000.017             00090X  CTLO    EQU     17Q             CTL-O
000.020             00091X  CTLP    EQU     20Q             CTL-P
000.021             00092X  CTLQ    EQU     21Q             CTL-Q
000.023             00093X  CTLS    EQU     23Q             CTL-S
000.032             00094X  CTLZ    EQU     32Q             CTL-Z
042.200             00095           XTEXT   H67DEF
                    00096X  **      H67 Disk Controller Definitions
                    00097X  *



                    00099X  **      Register addresses
                    00100X  *
000.170             00101X  BASE    EQU     170Q            Controller base address
                    00102X
000.000             00103X  RI.DAT  EQU     0               Data In/Out (Read/Write)
000.001             00104X  RI.CON  EQU     1               Control (Write Only)
000.001             00105X  RI.BST  EQU     1               Bus Status (Read Only)
```

```
                    00107X  *       Control Register Definition
                    00108X
000.100             00109X  BC.SEL  EQU     01000000B       Select and data bit 0
000.040             00110X  BC.IE   EQU     00100000B       Interrupt Enable
000.020             00111X  BC.RST  EQU     00010000B       Reset
000.002             00112X  BC.EDT  EQU     00000010B       Enable Data



                    00114X  *       Bus Status Register Definition
                    00115X
000.200             00116X  BS.REQ  EQU     10000000B       Bus Transfer Request
000.100             00117X  BS.DTD  EQU     01000000B       Data Transfer Direction
000.000             00118X  BS.IN   EQU     00000000B        Data to Host
000.100             00119X  BS.OUT  EQU     01000000B        Data to Controller
000.040             00120X  BS.LMB  EQU     00100000B       Last byte in data/command string
000.020             00121X  BS.MTY  EQU     00010000B       Message type
000.000             00122X  BS.DAT  EQU     00000000B        Data
000.020             00123X  BS.COM  EQU     00010000B        Command
000.010             00124X  BS.BSY  EQU     00001000B       Busy
000.004             00125X  BS.INT  EQU     00000100B       Interrupt Pending
000.002             00126X  BS.PE   EQU     00000010B       Parity Error
000.001             00127X  BS.HID  EQU     00000001B       Hardware Identification



                    00129X  *       Status Byte Definitions
                    00130X
000.140             00131X  ST.LUN  EQU     01100000B       Logical Unit
000.034             00132X  ST.SPR  EQU     00011100B       Spare
000.002             00133X  ST.ERR  EQU     00000010B       Error
000.001             00134X  ST.PER  EQU     00000001B       Parity Error



                    00136X  **      Commands
                    00137X  *
                    00138X
000.340             00139X  CLASSM  EQU     11100000B       Class Mask
                    00140X
000.000             00141X  CLASS0  EQU     00000000B       Class 0
000.040             00142X  CLASS1  EQU     00100000B       Class 1
000.300             00143X  CLASS6  EQU     11000000B       Class 6
                    00144X
000.037             00145X  OPCODM  EQU     00011111B       Op-code Mask
000.140             00146X  LUNM    EQU     01100000B       Logical Unit Mask
000.037             00147X  LSA.2   EQU     00011111B       Logical Sector Address (2)
```

```
                    00149X  *       Class 0 Commands
                    00150X
000.000             00151X  D.TDR   EQU     CLASS0+0        Test drive ready
000.001             00152X  D.REC   EQU     CLASS0+1        Recalibrate drive
000.002             00153X  D.RSY   EQU     CLASS0+2        Request Syndrome
000.003             00154X  D.RSE   EQU     CLASS0+3        Request Sense
000.004             00155X  D.FOR   EQU     CLASS0+4        Format Drive
000.005             00156X  D.CTF   EQU     CLASS0+5        Check track format
000.006             00157X  D.FT    EQU     CLASS0+6        Format Track
000.007             00158X  D.FBS   EQU     CLASS0+7        Format bad sector
000.010             00159X  D.REA   EQU     CLASS0+8        Read
000.011             00160X  D.WPS   EQU     CLASS0+9        Write protect the sector
000.012             00161X  D.WRI   EQU     CLASS0+10       Write
000.013             00162X  D.SEK   EQU     CLASS0+11       Seek



                    00164X  *       Class 1 Commands
                    00165X
000.040             00166X  D.CP3   EQU     CLASS1+0        Copy block



                    00168X  *       Class 6 Commands
                    00169X
000.300             00170X  D.FFD   EQU     CLASS6+0        Format floppy disk



                    00172X  *       Type 0 error codes (Drive error Codes)
                    00173X
000.000             00174X  T0.NST  EQU     0               No status
000.001             00175X  T0.NIS  EQU     1               No Index signal
000.002             00176X  T0.NSC  EQU     2               No seek complete
000.003             00177X  T0.WFT  EQU     3               Write fault
000.004             00178X  T0.DNR  EQU     4               Drive not ready
000.005             00179X  T0.DNS  EQU     5               Drive not selected
000.006             00180X  T0.NT0  EQU     6               No track zero
000.007             00181X  TO.MDS  EQU     7               Mult-drive selected



                    00183X  *       Type 1 error codes (data error codes)
                    00184X
000.000             00185X  T1.ID   EQU     0               ID Read Error
000.001             00186X  T1.UDE  EQU     1               Uncorrectable data error
000.002             00187X  T1.IDNF EQU     2               ID Address Mark not found
000.003             00188X  T1.DMNF EQU     3               Data Address Mark Not FOund
000.004             00189X  T1.RNF  EQU     4               Record Not Found
000.005             00190X  T1.SKE  EQU     5               Seek Error
000.006             00191X  T1.DTE  EQU     6               DMA Time-out Error (not used)
000.007             00192X  T1.WP   EQU     7               Write protected
000.010             00193X  T1.CDE  EQU     8               Correctable Data field Error
```

```
000.011             00194X  T1.BBF  EQU     9               Bad Block Found
000.012             00195X  T1.FE   EQU     10              Format Error



                    00197X  *       Type 2 Error Codes (Command error codes)
                    00198X
000.000             00199X  T2.ILC  EQU     0               Illegal Command
000.001             00200X  T2.IDA  EQU     1               Illegal Disk Address
000.002             00201X  T2.IFN  EQU     2               Illegal Function
042.200             00202           XTEXT   COMP



                    00204X  **      $COMP - COMPARE TWO CHARACTER STRINGS.
                    00205X  *
                    00206X  *       $COMP COMPARES TWO BYTE STRINGS.
                    00207X  *
                    00208X  *       ENTRY   (C) = COMPARE COUNT
                    00209X  *               (DE) = FWA OF STRING #1
                    00210X  *               (HL) = FWA OF STRING #2
                    00211X  *       EXIT    'Z' CLEAR, IS MIS-MATCH
                    00212X  *                (C) = LENGTH REMAINING
                    00213X  *                (DE) = ADDRESS OF MISMATCH IN STRING#1
                    00214X  *                (HL) = ADDRESS OF MISMATCH IN STRING #2
                    00215X  *               'C' SET, HAVE MATCH
                    00216X  *                (C) = 0
                    00217X  *                (DE) = (DE) + (0C)
                    00218X  *                (HL) = (HL) = (0C)
                    00219X  *       USES    A,F,C,D,E,H,L
                    00220X
                    00221X
030.060             00222X  $COMP   EQU     30060A          IN H17 ROM
042.200             00223           XTEXT   TYPTX



                    00225X  **      $TYPTX - TYPE TEXT.
                    00226X  *
                    00227X  *       $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
                    00228X  *
                    00229X  *       IMBEDDED ZER0 BYTES INDICATE A CARRIAGE RETURN LINE FEED,
                    00230X  *       A BYTE WITH THE 200Q BIT SET IS THE LAST BYTE IN THE MESSAGE.
                    00231X  *
                    00232X  *       ENTRY   (RET) = TEXT
                    00233X  *       EXIT    TO (RET+LENGTH)
                    00234X  *       USES    A,F
                    00235X
                    00236X
031.136             00237X  $TYPTX  EQU     31136A          IN H17 ROM
                    00238X
031.144             00239X  $TYPTX. EQU     31144A          IN H17 ROM
042.200             00240           XTEXT   HOSEQU
                    00241X
```

```
                00243X  **      HDOS SYSTEM EQUIVALENCES.
                00244X  *
                00245X
024.000         00246X  S.GRT0  EQU     24000A          SYSTEM AREA FOR  GRT0
025.000         00247X  S.GRT1  EQU     25000A          SYSTEM AREA FOR  GRT1
026.000         00248X  S.GRT2  EQU     26000A          SYSTEM AREA FOR  GRT2
                00249X
030.000         00250X  ROMBOOT EQU     30000A          ROM BOOT ENTRY
                00251X
040.100         00252X          ORG     40100A          FREE SPACE FROM PAM-8
                00253X
040.100         00254X          DS      8               JUMP TO SYSTEM EXIT
040.110         00255X  D.CON   DS      16              DISK CONSTANTS
040.130         00256X  SYDD    EQU     *               SYSTEM DISK ENTRY POINT
040.130         00257X  D.VEC   DS      24*3            SYSTEM ROM ENTRY VECTORS
040.240         00258X  D.RAM   DS      31              SYSTEM ROM WORK AREA
040.277         00259X  S.VAL   DS      36              SYSTEM VALUES
040.343         00260X  S.INT   DS      115             SYSTEM INTERNAL WORK AREAS
041.126         00261X          DS      16
041.146         00262X  S.SOVR  DS      2               STACK OVERFLOW WARNING
041.150         00263X          DS      42200A-*        SYSTEM STACK
001.032         00264X  STACKL  EQU     *-S.SOVR        STACK SIZE
                00265X
042.200         00266X  STACK   EQU     *               LWA+1 SYSTEM STACK
042.200         00267X  USERFWA EQU     *               USER FWA
042.200         00268           XTEXT   HOSDEF
                00269X



                00271X  **      HOSDEF - DEFINE HOS PARAMETER.
                00272X  *
                00273X
                00274X
000.040         00275X  VERS    EQU     2*16+0          VERSION 2.0
                00276X
000.377         00277X  SYSCALL EQU     377Q            SYSCALL INSTRUCTION
                00278X
                00279X
000.000         00280X          ORG     0
                00281X
                00282X  *       RESIDENT FUNCTIONS
                00283X
000.000         00284X  .EXIT   DS      1               EXIT (MUST BE FIRST)
000.001         00285X  .SCIN   DS      1               SCIN
000.002         00286X  .SCOUT  DS      1               SCOUT
000.003         00287X  .PRINT  DS      1               PRINT
000.004         00288X  .READ   DS      1               READ
000.005         00289X  .WRITE  DS      1               WRITE
000.006         00290X  .CONSL  DS      1               SET/CLEAR CONSOLE OPTIONS
000.007         00291X  .CLRCO  DS      1               CLEAR CONSOLE BUFFER
000.010         00292X  .LOADO  DS      1               LOAD AN OVERLAY
000.011         00293X  .VERS   DS      1               RETURN HDOS VERSION NUMBER
000.012         00294X  .SYSRES DS      1               PRECEDING FUNCTIONS ARE RESIDENT
                00295X
                00296X
                00297X  *       *HDOSOVL0.SYS*  FUNCTIONS
```

```
                00298X
000.040         00299X          ORG     40A
                00300X
000.040         00301X  .LINK   DS      1               LINK  (MUST BE FIRST)
000.041         00302X  .CTLC   DS      1               CTL-C
000.042         00303X  .OPENR  DS      1               OPENR
000.043         00304X  .OPENW  DS      1               OPENW
000.044         00305X  .OPENU  DS      1               OPENU
000.045         00306X  .OPENC  DS      1               OPENC
000.046         00307X  .CLOSE  DS      1               CLOSE
000.047         00308X  .POSIT  DS      1               POSITION
000.050         00309X  .DELET  DS      1               DELETE
000.051         00310X  .RENAM  DS      1               RENAME
000.052         00311X  .SETTP  DS      1               SETTOP
000.053         00312X  .DECODE DS      1               NAME DECODE
000.054         00313X  .NAME   DS      1               GET FILE NAME FROM CHANNEL
000.055         00314X  .CLEAR  DS      1               CLEAR CHAN
000.056         00315X  .CLEARA DS      1               CLEAR ALL CHANS
000.057         00316X  .ERROR  DS      1               LOOKUP ERROR
000.060         00317X  .CHFLG  DS      1               CHANGE FLAGS
000.061         00318X  .DISMT  DS      1               FLAG SYSTEM DISK DISMOUNTED
000.062         00319X  .LOADD  DS      1               LOAD DEVICE DRIVER
000.063         00320X  .OPEN   DS      1               Parametrized Open
                00321X
                00322X
                00323X  *       *HDOSOVL1.SYS*  FUNCTIONS
                00324X
000.200         00325X          ORG     200Q
                00326X
000.200         00327X  .MOUNT  DS      1               MOUNT  (MUST BE FIRST)
000.201         00328X  .DMOUN  DS      1               DISMOUNT
000.202         00329X  .MONMS  DS      1               MOUNT/NO MESSAGE
000.203         00330X  .DMNMS  DS      1               DISMOUNT/NO MESSAGE
000.204         00331X  .RESET  DS      1               RESET = DISMOUNT/MOUNT OF UNIT
000.205         00332X  .CLEAN  DS      1               Clean device
000.206         00333X  .DAD    DS      1               Dismount All Disks              /80.08.gc/
000.207         00334           XTEXT   DIRDEF
                00335X



                00337X  **      DIRECTORY ENTRY FORMAT.
                00338X
000.000         00339X          ORG     0
                00340X
                00341X
000.377         00342X  DF.EMP  EQU     377Q            FLAGS ENTRY EMPTY
000.376         00343X  DF.CLR  EQU     376Q            FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
                00344X
000.000         00345X  DIR.NAM DS      8               NAME
000.010         00346X  DIR.EXT DS      3               EXTENSION
000.013         00347X  DIR.PRO DS      1               PROJECT
000.014         00348X  DIR.VER DS      1               VERSION
000.015         00349X  DIRIDL  EQU     *               FILE IDENTIFICATION LENGTH
                00350X
000.015         00351X  DIR.CLU DS      1               CLUSTER FACTOR
000.016         00352X  DIR.FLG DS      1               FLAGS
```

```
000.017             00353X          DS      1               RESERVED
000.020             00354X  DIR.FGN DS      1               FIRST GROUP NUMBER
000.021             00355X  DIR.LGN DS      1               LAST GROUP NUMBER
000.022             00356X  DIR.LSI DS      1               LAST SECTOR INDEX (IN LAST GROUP)
000.023             00357X  DIR.CRD DS      2               CREATION DATE
000.025             00358X  DIR.ALD DS      2               LAST ALTERATION DATE
                    00359X
000.027             00360X  DIRELEN EQU     *               DIRECTORY ENTRY LENGTH
000.027             00361           XTEXT   ESINT



                    00363X  **      S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
                    00364X  *
                    00365X  *       THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
                    00366X  *       MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
                    00367X
                    00368X
040.343             00369X          ORG     S.INT
                    00370X
                    00371X  **      CONSOLE STATUS FLAGS
                    00372X
040.343             00373X  S.CDB   DS      1               CONSOLE DESCRIPTOR BYTE
000.000             00374X  CDB.H85 EQU     00000000B
000.001             00375X  CDB.H84 EQU     00000001B       =0 IF H8-5, =1 IF H8-4
040.344             00376X  S.BAUD  DS      2               [0-14]  H8-4 BAUD RATE, =0 IF H8-5
                    00377X  *                               [15]    =1 IF BAUD RATE => 2 STOP BITS
                    00378X
                    00379X  **      TABLE ADDRESS WORDS
                    00380X
040.346             00381X  S.DLINK DS      2               ADDRESS OF DATA IN HDOS CODE
040.350             00382X  S.OFWA  DS      2               FWA  OVERLAY  TABLE
040.352             00383X  S.CFWA  DS      2               FWA  CHANNEL  TABLE
040.354             00384X  S.DFWA  DS      2               FWA  DEVICE   TABLE
040.356             00385X  S.RFWA  DS      2               FWA  RESIDENT HDOS CODE
                    00386X
                    00387X  **      DEVICE DRIVER DELAYED LOAD FLAGS
                    00388X
040.360             00389X  S.DDLDA DS      2               DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
040.362             00390X  S.DDLEN DS      2               CODE LENGTH IN BYTES
040.364             00391X  S.DDGRP DS      1               GROUP NUMBER FOR DRIVER
040.365             00392X          DS      1               HOLD PLACE
                    00393X  *S.DDSEC        DS      2               SECTOR NUMBER FOR DRIVER ( * OBSOLETE ! * )
040.366             00394X  S.DDDTA DS      2               DEVICE'S ADDRESS IN DEVLST +DEV.RES
040.370             00395X  S.DDOPC DS      1               OPEN OPCODE PENDEDING
                    00396X
                    00397X  **      OVERLAY MANAGEMENT FLAGS
                    00398X
000.001             00399X  OVL.IN  EQU     00000001B       IN MEMORY
000.002             00400X  OVL.RES EQU     00000010B       PERMINANTLY RESIDENT
000.014             00401X  OVL.NUM EQU     00001100B       OVERLAY NUMBER MASK
000.200             00402X  OVL.UCS EQU     10000000B       USER CODE SWAPPED FOR OVERLAY
                    00403X
040.371             00404X  S.OVLFL DS      1               OVERLAY FLAG
040.372             00405X  S.UCSF  DS      2               FWA SWAPPED USER CODE
040.374             00406X  S.UCSL  DS      2               LENGTH SWAPPED USER CODE
```

```
040.376             00407X  S.OVLS  DS      2               SIZE OF OVERLAY CODE
041.000             00408X  S.OVLE  DS      2               ENTRY POINT OF OVERLAY CODE
                    00409X
041.002             00410X  S.SSN   DS      2               SWAP AREA SECTOR NUMBER
041.004             00411X  S.OSN   DS      2               OVERLAY SECTOR NUMBER
                    00412X
                    00413X  *       SYSCALL PROCESSING WORK AREAS
                    00414X
041.006             00415X  S.CACC  DS      1               (ACC) UPON SYSCALL
041.007             00416X  S.CODE  DS      1               SYSCALL INDEX IN PROGRESS
                    00417X
                    00418X  *       JUMPS TO ROUTINES IN RESIDENT HDOS CODE
                    00419X
041.010             00420X  S.JUMPS DS      0               START OF DUMP VECTORS
041.010             00421X  S.SDD   DS      3               JUMP TO STAND-IN DEVICE DRIVER
041.013             00422X  S.FASER DS      3               JUMP TO FATSERR (FATAL SYSTEM ERROR)
041.016             00423X  S.DIREA DS      3               JUMP TO DIREAD (DISK FILE READ)
041.021             00424X  S.FCI   DS      3               JUMP TO FCI (FETCH CHANNEL INFO)
041.024             00425X  S.SCI   DS      3               JUMP TO SCI (STORE CHANNEL INFO)
041.027             00426X  S.GUP   DS      3               JUMP TO GUP (GET UNIT POINTER)
                    00427X
041.032             00428X  S.MOUNT DS      1               <>0 IF THE SYSTEM DISK IS MOUNTED
041.033             00429X  S.DCS   DS      1               DEFAULT CLUSTER SIZE-1
                    00430X
041.034             00431X  S.BOOTF DS      1               BOOT FLAGS
000.001             00432X  BOOT.P  EQU     00000001B       EXECUTE PROLOGUE UPON BOOTUP
                    00433X
                    00434X  *       STACK VALUE SAVED FOR OVERLAY SYSCALLS
                    00435X
041.035             00436X  S.OVSTK DS      2               VALUE OF SP UPON SYSCALLS USING OVERLAY
                    00437X
041.037             00438X          DS      1               RESERVED



                    00440X  **      ACTIVE I/O AREA.
                    00441X  *
                    00442X  *       THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
                    00443X  *       CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
                    00444X  *       THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
                    00445X  *
                    00446X  *       NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
                    00447X  *       FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
                    00448X  *       8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
                    00449X  *       COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
                    00450X  *       BACKDATED AFTER PROCESSING.
                    00451X
041.040             00452X  AIO.VEC DS      3               JUMP INSTRUCTION
041.041             00453X  AIO.DDA EQU     *-2             DEVICE DRIVER ADDRESS
041.043             00454X  AIO.FLG DS      1               FLAG BYTE
041.044             00455X  AIO.GRT DS      2               ADDRESS OF GROUP RESERV TABLE
041.046             00456X  AIO.SPG DS      1               SECTORS PER GROUP
041.047             00457X  AIO.CGN DS      1               CURRENT GROUP NUMBER
041.050             00458X  AIO.CSI DS      1               CURRENT SECTOR INDEX
041.051             00459X  AIO.LGN DS      1               LAST GROUP NUMBER
041.052             00460X  AIO.LSI DS      1               LAST SECTOR INDEX
```

```
041.053                 00461X  AIO.DTA DS      2               DEVICE TABLE ADDRESS
041.055                 00462X  AIO.DES DS      2               DIRECTORY SECTOR
041.057                 00463X  AIO.DEV DS      2               DEVICE CODE
041.061                 00464X  AIO.UNI DS      1               UNIT NUMBER (0-9)
                        00465X
041.062                 00466X  AIO.DIR DS      DIRELEN         DIRECTORY ENTRY
                        00467X
041.111                 00468X  AIO.CNT DS      1               SECTOR COUNT
041.112                 00469X  AIO.EOM DS      1               END OF MEDIA FLAG
041.113                 00470X  AIO.EOF DS      1               END OF FILE FLAG
041.114                 00471X  AIO.TFP DS      2               TEMP FILE POINTERS
041.116                 00472X  AIO.CHA DS      2               ADDRESS OF CHANNEL BLOCK (IOC.DDA)



041.120                 00474X  S.BDA   DS      1               Boot Device Address (Setup by ROM) /80.09.gc/
041.121                 00475X  S.SCR   DS      2               SYSTEM SCRATCH AREA ADDRESS
                        00476
042.200                 00477           ORG     USERFWA
                        00478
042.200  041 052 046   00479   START   LXI     H,EXIT          set up CC handler
042.203  076 003       00480           MVI     A,3
042.205  377 041       00481           SCALL   .CTLC
                        00482   *
                        00483   *       get port parameter from command line (if any)
                        00484   *
042.207  041 000 000   00485           LXI     H,0
042.212  071           00486           DAD     SP              HL = SP (points to cmd line parameters)
042.213  175           00487           MOV     A,L             Get first byte
042.214  376 200       00488           CPI     200Q            anything there?
042.216  302 231 042   00489           JNZ     GETPRM          Yes, get it
                        00490   *
                        00491   *       no port specified,use default
                        00492   *
042.221  076 170       00493           MVI     A,PDFLT         default port
042.223  062 311 056   00494           STA     BASE67          store it
042.226  303 247 042   00495           JMP     INTRO
                        00496   *
                        00497   *       Get port parameter from command line
                        00498   *
042.231  176           00499   GETPRM  MOV     A,M             get a byte
042.232  376 040       00500           CPI     ' '             space?
042.234  043           00501           INX     H               and point to next
042.235  312 231 042   00502           JZ      GETPRM          if space, keep looking
042.240  053           00503           DCX     H               non space (fix pointer)
                        00504   *
                        00505   *       Convert ASCII string to octal port value
                        00506   *
042.241  021 311 056   00507           LXI     D,BASE67        DE = location to store port
042.244  315 204 056   00508           CALL    GETVAL
                        00509
                        00510
042.247  315 136 031   00511   INTRO   CALL    $TYPTX
042.252  012 120 122   00512           DB      LF,'PREP67 for the Winchester disk.'
042.312  012 103 157   00513           DB      LF,'Copyright (c) 1981 Heath/Zenith Data Systems.'
042.370  012 012 124   00514           DB      LF,LF,'This routine is used to:'
```

```
043.022  012 012 011 00515           DB      LF,LF,TAB,'1. Initialize the Winchester disk.'
043.067  012 011 062 00516           DB      LF,TAB,'2. Perform media check on the Winchester disk surface.'
043.157  012 011 063 00517           DB      LF,TAB,'3. Initialize tables for use with the '
043.227  160 141 162 00518           DB      'partitioning utility PART.'
043.261  012 012 120 00519           DB      LF,LF,'PREP67 is a stand-alone utility. '
043.324  111 164 040 00520           DB      'It will destroy all files on'
043.360  012 164 150 00521           DB      LF,'the Winchester disk.'
044.005  012 012 104 00522           DB      LF,LF,'Do not use PREP67 until you have made a backup '
044.066  157 146 040 00523           DB      'of the files'
044.102  012 143 165 00524           DB      LF,'currently on the Winchester disk.'
044.144  012 012 120 00525           DB      LF,LF,'Proceed(yes/no)?',' '+200Q
                     00526
044.167  041 332 056 00527           LXI     H,RTBUFF                point to buffer
044.172  315 075 056 00528           CALL    $RTL.                   read text line
044.175  021 344 056 00529           LXI     D,ASCYES                is it "YES"?
044.200  016 003     00530           MVI     C,3                     compare 3 characters
044.202  315 060 030 00531           CALL    $COMP
044.205  302 052 046 00532           JNZ     EXIT                    no, EXIT immediately...
                     00533
044.210  315 136 031 00534           CALL    $TYPTX
044.213  012 125 163 00535           DB      LF,'Using port <',200Q
044.231  021 311 056 00536           LXI     D,BASE67
044.234  315 237 056 00537           CALL    PROVAL
044.237  315 136 031 00538           CALL    $TYPTX
044.242  121 076 012 00539           DB      'Q>',NL,ENL
044.246  315 136 031 00540           CALL    $TYPTX                  else, prompt to proceed...
044.251  012 120 154 00541           DB      LF,'Please type P to proceed:',' '+200Q
                     00542
044.304  041 332 056 00543           LXI     H,RTBUFF                point to buffer
044.307  315 075 056 00544           CALL    $RTL.                   read text line
044.312  021 347 056 00545           LXI     D,ASCIIP                is it "P"?
044.315  016 001     00546           MVI     C,1                     compare 1 character
044.317  315 060 030 00547           CALL    $COMP
044.322  302 052 046 00548           JNZ     EXIT                    no, EXIT immediately
                     00549   *
                     00550   *       Initialize buffers and data structures
                     00551   *
044.325  072 311 056 00552           LDA     BASE67                  get base port number
044.330  062 251 051 00553           STA     RDDATA+1                set port to read data from
044.333  062 254 051 00554           STA     WRDATA+1                set port to write data to
044.336  074         00555           INR     A                       control/status is one more than r/w
044.337  062 246 051 00556           STA     WRCTRL+1                set port for control status write
044.342  062 243 051 00557           STA     RDSTAT+1                set port for reading status
                     00558
044.345  257         00559           XRA     A                       A = 0
044.346  062 375 064 00560           STA     CDRBLK+C0.AD2           logical addr 2 = 0
044.351  076 300     00561           MVI     A,11000000B             disable data error correction & retry
044.353  062 001 065 00562           STA     CDRBLK+C0.CTL           control
                     00563
044.356  315 143 046 00564           CALL    FILPAT                  fill buffer with 011011...
                     00565
044.361  257         00566           XRA     A                       Zero error counts
044.362  062 352 056 00567           STA     ERRCT1
044.365  062 353 056 00568           STA     ERRCT2
044.370  062 354 056 00569           STA     ERRCT3
                     00570
044.373  062 373 064 00571           STA     SECCNT                  Zero sector counter
```

```
                    00572
044.376  107       00573           MOV     B,A                     Fill byte = 0
044.377  041 371 056 00574           LXI     H,BUFF6
045.002  021 000 006 00575           LXI     D,6*SECSIZ              6 sectors
045.005  315 045 050 00576           CALL    FILBYT                  Fill with NUL
                    00577
045.010  041 264 065 00578           LXI     H,WRKBUF                set up buffer pointers
045.013  042 060 065 00579           SHLD    UNK191
045.016  041 371 056 00580           LXI     H,BUFF6
045.021  042 364 056 00581           SHLD    BSTPTR                  next entry in bad sector table
                    00582
045.024  041 000 000 00583           LXI     H,0
045.027  042 362 056 00584           SHLD    BSCOUNT                 zero counters
045.032  042 360 056 00585           SHLD    CURSEC
                    00586
045.035  315 314 053 00587           CALL    RESET                   reset the controller
045.040  315 310 046 00588           CALL    TSTRDY                  drive ready?
045.043  315 106 047 00589           CALL    DD.REC                  recalibrate
045.046  315 200 046 00590           CALL    S3200                   Seek sector #3200
045.051  315 106 047 00591           CALL    DD.REC                  recalibrate
                    00592
045.054  315 136 031 00593           CALL    $TYPTX
045.057  012 111 156 00594           DB      LF,'Initialize the disk...',LF,LF+200Q
                    00595
045.110  315 370 046 00596           CALL    FORDRV                  format drive
045.113  315 122 047 00597           CALL    DD.CTF                  Check track format (0x05)
045.116  322 144 045 00598           JNC     MAIN2                   normal return is for 'C' to be set! (no jump)
045.121  315 114 047 00599           CALL    DD.RSE                  process request sense (to look up error)
045.124  072 052 065 00600           LDA     DSENSE                  4-byte error data structure
045.127  376 040     00601           CPI     00100000B+T2.ILC        Illegal Command? (Type II error)
045.131  312 140 045 00602           JZ      MAIN1                   yes! (that's good - should be! ??)
045.134  257       00603           XRA     A                       no, clear 'C'
045.135  303 144 045 00604           JMP     MAIN2                   wrong drive type, abort.
                    00605
045.140  067       00606   MAIN1   STC
045.141  303 144 045 00607           JMP     MAIN2
                    00608   *
                    00609   *       if 'C' set then we got an "illegal command" from DT.CTF (0x05) call
                    00610   *
045.144  332 217 045 00611   MAIN2   JC      MAIN3                   'C' expected, jump to proceed
                    00612   *
                    00613   *       Wrong drive type, abort
                    00614   *
045.147  315 136 031 00615           CALL    $TYPTX
045.152  012 127 162 00616           DB      LF,'Wrong drive type.'
045.174  012 124 145 00617           DB      LF,'Test aborted.',LF,LF+200Q
045.214  303 052 046 00618           JMP     EXIT
                    00619   *
                    00620   *       Media test
                    00621   *
045.217  315 136 031 00622   MAIN3   CALL    $TYPTX
045.222  115 145 144 00623           DB      'Media test in progress...'
045.253  012 012 124 00624           DB      LF,LF,'Testing cylinder 000',CR+200Q
                    00625
045.302  076 001     00626   MAIN4   MVI     A,1                     One block
045.304  315 057 047 00627           CALL    RTN57                   load block address
045.307  315 171 047 00628           CALL    PREAD                   read it
```

```
045.312  322 323 045 00629          JNC     MAIN5                   OK, continue
                     00630
045.315  315 056 050 00631          CALL    HAVEBS                  read error - bad sector!
045.320  303 356 045 00632          JMP     MAIN8                   keep looping...
                     00633
045.323  315 245 047 00634  MAIN5   CALL    PWRITE                  write it
045.326  322 337 045 00635          JNC     MAIN6                   OK, continue
                     00636
045.331  315 056 050 00637          CALL    HAVEBS                  write error - bad sector!
045.334  303 356 045 00638          JMP     MAIN8                   keep looping...
                     00639
045.337  315 171 047 00640  MAIN6   CALL    PREAD                   read it
045.342  322 353 045 00641          JNC     MAIN7                   OK, continue
                     00642
045.345  315 056 050 00643          CALL    HAVEBS                  read error - bad sector!
045.350  303 356 045 00644          JMP     MAIN8                   keep looping...
                     00645
045.353  315 015 050 00646  MAIN7   CALL    CMPSEC                  compare (and mark bad if mismatch)
                     00647
045.356  052 360 056 00648  MAIN8   LHLD    CURSEC                  increment sector counter
045.361  043         00649          INX     H
045.362  042 360 056 00650          SHLD    CURSEC
045.365  021 300 344 00651          LXI     D,TOTSEC                total number of sectors
045.370  315 055 056 00652          CALL    HLCPDE
045.373  322 004 046 00653          JNC     MAIN9                   yes.
045.376  315 055 046 00654          CALL    UPDCYL                  update "testing cylinder..." (if needed)
046.001  303 302 045 00655          JMP     MAIN4                   and continue...
                     00656  *
                     00657  *       now prepare and write critical tables to disk
                     00658  *
046.004  315 322 053 00659  MAIN9   CALL    RTN123
046.007  315 145 054 00660          CALL    ZTBLS                   Zero out key tables
046.012  315 233 054 00661          CALL    GENBST                  Create and write Bad Sector Tables
046.015  315 342 054 00662          CALL    GENSBK                  Create and write Superblocks
046.020  315 053 055 00663          CALL    WRTSBC                  write SBC sector
                     00664  *
                     00665  *       Finished! print message and exit normally
                     00666  *
046.023  315 136 031 00667          CALL    $TYPTX
046.026  012 012 120 00668          DB      LF,LF,'PREP67 complete.'
046.050  012 212     00669          DB      LF,LF+200Q
                     00670  *
                     00671  *       EXIT program
                     00672  *
046.052  257         00673  EXIT    XRA     A
046.053  377 000     00674          SCALL   .EXIT
                     00675
                     00676
                     00677  *
                     00678  *       UPDCYL - update "testing cylinder..." message (if needed)
                     00679  *
046.055  072 373 064 00680  UPDCYL  LDA     SECCNT                  increment sector count
046.060  074         00681          INR     A
046.061  062 373 064 00682          STA     SECCNT
046.064  376 360     00683          CPI     SPCYL                   compare to sectors/cylinder
046.066  330         00684          RC                              ok, keep going
                     00685  *
```

```
                    00686   *       Increment cylinder count in message
                    00687   *
046.067  041 141 046 00688           LXI     H,TSTCYL                point to cylinder number (ASCII)
046.072  064         00689   UPDCY1  INR     M                       increment it
046.073  176         00690           MOV     A,M                     get the count
046.074  376 072     00691           CPI     '9'+1
046.076  332 107 046 00692           JC      UPDCY2
046.101  066 060     00693           MVI     M,'0'
046.103  053         00694           DCX     H
046.104  303 072 046 00695           JMP     UPDCY1
                    00696
046.107  257         00697   UPDCY2  XRA     A
046.110  062 373 064 00698           STA     SECCNT
046.113  315 136 031 00699           CALL    $TYPTX
046.116  124 145 163 00700           DB      'Testing cylinder 00'
046.141  060 215     00701   TSTCYL  DB      '0',CR+200Q
                    00702
                    00703
                    00704   *
                    00705   *       Fill 256 bytes with pattern 011011011011011011011011
                    00706   *
046.143  041 264 065 00707   FILPAT  LXI     H,WRKBUF                buffer
046.146  006 000     00708           MVI     B,0                     count = 256
046.150  076 155     00709   FILLP1  MVI     A,01101101B             first 1/3 of pattern
046.152  315 172 046 00710           CALL    STAINC                  store it
046.155  076 266     00711           MVI     A,10110110B             second 1/3 of pattern
046.157  315 172 046 00712           CALL    STAINC                  store it
046.162  076 333     00713           MVI     A,11011011B             third 1/3 of pattern
046.164  315 172 046 00714           CALL    STAINC                  store it
046.167  303 150 046 00715           JMP     FILLP1                  loop 'til done
                    00716   *
                    00717   *       STAINC - Store A in memory and increment
                    00718   *
                    00719   *       ENTRY:  A = data value
                    00720   *               B = counter
                    00721   *               HL = memory location
                    00722   *
                    00723   *       EXIT:   B = count (decremented)
                    00724   *               HL = location (incremented)
                    00725   *               'Z' set if done
                    00726   *
046.172  167         00727   STAINC  MOV     M,A
046.173  043         00728           INX     H
046.174  005         00729           DCR     B
046.175  300         00730           RNZ
046.176  341         00731           POP     H                       tricky return
046.177  311         00732           RET
                    00733   *
                    00734   *       Seek logical address 3200
                    00735   *
046.200  076 013     00736   S3200   MVI     A,D.SEK                 Seek
046.202  062 374 064 00737           STA     CDRBLK+C0.OPC           Set op code
046.205  076 014     00738           MVI     A,14Q                   Logical address hi byte
046.207  062 376 064 00739           STA     CDRBLK+C0.AD1
046.212  076 200     00740           MVI     A,200Q                  Logical address lo byte
046.214  062 377 064 00741           STA     CDRBLK+C0.AD0
046.217  076 001     00742           MVI     A,1
```

```
046.221  062 000 065 00743           STA     CDRBLK+C0.NB            1 block
046.224  315 256 051 00744           CALL    DOCMD
046.227  320         00745           RNC                             normal exit
                     00746   *
                     00747   *       error
                     00748   *
046.230  315 251 046 00749           CALL    TSTRSE
046.233  315 200 046 00750           CALL    S3200
046.236  320         00751           RNC
                     00752   *
                     00753   *       again
                     00754   *
046.237  315 251 046 00755           CALL    TSTRSE
046.242  315 200 046 00756           CALL    S3200
046.245  320         00757           RNC
                     00758   *
                     00759   *       give up!
                     00760   *
046.246  332 331 050 00761           JC      ERRIE                   Erro - Internal error
                     00762
                     00763   *
                     00764   *       Test Request Sense
                     00765   *
046.251  315 114 047 00766   TSTRSE  CALL    DD.RSE                  process request sense
046.254  072 052 065 00767           LDA     DSENSE                  check first byte
046.257  346 177     00768           ANI     01111111B               clear Block Address Valid bit
046.261  107         00769           MOV     B,A
046.262  376 002     00770           CPI     T1.IDNF                 ID not found
046.264  310         00771           RZ
046.265  170         00772           MOV     A,B
046.266  376 027     00773           CPI     27Q
046.270  332 025 000 00774           JC      000025A                 ??
046.273  170         00775           MOV     A,B
046.274  376 030     00776           CPI     00010000B+T1.CDE        correctable data field error
046.276  310         00777           RZ
046.277  170         00778           MOV     A,B
046.300  376 031     00779           CPI     00010000B+T1.BBF        bad block found
046.302  310         00780           RZ
046.303  303 331 050 00781           JMP     ERRIE                   Error - Internal error
                     00782
046.306  257         00783           XRA     A                       unreachable?
046.307  311         00784           RET
                     00785   *
                     00786   *       TSTRDY - test if drive is ready
                     00787   *
046.310  076 000     00788   TSTRDY  MVI     A,D.TDR                 Test for Drive Ready
046.312  315 033 047 00789           CALL    CMD00                   issue command...
046.315  320         00790           RNC
                     00791
046.316  315 356 046 00792           CALL    DELAY                   wait a bit...
046.321  315 356 046 00793           CALL    DELAY
046.324  315 356 046 00794           CALL    DELAY
                     00795
046.327  315 314 053 00796           CALL    RESET                   try again
046.332  076 000     00797           MVI     A,0
046.334  315 033 047 00798           CALL    CMD00
046.337  320         00799           RNC
```

```
                    00800   *
                    00801   *       second time didn't work either, error!
                    00802   *
046.340  315 114 047 00803           CALL    DD.RSE                  process request sense
046.343  072 052 065 00804           LDA     DSENSE
046.346  346 060     00805           ANI     60Q
046.350  312 002 051 00806           JZ      ERRDNR
046.353  303 331 050 00807           JMP     ERRIE                   Error - Internal error
                    00808   *
                    00809   *       delay counter, count down from max
                    00810   *
046.356  001 377 377 00811   DELAY   LXI     B,377377A
046.361  013         00812   DELAY1  DCX     B
046.362  170         00813           MOV     A,B
046.363  261         00814           ORA     C
046.364  302 361 046 00815           JNZ     DELAY1
046.367  311         00816           RET
                    00817   *
                    00818   *       Format drive
                    00819   *
046.370  315 044 047 00820   FORDRV  CALL    ZEROAD                  Block zero, nblocks = 0
046.373  076 004     00821           MVI     A,D.FOR                 Format
046.375  062 374 064 00822           STA     CDRBLK+C0.OPC           op code = format
047.000  062 000 065 00823           STA     CDRBLK+C0.NB            A = number of blocks
047.003  315 256 051 00824           CALL    DOCMD
047.006  320         00825           RNC
                    00826   *
                    00827   *       error in formatting
                    00828   *
047.007  315 114 047 00829           CALL    DD.RSE                  process request sense
047.012  072 052 065 00830           LDA     DSENSE
047.015  376 003     00831           CPI     3
047.017  302 240 050 00832           JNZ     ERRFMT                  Error - during formatting
047.022  315 106 047 00833           CALL    DD.REC
047.025  322 136 050 00834           JNC     ERRWP                   Error - write protected
047.030  303 331 050 00835           JMP     ERRIE                   Error - internal error
                    00836   *
                    00837   *       CMD00 - issue command to the controller using
                    00838   *               block zero and 0 block count
                    00839   *
                    00840   *       ENTRY:  A = command byte
                    00841   *
047.033  062 374 064 00842   CMD00   STA     CDRBLK+C0.OPC
047.036  315 044 047 00843           CALL    ZEROAD                  Block zero, nblocks = 0
047.041  303 256 051 00844           JMP     DOCMD
                    00845   *
                    00846   *       Zero block address and # blocks
                    00847   *
047.044  041 000 000 00848   ZEROAD  LXI     H,0
047.047  042 376 064 00849           SHLD    CDRBLK+C0.AD1           store 2 bytes (hi and lo add)
047.052  257         00850           XRA     A
047.053  062 000 065 00851           STA     CDRBLK+C0.NB            num. blocks
047.056  311         00852           RET
                    00853
                    00854
047.057  062 000 065 00855   RTN57   STA     CDRBLK+C0.NB            num. blocks
047.062  052 360 056 00856           LHLD    CURSEC
```

```
047.065  315 040 056 00857           CALL    HLSWAP                  H <-> L
047.070  042 376 064 00858           SHLD    CDRBLK+C0.AD1           hi address
047.073  311         00859           RET
                     00860   *
                     00861   *       DOREC - recalibrate
                     00862   *
047.074  315 314 053 00863   DOREC   CALL    RESET                   reset
047.077  315 106 047 00864           CALL    DD.REC                  recalibrate
047.102  320         00865           RNC
047.103  332 331 050 00866           JC      ERRIE                   Error - Internal error
                     00867   *
                     00868   *       Recalibrate.
                     00869   *
                     00870   *       Positions the R/W arm to Track 00, clears possible
                     00871   *       error status in the drive.
                     00872   *
047.106  076 001     00873   DD.REC  MVI     A,D.REC                 recalibrate
047.110  315 033 047 00874           CALL    CMD00
047.113  311         00875           RET
                     00876   *
                     00877   *       Process Request Sense
                     00878   *
                     00879   *       This command normally issued immediately after an error.  It returns
                     00880   *       4 bytes of drive and controller sense for the specified LUN
                     00881   *       (see copy block for exception)
                     00882   *
047.114  076 003     00883   DD.RSE  MVI     A,D.RSE                 Request sense
047.116  315 033 047 00884           CALL    CMD00
047.121  311         00885           RET
                     00886   *
                     00887   *       Check track format
                     00888   *
047.122  076 005     00889   DD.CTF  MVI     A,D.CTF                 Check track format
047.124  315 033 047 00890           CALL    CMD00
047.127  311         00891           RET
                     00892   *
                     00893   *       BEEEEP - Make six consecutive beeps on console
                     00894   *
047.130  315 136 031 00895   BEEEEP  CALL    $TYPTX
047.133  007 007 007 00896           DB      BELL,BELL,BELL,BELL,BELL,BELL,LF+200Q
047.142  311         00897           RET
                     00898   *
                     00899   *       SAVCDB - save command block
                     00900   *
047.143  006 006     00901   SAVCDB  MVI     B,C0.LEN                block length
047.145  041 374 064 00902           LXI     H,CDRBLK                from here
047.150  021 002 065 00903           LXI     D,TMPBLK                to here
047.153  303 044 056 00904           JMP     MOVEM                   move 'em!
                     00905   *
                     00906   *       RESCDB - restore command block
                     00907   *
047.156  006 006     00908   RESCDB  MVI     B,C0.LEN                block length
047.160  041 002 065 00909           LXI     H,TMPBLK                from here
047.163  021 374 064 00910           LXI     D,CDRBLK        to here
047.166  303 044 056 00911           JMP     MOVEM                   move 'em!
                     00912   *
                     00913   *       Process read (and handle error)
```

```
                    00914   *
047.171  315 235 047 00915   PREAD   CALL    DOREA                   Read
047.174  320         00916           RNC                             if no error, return
                    00917   *
                    00918   *       error handling
                    00919   *
047.175  315 143 047 00920           CALL    SAVCDB                  save command block
047.200  315 321 047 00921           CALL    DORSER                  request sense after read
047.203  322 230 047 00922           JNC     RTN66
047.206  315 074 047 00923           CALL    DOREC                   recalibrate
047.211  315 156 047 00924           CALL    RESCDB                  restore command block
047.214  315 256 051 00925           CALL    DOCMD                   re-try the command
047.217  322 230 047 00926           JNC     RTN66
047.222  315 321 047 00927           CALL    DORSER
047.225  332 331 050 00928           JC      ERRIE                   Error - Internal error
047.230  315 156 047 00929   RTN66   CALL    RESCDB                  restore  command block
047.233  067         00930           STC                             indicate error
047.234  311         00931           RET
                    00932   *
                    00933   *       DOREA - read
                    00934   *
047.235  076 010     00935   DOREA   MVI     A,D.REA                 Read
047.237  062 374 064 00936           STA     CDRBLK+C0.OPC           opcode = read
047.242  303 256 051 00937           JMP     DOCMD
                    00938   *
                    00939   *       Process write (and handle error)
                    00940   *
047.245  315 311 047 00941   PWRITE  CALL    DOWRI
047.250  320         00942           RNC
047.251  315 143 047 00943           CALL    SAVCDB                  save 6 byte command block
047.254  315 357 047 00944           CALL    DORSEW                  request sense after write
047.257  322 304 047 00945           JNC     RTN69
047.262  315 074 047 00946           CALL    DOREC                   recalibrate
047.265  315 156 047 00947           CALL    RESCDB                  restore 6 byte command block
047.270  315 256 051 00948           CALL    DOCMD                   re-try the command
047.273  322 304 047 00949           JNC     RTN69
047.276  315 357 047 00950           CALL    DORSEW
047.301  332 331 050 00951           JC      ERRIE                   Error - Internal error
047.304  315 156 047 00952   RTN69   CALL    RESCDB
047.307  067         00953           STC
047.310  311         00954           RET
                    00955   *
                    00956   *       DOWRI - write
                    00957   *
047.311  076 012     00958   DOWRI   MVI     A,D.WRI                 Write
047.313  062 374 064 00959           STA     CDRBLK+C0.OPC           op code = write
047.316  303 256 051 00960           JMP     DOCMD
                    00961   *
                    00962   *       Process request sense after error (read)
                    00963   *
047.321  315 114 047 00964   DORSER  CALL    DD.RSE                  process request sense
047.324  072 052 065 00965           LDA     DSENSE
047.327  346 177     00966           ANI     01111111B               clear Block Address Valid bit
047.331  107         00967           MOV     B,A
047.332  376 020     00968           CPI     00010000B               Test error
047.334  330         00969           RC                              none?
047.335  170         00970           MOV     A,B
```

```
047.336  376 027     00971           CPI     00010111B
047.340  322 345 047 00972           JNC     RTN72
047.343  257         00973           XRA     A                       clear flags
047.344  311         00974           RET
                     00975
047.345  170         00976   RTN72   MOV     A,B
047.346  376 030     00977           CPI     00011000B
047.350  310         00978           RZ                              OK
047.351  170         00979           MOV     A,B
047.352  376 031     00980           CPI     00011001B
047.354  310         00981           RZ                              OK
                     00982
047.355  067         00983           STC                             flag a problem
047.356  311         00984           RET
                     00985   *
                     00986   *       Process request sense after error (write)
                     00987   *
047.357  315 114 047 00988   DORSEW  CALL    DD.RSE                  process request sense
047.362  072 052 065 00989           LDA     DSENSE
047.365  346 177     00990           ANI     01111111B
047.367  107         00991           MOV     B,A
047.370  376 020     00992           CPI     00010000B
047.372  310         00993           RZ
047.373  170         00994           MOV     A,B
047.374  376 022     00995           CPI     00010010B
047.376  310         00996           RZ
047.377  170         00997           MOV     A,B
050.000  376 024     00998           CPI     00010100B
050.002  310         00999           RZ
050.003  170         01000           MOV     A,B
050.004  376 025     01001           CPI     00010101B
050.006  310         01002           RZ
050.007  170         01003           MOV     A,B
050.010  376 031     01004           CPI     00011001B
050.012  310         01005           RZ
050.013  067         01006           STC
050.014  311         01007           RET
                     01008
                     01009   *
                     01010   *       CMPSEC - compare two sectors.  if mismatch mark as bad.
                     01011   *
050.015  006 000     01012   CMPSEC  MVI     B,0                     256 bytes
050.017  041 264 135 01013           LXI     H,TMPBUF
050.022  021 264 065 01014           LXI     D,WRKBUF
050.025  032         01015   CMPSC1  LDAX    D
050.026  276         01016           CMP     M
050.027  023         01017           INX     D
050.030  043         01018           INX     H
050.031  302 041 050 01019           JNZ     CMPSC2                  mismatch!
050.034  005         01020           DCR     B                       decrement counter
050.035  302 025 050 01021           JNZ     CMPSC1                  and loop 'til done
050.040  311         01022           RET
                     01023   *
                     01024   *       Mismatch, mark as bad
                     01025   *
050.041  315 056 050 01026   CMPSC2  CALL    HAVEBS                  bad sector
050.044  311         01027           RET
```

```
                    01028
                    01029
                    01030   *
                    01031   *       FILBYT - Fills memory with contents of B
                    01032   *
                    01033   *       Entry:  (HL) = destination
                    01034   *               B = fill byte
                    01035   *               DE = count
                    01036   *
050.045  160        01037   FILBYT  MOV     M,B                     Store it
050.046  043        01038           INX     H                       point to next storage location
050.047  033        01039           DCX     D                       decrement counter
050.050  172        01040           MOV     A,D
050.051  263        01041           ORA     E
050.052  302 045 050 01042           JNZ     FILBYT                  loop 'til done
050.055  311        01043           RET
                    01044
                    01045
                    01046   *
                    01047   *       HAVEBS - process bad sector.  Bad sectors in the first
                    01048   *               track are considered a fatal error, otherwise mark it
                    01049   *               in bad sector table.  If too many errors abort.
                    01050   *
050.056  052 360 056 01051   HAVEBS  LHLD    CURSEC                  bad sector no.
050.061  174        01052           MOV     A,H
050.062  247        01053           ANA     A
050.063  302 074 050 01054           JNZ     HAVBS1                  > 256? (OK, flag it)
050.066  175        01055           MOV     A,L
050.067  376 050    01056           CPI     SPTRK                   error on first track?
050.071  332 057 051 01057           JC      ERRTK0                  that's considered fatal...
                    01058
050.074  052 362 056 01059   HAVBS1  LHLD    BSCOUNT                 bad sector count?
050.077  043        01060           INX     H                       increment it
050.100  042 362 056 01061           SHLD    BSCOUNT                 and save it
050.103  021 253 000 01062           LXI     D,MAXBS+1               too many for bad sector table to hold?
050.106  315 055 056 01063           CALL    HLCPDE
050.111  322 145 051 01064           JNC     ERRTBS                  Error, too many bad sectors
                    01065   *
                    01066   *       Add entry to Bad Sector Table
                    01067   *
                    01068   *
050.114  052 360 056 01069           LHLD    CURSEC
050.117  353        01070           XCHG                            DE = current (bad) sector
050.120  052 364 056 01071           LHLD    BSTPTR                  HL = next BST entry
050.123  163        01072           MOV     M,E                     low byte
050.124  043        01073           INX     H
050.125  162        01074           MOV     M,D                     mid byte
050.126  043        01075           INX     H
050.127  257        01076           XRA     A                       (hi byte always 0)
050.130  167        01077           MOV     M,A
050.131  043        01078           INX     H                       point to next
050.132  042 364 056 01079           SHLD    BSTPTR                  and save it
                    01080
050.135  311        01081           RET
                    01082   *
                    01083   *       ERRWP - Error - drive is Write Protected
                    01084   *
```

```
050.136  315 130 047 01085   ERRWP   CALL    BEEEEP
050.141  315 136 031 01086           CALL    $TYPTX
050.144  012 124 150 01087           DB      LF,'The Wincherster disk is write protected.'
050.215  012 124 145 01088           DB      LF,'Test aborted.',LF,LF+200Q
                     01089
050.235  303 052 046 01090           JMP     EXIT
                     01091   *
                     01092   *       ERRFMT - Error during formatting
                     01093   *
050.240  315 136 031 01094   ERRFMT  CALL    $TYPTX
050.243  012 105 162 01095           DB      LF,'Error during formatting the drive.'
050.306  012 124 145 01096           DB      LF,'Test aborted.'
050.324  012 212     01097           DB      LF,LF+200Q
050.326  303 052 046 01098           JMP     EXIT
                     01099   *
                     01100   *       ERRIE - Error - internal error
                     01101   *
050.331  315 130 047 01102   ERRIE   CALL    BEEEEP
050.334  315 136 031 01103           CALL    $TYPTX
050.337  012 111 156 01104           DB      LF,'Internal error.'
050.357  012 124 145 01105           DB      LF,'Test aborted.'
050.375  012 212     01106           DB      LF,LF+200Q
050.377  303 052 046 01107           JMP     EXIT
                     01108   *
                     01109   *       ERRDNR - Error - drive not ready
                     01110   *
051.002  315 130 047 01111   ERRDNR  CALL    BEEEEP
051.005  315 136 031 01112           CALL    $TYPTX
051.010  012 104 162 01113           DB      LF,'Drive is not ready.'
051.034  012 124 145 01114           DB      LF,'Test aborted.'
051.052  012 212     01115           DB      LF,LF+200Q
051.054  303 052 046 01116           JMP     EXIT
                     01117   *
                     01118   *       ERRTK0 - Error - track 0 bad sector
                     01119   *
051.057  315 130 047 01120   ERRTK0  CALL    BEEEEP
051.062  315 136 031 01121           CALL    $TYPTX
051.065  012 124 162 01122           DB      LF,'Track 0 contains bad sector.'
051.122  012 124 145 01123           DB      LF,'Test aborted.'
051.140  012 212     01124           DB      LF,LF+200Q
051.142  303 052 046 01125           JMP     EXIT
                     01126   *
                     01127   *       ERRTBS - Error - too many bad sectors
                     01128   *
051.145  315 130 047 01129   ERRTBS  CALL    BEEEEP
051.150  315 136 031 01130           CALL    $TYPTX
051.153  012 124 157 01131           DB      LF,'Too many bad sectors in this drive.'
051.217  012 124 145 01132           DB      LF,'Test aborted.'
051.235  012 212     01133           DB      LF,LF+200Q
051.237  303 052 046 01134           JMP     EXIT
                     01135
                     01136
                     01137   *
                     01138   *       RDSTAT - Read status
                     01139   *
051.242  333 171     01140   RDSTAT  IN      BASE+RI.BST             Bus Status
051.244  311         01141           RET
```

```
                        01142   *
                        01143   *       WRCTRL - Write to control bus
                        01144   *
051.245  323 171       01145   WRCTRL  OUT     BASE+RI.CON             Control
051.247  311           01146           RET
                        01147   *
                        01148   *       RDDATA - Read data
                        01149   *
051.250  333 170       01150   RDDATA  IN      BASE+RI.DAT             Data In
051.252  311           01151           RET
                        01152   *
                        01153   *       WRDATA - Write data
                        01154   *
051.253  323 170       01155   WRDATA  OUT     BASE+RI.DAT             Data Out
051.255  311           01156           RET
                        01157
                        01158   ************************************************************************
                        01159   *
                        01160   *       DOCMD - Issue command to the controller
                        01161   *
                        01162   *       The command data structure (CDRBLK) should have
                        01163   *       already been filled out with all key information
                        01164   *
051.256  363           01165   DOCMD   DI                              NO INTERRUPTS
                        01166   *
                        01167   *       First get controller's attention.
                        01168   *
051.257  001 377 377   01169           LXI     B,-1                    loop counter
051.262  315 242 051   01170   DOCMD1  CALL    RDSTAT
051.265  346 010       01171           ANI     BS.BSY                  Busy?
051.267  312 303 051   01172           JZ      DOCMD2                  no...
051.272  315 067 056   01173           CALL    DCRBC                   count down ...
051.275  302 262 051   01174           JNZ     DOCMD1                  keep trying
051.300  303 002 051   01175           JMP     ERRDNR                  Error - Drive not ready
                        01176   *
                        01177   *       Next assert SELect and DATA0
                        01178   *
051.303  076 100       01179   DOCMD2  MVI     A,BC.SEL                Assert select
051.305  315 245 051   01180           CALL    WRCTRL                  to control register
                        01181   *
                        01182   *       Now wait for controller to respond with BUSY...
                        01183   *
051.310  001 377 377   01184           LXI     B,-1                    loop counter
051.313  315 242 051   01185   DOCMD3  CALL    RDSTAT
051.316  346 010       01186           ANI     BS.BSY                  Busy?
051.320  302 355 051   01187           JNZ     DOCMD5                  yes...
051.323  315 067 056   01188           CALL    DCRBC                   count down ...
051.326  302 313 051   01189           JNZ     DOCMD3                  keep trying
                        01190   *
                        01191   *       timed out! try one more time
                        01192   *
051.331  001 377 377   01193           LXI     B,-1                    loop counter
051.334  315 242 051   01194   DOCMD4  CALL    RDSTAT
051.337  346 010       01195           ANI     BS.BSY                  Busy?
051.341  302 355 051   01196           JNZ     DOCMD5                  yes...
051.344  315 067 056   01197           CALL    DCRBC                   count down ...
051.347  302 334 051   01198           JNZ     DOCMD4                  keep trying
```

```
051.352  303 002 051 01199          JMP     ERRDNR                  Failed twice - Drive not ready
                     01200   *
                     01201   *       controller now has control of the buss
                     01202   *
051.355  076 002     01203   DOCMD5  MVI     A,BC.EDT                Assert enable data
051.357  315 245 051 01204          CALL    WRCTRL                  to control register
                     01205   *
                     01206   *       now we're ready to output the command to the controller
                     01207   *       using REQ/ACK handshaking
                     01208   *
051.362  041 374 064 01209          LXI     H,CDRBLK                point to command block
051.365  315 242 051 01210   DOCMD6  CALL    RDSTAT                  read status
051.370  117         01211          MOV     C,A                     save it ...
051.371  267         01212          ORA     A
051.372  362 365 051 01213          JP      DOCMD6                  loop 'til REQ
051.375  346 020     01214          ANI     BS.COM                  command in progress?
051.377  312 214 053 01215          JZ      RTN118                  no
052.002  171         01216          MOV     A,C                     yes, get back saved byte
052.003  346 100     01217          ANI     BS.DTD                  see if controller switched direction
052.005  312 020 052 01218          JZ      DOCMD7                  is sending data, jump ...
052.010  176         01219          MOV     A,M                     get next byte of command
052.011  315 253 051 01220          CALL    WRDATA                  write byte to controller
052.014  043         01221          INX     H                       point to next
052.015  303 365 051 01222          JMP     DOCMD6                  and loop
                     01223   *
                     01224   *       done! - read completion status
                     01225   *
052.020  315 242 051 01226   DOCMD7  CALL    RDSTAT                  Looking for last REQ
052.023  346 320     01227          ANI     BS.REQ+BS.DTD+BS.COM
052.025  376 220     01228          CPI     BS.REQ+BS.COM
052.027  302 020 052 01229          JNZ     DOCMD7                  Wait for REQ and COM
                     01230
052.032  315 250 051 01231          CALL    RDDATA                  Input completion status
052.035  117         01232          MOV     C,A                     save it..
                     01233
052.036  315 242 051 01234   DOCMD8  CALL    RDSTAT                  read status
052.041  107         01235          MOV     B,A
052.042  346 360     01236          ANI     BS.REQ+BS.DTD+BS.COM+BS.LMB
052.044  376 260     01237          CPI     BS.REQ+BS.COM+BS.LMB    last message byte?
052.046  302 036 052 01238          JNZ     DOCMD8                  no, keep looping
                     01239
052.051  315 250 051 01240          CALL    RDDATA                  input last byte
052.054  127         01241          MOV     D,A                     save copy
                     01242
052.055  373         01243          EI                              INTERRUPTS ON
                     01244
052.056  171         01245          MOV     A,C                     Look at completion status
052.057  346 002     01246          ANI     ST.ERR                  error?
052.061  312 073 052 01247          JZ      DOCMD9
                     01248
052.064  315 244 052 01249          CALL    CHKRSE                  if Request Sense check error count
052.067  303 127 052 01250          JMP     DOCMDD                  Return with error
                     01251
052.072  171         01252          MOV     A,C
                     01253
                     01254   *       have error
                     01255   *
```

```
052.073  346 001      01256   DOCMD9  ANI     ST.PER                  is it a parity error?
052.075  312 103 052  01257           JZ      DOCMDA                  no
                       01258
052.100  315 364 052  01259           CALL    CHKPER                  parity error
                       01260   *
                       01261   *       no parity error
                       01262   *
052.103  172          01263   DOCMDA  MOV     A,D                     get the last data byte
052.104  267          01264           ORA     A                       should be NUL
052.105  312 113 052  01265           JZ      DOCMDB                  Good, jump
052.110  315 131 052  01266           CALL    CHKNUL
                       01267
052.113  170          01268   DOCMDB  MOV     A,B                     last buss status
052.114  346 002      01269           ANI     ST.ERR                  error?
052.116  312 124 052  01270           JZ      DOCMDC                  not set, done
052.121  315 067 053  01271           CALL    CHKERR                  host adapter parity error
                       01272   *
                       01273   *       Normal return ('C' clear)
                       01274   *
052.124  067          01275   DOCMDC  STC
052.125  077          01276           CMC
052.126  311          01277           RET
                       01278   *
                       01279   *       Error return ('C' set)
                       01280   *
052.127  067          01281   DOCMDD  STC
052.130  311          01282           RET
                       01283   *
                       01284   *       completion byte not NUL
                       01285   *
052.131  072 352 056  01286   CHKNUL  LDA     ERRCT1                  error count
052.134  074          01287           INR     A                       increment it
052.135  062 352 056  01288           STA     ERRCT1                  and save
052.140  376 002      01289           CPI     2                       max error?
052.142  322 156 052  01290           JNC     ERRCC                   max, abort! Error - completion code non-zero
052.145  315 200 053  01291           CALL    RETRY                   recalibrate
052.150  076 000      01292           MVI     A,0
052.152  062 352 056  01293           STA     ERRCT1                  reset error count
052.155  311          01294           RET
                       01295   *
                       01296   *       ERRCC - Error - Completion Code non zero
                       01297   *
052.156  315 130 047  01298   ERRCC   CALL    BEEEEP
052.161  315 136 031  01299           CALL    $TYPTX
052.164  012 103 157  01300           DB      LF,'Completion byte is non-zero.'
052.221  012 124 145  01301           DB      LF,'Test aborted.'
052.237  012 212      01302           DB      LF,LF+200Q
052.241  303 052 046  01303           JMP     EXIT
                       01304   *
                       01305   *       Count how many times Request Senese is called
                       01306   *
052.244  072 374 064  01307   CHKRSE  LDA     CDRBLK                  first byte of command descriptor
052.247  376 003      01308           CPI     D.RSE                   Request sense command?
052.251  300          01309           RNZ                             no, exit
                       01310
052.252  072 353 056  01311           LDA     ERRCT2                  error count?
052.255  074          01312           INR     A                       increment it and store
```

```
052.256  062 353 056 01313           STA     ERRCT2
052.261  376 002     01314           CPI     2
052.263  312 277 052 01315           JZ      ERRSSE                  max, abort! Error - Sense status
052.266  315 200 053 01316           CALL    RETRY                   recalibrate
052.271  076 000     01317           MVI     A,0
052.273  062 353 056 01318           STA     ERRCT2                  reset error count
052.276  311         01319           RET
                     01320   *
                     01321   *       ERRSSE - Error - Sense Status error
                     01322   *
052.277  315 130 047 01323   ERRSSE  CALL    BEEEEP
052.302  315 136 031 01324           CALL    $TYPTX
052.305  012 122 145 01325           DB      LF,'Request sense status error.'
052.341  012 124 145 01326           DB      LF,'Test aborted.'
052.357  012 212     01327           DB      LF,LF+200Q
052.361  303 052 046 01328           JMP     EXIT
                     01329   *
                     01330   *       Count how many times parity error occurs
                     01331   *
052.364  072 351 056 01332   CHKPER  LDA     ERRCT0                  get error count
052.367  074         01333           INR     A                       increment
052.370  062 351 056 01334           STA     ERRCT0                  save
052.373  376 002     01335           CPI     2                       compare to max
052.375  312 011 053 01336           JZ      ERRPE                   at max, abort! Error - status parity
053.000  315 200 053 01337           CALL    RETRY                   else recalibrate
053.003  076 000     01338           MVI     A,0
053.005  062 351 056 01339           STA     ERRCT0                  reset error count
053.010  311         01340           RET
                     01341   *
                     01342   *       ERRPE - Error - Status parity error
                     01343   *
053.011  315 130 047 01344   ERRPE   CALL    BEEEEP
053.014  315 136 031 01345           CALL    $TYPTX
053.017  012 123 164 01346           DB      LF,'Status parity error.'
053.044  012 124 145 01347           DB      LF,'Test aborted.'
053.062  012 212     01348           DB      LF,LF+200Q
053.064  303 052 046 01349           JMP     EXIT
                     01350   *
                     01351   *       Count how many times host adapter parity error occurs
                     01352   *
053.067  072 354 056 01353   CHKERR  LDA     ERRCT3                  get error count
053.072  074         01354           INR     A                       increment
053.073  062 354 056 01355           STA     ERRCT3                  and save
053.076  376 002     01356           CPI     2                       at max?
053.100  312 114 053 01357           JZ      ERRHAP                  yes, abort! Error - Host adapter parity
053.103  315 200 053 01358           CALL    RETRY                   else recalibrate and re-issue
053.106  076 000     01359           MVI     A,0
053.110  062 354 056 01360           STA     ERRCT3                  reset error
053.113  311         01361           RET
                     01362   *
                     01363   *       ERRHAP - Error - Host Adapter parity
                     01364   *
053.114  315 130 047 01365   ERRHAP  CALL    BEEEEP
053.117  315 136 031 01366           CALL    $TYPTX
053.122  012 110 157 01367           DB      LF,'Host adapter parity error.'
053.155  012 124 145 01368           DB      LF,'Test aborted.'
053.173  012 212     01369           DB      LF,LF+200Q
```

```
053.175  303 052 046 01370           JMP     EXIT
                     01371   *
                     01372   *       Recalibrate and re-execute
                     01373   *
                     01374   *       Save command block, recalibrate, then restore and execute command
                     01375   *
053.200  315 143 047 01376   RETRY   CALL    SAVCDB                  Save the command block
053.203  315 074 047 01377           CALL    DOREC                   recalibrate...
053.206  315 156 047 01378           CALL    RESCDB                  Restore the command byte
053.211  303 256 051 01379           JMP     DOCMD                   and re-execute
                     01380
                     01381
                     01382   *       do request sense?
                     01383
053.214  041 264 135 01384   RTN118  LXI     H,TMPBUF
053.217  072 311 056 01385           LDA     BASE67                  get base port
053.222  117         01386           MOV     C,A
053.223  006 000     01387           MVI     B,0                     BC = base port
053.225  072 374 064 01388           LDA     CDRBLK                  first byte of command descriptor
053.230  376 010     01389           CPI     D.REA                   read?
053.232  312 255 053 01390           JZ      RTN120
053.235  376 012     01391           CPI     D.WRI                   write?
053.237  302 250 053 01392           JNZ     RTN119                  not read or write
053.242  052 060 065 01393           LHLD    UNK191
053.245  303 255 053 01394           JMP     RTN120
                     01395
053.250  006 004     01396   RTN119  MVI     B,4
053.252  041 052 065 01397           LXI     H,DSENSE
                     01398
053.255  315 242 051 01399   RTN120  CALL    RDSTAT
053.260  127         01400           MOV     D,A
053.261  346 200     01401           ANI     10000000B
053.263  312 255 053 01402           JZ      RTN120
053.266  172         01403           MOV     A,D
053.267  346 020     01404           ANI     00010000B
053.271  302 020 052 01405           JNZ     DOCMD7
053.274  172         01406           MOV     A,D
053.275  346 100     01407           ANI     01000000B
053.277  312 307 053 01408           JZ      RTN121
                     01409   *       Z80 "OTIR"      output from memory to port (C); loop 'til B=0
053.302  355 263     01410           DW      MI.OTIR
053.304  303 255 053 01411           JMP     RTN120
                     01412
                     01413   *       Z80 "INIR"      input to memory from port (C); loop 'til B=0*
053.307  355 262     01414   RTN121  DW      MI.INIR
053.311  303 255 053 01415           JMP     RTN120
                     01416   *
                     01417   *       RESET - reset the controller
                     01418   *
053.314  076 020     01419   RESET   MVI     A,BC.RST                reset
053.316  315 245 051 01420           CALL    WRCTRL
053.321  311         01421           RET
                     01422   *
                     01423   *       some kind of processing of bad sector table
                     01424   *       create second copy? if needed?
                     01425   *
053.322  052 362 056 01426   RTN123  LHLD    BSCOUNT
```

```
053.325  042 330 056 01427           SHLD    UNK165
053.330  001 371 056 01428           LXI     B,BUFF6                 buffer for bad sector table
053.333  021 120 000 01429           LXI     D,80
                     01430
053.336  041 314 056 01431           LXI     H,UNK159
053.341  042 312 056 01432           SHLD    UNK158
053.344  076 011     01433           MVI     A,9
053.346  066 000     01434   RTN124  MVI     M,0                     fill 9 bytes with 0
053.350  043         01435           INX     H
053.351  075         01436           DCR     A
053.352  302 346 053 01437           JNZ     RTN124
                     01438
053.355  066 377     01439           MVI     M,377Q                  -1
053.357  062 327 056 01440           STA     UNK164
053.362  257         01441   RTN125  XRA     A
053.363  062 326 056 01442           STA     UNK163
053.366  315 056 054 01443   RTN126  CALL    RTN130
053.371  315 055 056 01444           CALL    HLCPDE
053.374  322 007 054 01445           JNC     RTN127
053.377  076 001     01446           MVI     A,1
054.001  062 326 056 01447           STA     UNK163
054.004  303 366 053 01448           JMP     RTN126
                     01449
054.007  072 326 056 01450   RTN127  LDA     UNK163
054.012  247         01451           ANA     A
054.013  302 024 054 01452           JNZ     RTN128
054.016  315 113 054 01453           CALL    RTN132
054.021  176         01454           MOV     A,M
054.022  074         01455           INR     A
054.023  310         01456           RZ
                     01457
054.024  072 327 056 01458   RTN128  LDA     UNK164
054.027  247         01459           ANA     A
054.030  302 045 054 01460           JNZ     RTN129
054.033  013         01461           DCX     B
054.034  013         01462           DCX     B
054.035  013         01463           DCX     B
054.036  052 330 056 01464           LHLD    UNK165          inc bscount
054.041  043         01465           INX     H
054.042  042 330 056 01466           SHLD    UNK165
054.045  056 050     01467   RTN129  MVI     L,40
054.047  046 000     01468           MVI     H,0
054.051  031         01469           DAD     D               HL = DE + 40
054.052  353         01470           XCHG
054.053  303 362 053 01471           JMP     RTN125
                     01472
054.056  052 330 056 01473   RTN130  LHLD    UNK165
054.061  175         01474           MOV     A,L
054.062  264         01475           ORA     H               HL == 0?
054.063  302 077 054 01476           JNZ     RTN131          no, load HL from BC
054.066  076 001     01477           MVI     A,1
054.070  062 327 056 01478           STA     UNK164
054.073  041 377 377 01479           LXI     H,377377A       -1
054.076  311         01480           RET
                     01481   *
                     01482   *       Load HL from triple pointed to by BC
                     01483   *       (third byte is ignored)
```

```
                    01484   *
054.077  053       01485   RTN131  DCX     H
054.100  042 330 056 01486           SHLD    UNK165
054.103  012       01487           LDAX    B
054.104  157       01488           MOV     L,A
054.105  003       01489           INX     B
054.106  012       01490           LDAX    B
054.107  147       01491           MOV     H,A
054.110  003       01492           INX     B
054.111  003       01493           INX     B
054.112  311       01494           RET
                    01495
                    01496
054.113  325       01497   RTN132  PUSH    D
054.114  076 050   01498           MVI     A,40
054.116  057       01499           CMA
054.117  157       01500           MOV     L,A
054.120  046 377   01501           MVI     H,377Q
054.122  043       01502           INX     H               HL = -40
054.123  353       01503           XCHG
054.124  031       01504           DAD     D
054.125  353       01505           XCHG
054.126  052 312 056 01506         LHLD    UNK158
054.131  257       01507           XRA     A
054.132  167       01508           MOV     M,A             zero high byte
054.133  043       01509           INX     H
054.134  162       01510           MOV     M,D             mid byte
054.135  043       01511           INX     H
054.136  163       01512           MOV     M,E             low byte
054.137  043       01513           INX     H
054.140  042 312 056 01514         SHLD    UNK158
054.143  321       01515           POP     D
054.144  311       01516           RET
                    01517   *
                    01518   *       zero out key tables on the disk
                    01519   *
054.145  041 264 065 01520   ZTBLS   LXI     H,WRKBUF                buffer
054.150  021 000 050 01521           LXI     D,SPTRK*SECSIZ          40 sectors (10K!)
054.153  006 000     01522           MVI     B,0                     fill with zero
054.155  315 045 050 01523           CALL    FILBYT                  fill it!
                    01524
054.160  041 264 065 01525           LXI     H,WRKBUF
054.163  042 060 065 01526           SHLD    UNK191
054.166  041 000 000 01527           LXI     H,0                     boot/SBC
054.171  315 217 054 01528           CALL    WTRACK                  write the track
054.174  052 315 056 01529           LHLD    SUPERA
054.177  315 217 054 01530           CALL    WTRACK
054.202  052 320 056 01531           LHLD    SUPERB
054.205  315 217 054 01532           CALL    WTRACK
054.210  052 323 056 01533           LHLD    BADSTB
054.213  315 217 054 01534           CALL    WTRACK
054.216  311       01535           RET
                    01536   *
                    01537   *       WTRACK - write a track (40 sectors)
                    01538   *
                    01539   *       ENTRY:  HL=block address to write to
                    01540   *
```

```
054.217  076 050      01541   WTRACK  MVI     A,SPTRK                 40 sectors
054.221  062 000 065  01542           STA     CDRBLK+C0.NB
054.224  042 376 064  01543           SHLD    CDRBLK+C0.AD1           address
054.227  315 245 047  01544           CALL    PWRITE                  write the track
054.232  311          01545           RET
                      01546
                      01547
                      01548   *
                      01549   *       GENBST - generate and write Bad Sector Table(s)
                      01550   *
054.233  052 362 056  01551   GENBST  LHLD    BSCOUNT
054.236  042 056 065  01552           SHLD    UNK190
054.241  174          01553           MOV     A,H
054.242  265          01554           ORA     L                       no bad sectors?
054.243  310          01555           RZ                              great, we're done!
                      01556
054.244  175          01557           MOV     A,L                     store as triple
054.245  062 366 056  01558           STA     UNK177
054.250  174          01559           MOV     A,H
054.251  062 367 056  01560           STA     UNK177+1
054.254  257          01561           XRA     A
054.255  062 370 056  01562           STA     UNK177+2
                      01563
054.260  043          01564           INX     H                       HL=c+1
054.261  345          01565           PUSH    H
054.262  321          01566           POP     D                       DE=c+1
054.263  031          01567           DAD     D                       *2
054.264  031          01568           DAD     D                       *4
054.265  042 056 065  01569           SHLD    UNK190
054.270  175          01570           MOV     A,L
054.271  247          01571           ANA     A
054.272  174          01572           MOV     A,H                     A = n blocks
054.273  312 277 054  01573           JZ      GNBST1
054.276  074          01574           INR     A                       bump up if L!=0
054.277  062 000 065  01575   GNBST1  STA     CDRBLK+C0.NB            # blocks
054.302  052 356 056  01576           LHLD    BADSTA
054.305  042 376 064  01577           SHLD    CDRBLK+C0.AD1
054.310  041 366 056  01578   GNBST2  LXI     H,UNK177
054.313  042 060 065  01579           SHLD    UNK191
054.316  315 245 047  01580           CALL    PWRITE
054.321  052 376 064  01581           LHLD    CDRBLK+C0.AD1
054.324  353          01582           XCHG
054.325  052 323 056  01583           LHLD    BADSTB
054.330  315 055 056  01584           CALL    HLCPDE                  BADSTB=BADSTA?
054.333  310          01585           RZ
054.334  042 376 064  01586           SHLD    CDRBLK+C0.AD1
054.337  303 310 054  01587           JMP     GNBST2
                      01588
                      01589
                      01590   *
                      01591   *       GENSBK - generate (and write) the Superblock
                      01592   *
054.342  041 264 065  01593   GENSBK  LXI     H,WRKBUF                First blank the OS ID table
054.345  021 000 001  01594           LXI     D,SECSIZ                        one sector
054.350  006 040      01595           MVI     B,' '
054.352  315 045 050  01596           CALL    FILBYT                  fill with blanks
                      01597
```

```
054.355  041 264 066 01598           LXI     H,SATBUF                Now blank the sector allocation table
054.360  021 000 001 01599           LXI     D,SECSIZ                        one sector
054.363  006 000     01600           MVI     B,0
054.365  315 045 050 01601           CALL    FILBYT                  fill with NUL
                     01602   *
                     01603   *       Initially SAT has only 3 entries:
                     01604   *
                     01605   *       0   0   0   0
                     01606   *       UA  0   0   0
                     01607   *       ET  TOTSEC  0
                     01608   *
054.370  076 036     01609           MVI     A,SAT.UA                unallocated
054.372  062 270 066 01610           STA     SATBUF+4
054.375  076 360     01611           MVI     A,SPTRK*TPCYL
054.377  062 271 066 01612           STA     SATBUF+5
055.002  076 037     01613           MVI     A,SAT.ET                end of table
055.004  062 274 066 01614           STA     SATBUF+8
055.007  041 300 344 01615           LXI     H,TOTSEC
055.012  042 275 066 01616           SHLD    SATBUF+9
                     01617
055.015  041 264 065 01618           LXI     H,WRKBUF
055.020  042 060 065 01619           SHLD    UNK191
                     01620   *
                     01621   *       write superblocks
                     01622   *
055.023  076 002     01623           MVI     A,2                     two blocks each
055.025  062 000 065 01624           STA     CDRBLK+C0.NB
055.030  052 315 056 01625           LHLD    SUPERA
055.033  042 376 064 01626           SHLD    CDRBLK+C0.AD1           address of buffer
055.036  315 245 047 01627           CALL    PWRITE                  write Superblock A
                     01628
055.041  052 320 056 01629           LHLD    SUPERB
055.044  042 376 064 01630           SHLD    CDRBLK+C0.AD1
055.047  315 245 047 01631           CALL    PWRITE                  write Superblock B
                     01632
055.052  311         01633           RET
                     01634
                     01635   *
                     01636   *       Update and write out the Software Boot Code block
                     01637   *
055.053  041 000 001 01638   WRTSBC  LXI     H,SECSIZ                Sector size
055.056  042 126 065 01639           SHLD    S.SSZ
055.061  041 050 000 01640           LXI     H,SPTRK                 Sectors per track
055.064  042 130 065 01641           SHLD    S.SPT
055.067  041 006 000 01642           LXI     H,TPCYL
055.072  042 132 065 01643           SHLD    S.TPC                   Tracks/cylinder
055.075  041 364 000 01644           LXI     H,TOTCYL
055.100  042 134 065 01645           SHLD    S.CPV                   Cylinders/volume
055.103  041 360 000 01646           LXI     H,SPCYL
055.106  042 136 065 01647           SHLD    S.SPC                   Sectors per cylinder (region)
055.111  041 300 344 01648           LXI     H,TOTSEC
055.114  042 140 065 01649           SHLD    S.STOT                  Total number of sectors
                     01650
055.117  257         01651           XRA     A                       zero third bytes
055.120  062 142 065 01652           STA     S.STOT+2
055.123  062 117 065 01653           STA     S.BSTB+2
055.126  062 122 065 01654           STA     S.SBA+2
```

```
055.131  062 125 065 01655           STA     S.SBB+2
                     01656
055.134  052 323 056 01657           LHLD    BADSTB                  Store Bad Sector Table B sector
055.137  315 040 056 01658           CALL    HLSWAP
055.142  042 115 065 01659           SHLD    S.BSTB                  Bad sector table B
                     01660
055.145  052 315 056 01661           LHLD    SUPERA                  Store superblock A sector
055.150  315 040 056 01662           CALL    HLSWAP
055.153  042 120 065 01663           SHLD    S.SBA                   Superblock A
                     01664
055.156  052 320 056 01665           LHLD    SUPERB                  Store superblock B sector
055.161  315 040 056 01666           CALL    HLSWAP
055.164  042 123 065 01667           SHLD    S.SBB                   Superblock B
                     01668
055.167  052 315 056 01669           LHLD    SUPERA
055.172  001 000 003 01670           LXI     B,3*SECSIZ                      superblocks are 3 sectors
055.175  076 003     01671           MVI     A,3                     3 sectors
055.177  315 320 055 01672           CALL    GETCRC
055.202  042 144 065 01673           SHLD    S.CSBA                  Checksum for Superblock A
                     01674
055.205  052 320 056 01675           LHLD    SUPERB                  now do superblock B
055.210  001 000 003 01676           LXI     B,3*SECSIZ
055.213  076 003     01677           MVI     A,3
055.215  315 320 055 01678           CALL    GETCRC
055.220  042 146 065 01679           SHLD    S.CSBB                  Checksum for Superblock B
                     01680
055.223  052 323 056 01681           LHLD    BADSTB
055.226  001 000 002 01682           LXI     B,2*SECSIZ                      2 sectors
055.231  076 002     01683           MVI     A,2
055.233  315 320 055 01684           CALL    GETCRC
055.236  042 150 065 01685           SHLD    S.CBSA                  Checksum for bad sector table A
                     01686   *
                     01687   *       Compute checksum for Bad Sector Table B
                     01688   *
055.241  046 024     01689           MVI     H,20                    block 20
055.243  056 000     01690           MVI     L,0
055.245  001 000 002 01691           LXI     B,2*SECSIZ                      512 bytes
055.250  076 002     01692           MVI     A,2                     2 sectors
055.252  315 320 055 01693           CALL    GETCRC
055.255  042 152 065 01694           SHLD    S.CBSB                  Checksum for bad sector table B
                     01695
055.260  021 200 000 01696           LXI     D,128                   fill second half with NUL
055.263  041 262 065 01697           LXI     H,SBCTBL+128
055.266  006 000     01698           MVI     B,0
055.270  315 045 050 01699           CALL    FILBYT
                     01700
055.273  041 000 000 01701           LXI     H,0
055.276  042 376 064 01702           SHLD    CDRBLK+C0.AD1           Write to sector 0
055.301  076 001     01703           MVI     A,1                     one sector
055.303  062 000 065 01704           STA     CDRBLK+C0.NB
055.306  041 062 065 01705           LXI     H,SBCTBL                point to the table
055.311  042 060 065 01706           SHLD    UNK191
055.314  315 245 047 01707           CALL    PWRITE                  and write it!
055.317  311         01708           RET
                     01709
                     01710   *
                     01711   *       GETCRC - read bytes from disk and compute CRC
```

```
                    01712   *
                    01713   *       ENTRY:  HL = block address
                    01714   *               A = number of blocks
                    01715   *               BC = byte count
                    01716
055.320  305       01717   GETCRC  PUSH    B                       rtn140
055.321  042 376 064 01718           SHLD    CDRBLK+C0.AD1
055.324  062 000 065 01719           STA     CDRBLK+C0.NB
055.327  315 171 047 01720           CALL    PREAD
055.332  301       01721           POP     B
055.333  041 264 135 01722           LXI     H,TMPBUF
055.336  021 000 000 01723           LXI     D,0
055.341  315 346 055 01724           CALL    $BCRC
055.344  353       01725           XCHG
055.345  311       01726           RET
                    01727
055.346             01728           XTEXT   BCRC



                    01730X  **      $BCRC - GENERATE CRC16 ON A BLOCK OF DATA.
                    01731X  *
                    01732X  *       *** WARNING ***
                    01733X  *
                    01734X  *       THIS CRC-16 IS NOT COMPATIBLE WITH THE ONE
                    01735X  *       PRODUCED BY PAM-8, AND THE DECK CRC.COM!
                    01736X  *
                    01737X  *       ENTRY   (BC) = BYTE COUNT
                    01738X  *               (HL) = ADDRESS
                    01739X  *               (DE) = CRC ACCUMULATOR
                    01740X  *       EXIT    (HL) = (HL)+(BC)
                    01741X  *               (DE) = NEW CRC
                    01742X  *       USES    ALL
                    01743X
                    01744X
055.346  170       01745X  $BCRC   MOV     A,B
055.347  261       01746X          ORA     C
055.350  310       01747X          RZ                      NO MORE
055.351  176       01748X          MOV     A,M             (A) = NEW BYTE
055.352  345       01749X          PUSH    H
055.353  305       01750X          PUSH    B               SAVE REGISTERS
055.354  253       01751X          XRA     E
055.355  107       01752X          MOV     B,A
055.356  017       01753X          RRC
055.357  017       01754X          RRC
055.360  017       01755X          RRC
055.361  017       01756X          RRC
055.362  117       01757X          MOV     C,A
055.363  250       01758X          XRA     B
055.364  346 360   01759X          ANI     0F0H
055.366  252       01760X          XRA     D
055.367  157       01761X          MOV     L,A
055.370  171       01762X          MOV     A,C
055.371  007       01763X          RLC
055.372  346 037   01764X          ANI     1FH
055.374  255       01765X          XRA     L
```

```
055.375  157          01766X          MOV     L,A
055.376  170          01767X          MOV     A,B
055.377  007          01768X          RLC
056.000  346 001      01769X          ANI     1
056.002  252          01770X          XRA     D
056.003  255          01771X          XRA     L
056.004  127          01772X          MOV     D,A
056.005  171          01773X          MOV     A,C
056.006  346 360      01774X          ANI     0F0H
056.010  250          01775X          XRA     B
056.011  137          01776X          MOV     E,A
056.012  171          01777X          MOV     A,C
056.013  250          01778X          XRA     B
056.014  007          01779X          RLC
056.015  346 340      01780X          ANI     0E0H
056.017  253          01781X          XRA     E
056.020  137          01782X          MOV     E,A
056.021  301          01783X          POP     B
056.022  341          01784X          POP     H
056.023  043          01785X          INX     H
056.024  013          01786X          DCX     B
056.025  303 346 055  01787X          JMP     $BCRC
                      01788   *
                      01789   *       HLCMI - Complement HL and then increment it
                      01790   *
056.030  174          01791   HLCMI   MOV     A,H
056.031  057          01792           CMA
056.032  147          01793           MOV     H,A
056.033  175          01794           MOV     A,L
056.034  057          01795           CMA
056.035  157          01796           MOV     L,A
056.036  043          01797           INX     H
056.037  311          01798           RET
                      01799   *
                      01800   *       HLSWAP
                      01801   *
                      01802   *       L <=> H
                      01803   *
056.040  175          01804   HLSWAP  MOV     A,L
056.041  154          01805           MOV     L,H
056.042  147          01806           MOV     H,A
056.043  311          01807           RET
                      01808   *
                      01809   *       MOVEM - move bytes from source to destination
                      01810   *
                      01811   *       ENTRY:  BC = byte count
                      01812   *               HL = pointer to source
                      01813   *               DE = pointer to destination
                      01814   *
056.044  176          01815   MOVEM   MOV     A,M                     Fetch byte
056.045  022          01816           STAX    D                       store it
056.046  043          01817           INX     H                       next source
056.047  023          01818           INX     D                       next destination
056.050  005          01819           DCR     B                       count down
056.051  302 044 056  01820           JNZ     MOVEM                   loop 'til zero
056.054  311          01821           RET
                      01822
```

```
056.055                 01823           XTEXT   HLCPDE
                        01824X
                        01825X  **      HLCPDE  - (HL) COMPARED TO (DE)
                        01826X  *
                        01827X  *       THIS ROUTINE IS DOUBLE WORD COMPARE OF REGISTER PAIRS (DE) AND (HL).
                        01828X  *
                        01829X  *       ENTRY:  (HL)&(DE) SET UP
                        01830X  *
                        01831X  *       EXIT:   (PSW)   =
                        01832X  *                        'Z' SET   IF (HL) =  (DE)
                        01833X  *                        'C' SET   IF (HL) <  (DE)
                        01834X  *                        'C' CLEAR IF (HL) >= (DE)
                        01835X  *
                        01836X  *
                        01837X  *       USES:   (PSW)
                        01838X  *
                        01839X
056.055  174           01840X  HLCPDE  MOV     A,H
056.056  272           01841X          CMP     D       'C' SET => (A) < (D)
056.057  300           01842X          RNZ
056.060  175           01843X          MOV     A,L
056.061  273           01844X          CMP     E       'C' SET => (L) < (E)
056.062  311           01845X          RET
                        01846
                        01847   *
                        01848   *       STORDE - store DE via HL
                        01849   *
056.063  162           01850   STORDE  MOV     M,D
056.064  043           01851           INX     H
056.065  163           01852           MOV     M,E
056.066  311           01853           RET
                        01854
                        01855   *
                        01856   *       DCRBC - decrement BC and set flags
                        01857   *
                        01858   *       EXIT:   'Z' set if BC = 0
                        01859   *
056.067  013           01860   DCRBC   DCX     B
056.070  170           01861           MOV     A,B
056.071  261           01862           ORA     C
056.072  076 000       01863           MVI     A,0
056.074  311           01864           RET
                        01865
056.075                 01866           XTEXT   RTL
                        01867X



                        01869X  **      $RTL - READ TEXT LINE.
                        01870X  *
                        01871X  *       $RTL READS A LINE FROM THE TERMINAL.
                        01872X  *
                        01873X  *       CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
                        01874X  *       CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
                        01875X  *       $RTL RETURNS.
                        01876X  *
```

```
                    01877X  *       ENTRY   (HL) = BUFFER FWA
                    01878X  *       EXIT    'C' CLEAR IF OK
                    01879X  *                DATA IN BUFFER
                    01880X  *                (A) = TEXT LENGTH
                    01881X  *               'C' SET IF CTL-D STRUCK
                    01882X  *       USES    A,F
                    01883X
                    01884X
056.075  315 104 056 01885X  $RTL.   CALL    $RTL            $RTL IN UPPER CASE
056.100  330         01886X          RC                      CTL-D
056.101  303 142 056 01887X          JMP     $MLU            MAP LINE TO UPPER CASE
                    01888X
056.104              01889X  $RTL    EQU     *
056.104  345         01890X          PUSH    H               SAVE FWA
056.105  315 173 056 01891X  $RTL1   CALL    $RCHAR
056.110  376 004     01892X          CPI     CTLD
056.112  312 137 056 01893X          JE      $RTL2           CTL-D STRUCK
056.115  167         01894X          MOV     M,A
056.116  043         01895X          INX     H
056.117  376 012     01896X          CPI     NL
056.121  302 105 056 01897X          JNE     $RTL1
056.124  053         01898X          DCX     H
056.125  066 000     01899X          MVI     M,0
056.127  043         01900X          INX     H
                    01901X
                    01902X  *       ALL DONE. COMPUTE LENGTH
                    01903X
056.130  353         01904X          XCHG                    (DE) = LWA+1
056.131  343         01905X          XTHL                    (HL) = FWA
056.132  173         01906X          MOV     A,E
056.133  225         01907X          SUB     L               (A) = LENGTH
056.134  247         01908X          ANA     A               CLEAR CARRY
056.135  321         01909X          POP     D               RESTORE (DE)
056.136  311         01910X          RET
                    01911X
                    01912X  *       CTL-D STRUCK
                    01913X
056.137  341         01914X  $RTL2   POP     H               (HL) = FWA
056.140  067         01915X          STC
056.141  311         01916X          RET
056.142              01917           XTEXT   MLU



                    01919X  **      MLU - MAP LOWER CASE LINE TO UPPER CASE.
                    01920X  *
                    01921X  *       MLU MAPS THE LOWER CASE ALPHABETICS IN A LINE TO UPPER CASE.
                    01922X  *
                    01923X  *       ENTRY   (HL)  = LINE FWA
                    01924X  *       EXIT    NONE
                    01925X  *       USES    NONE
                    01926X
                    01927X
056.142  365         01928X  $MLU    PUSH    PSW             SAVE (PSW)
056.143  345         01929X          PUSH    H               SAVE FWA
056.144  053         01930X          DCX     H               ANTICIPATE INX H
```

```
056.145  043          01931X  $MLU1   INX     H
056.146  176          01932X          MOV     A,M             (A)= CHARACTER
056.147  315 162 056  01933X          CALL    $MCU            MAP CHAR TO UPPER
056.152  167          01934X          MOV     M,A
056.153  247          01935X          ANA     A
056.154  302 145 056  01936X          JNZ     $MLU1           MORE TO GO
056.157  341          01937X          POP     H               RESTORE (HL)
056.160  361          01938X          POP     PSW             RESTORE (PSW)
056.161  311          01939X          RET
056.162               01940           XTEXT   MCU
                      01941X


                      01943X  **      MCU - MAP LOWER CASE TO UPPER CASE.
                      01944X  *
                      01945X  *       MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
                      01946X  *       CASE.
                      01947X  *
                      01948X  *       ENTRY   (A) = CHARACTER
                      01949X  *       EXIT    (A) = CHARACTER RESULT
                      01950X  *       USES    A,F
                      01951X
                      01952X
056.162  376 141      01953X  $MCU    CPI     'a'
056.164  330          01954X          RC                      NOT LOWER CASE
056.165  376 173      01955X          CPI     'z'+1
056.167  320          01956X          RNC                     NOT LOWER CASE
056.170  326 040      01957X          SUI     'a'-'A'
056.172  311          01958X          RET
056.173               01959           XTEXT   RCHAR
                      01960X


                      01962X  **      $RCHAR - READ SINGLE CHARACTER FROM CONSOLE.
                      01963X  *
                      01964X  *       ENTRY   NONE
                      01965X  *       EXIT    (A) = CHARACTER
                      01966X  *       USES    A,F
                      01967X
                      01968X
056.173  377 001      01969X  $RCHAR  DB      SYSCALL,.SCIN
056.175  332 173 056  01970X          JC      $RCHAR          NOT READY
056.200  311          01971X          RET
                      01972X
056.201  377 002      01973X  $WCHAR  DB      SYSCALL,.SCOUT
056.203  311          01974X          RET
                      01975
                      01976   *       GETVAL -- Get octal value from string
                      01977   *
                      01978   *       ENTRY:  HL = address of string value
                      01979   *               DE = address of result
                      01980   *
                      01981   *       USES:   B, PSW
```

```
                    01982   *
056.204  176       01983   GETVAL  MOV     A,M             get a byte
056.205  267       01984           ORA     A               set flags
056.206  312 236 056 01985           JZ      GVDONE          if NULL then done!
056.211  326 060   01986           SUI     '0'             remove ASCII offset
056.213  332 236 056 01987           JC      GVDONE          If <'0' then done
056.216  376 010   01988           CPI     8
056.220  322 236 056 01989           JNC     GVDONE          If >'7' then also done
056.223  107       01990           MOV     B,A             keep a copy
056.224  032       01991           LDAX    D               get the working value
056.225  207       01992           ADD     A               *2
056.226  207       01993           ADD     A               *4
056.227  207       01994           ADD     A               *8
056.230  200       01995           ADD     B               + digit
056.231  022       01996           STAX    D               save it
056.232  043       01997           INX     H               point to next digit
056.233  303 204 056 01998           JMP     GETVAL
056.236  311       01999   GVDONE  RET
                    02000   *
                    02001   *       PROVAL - Print Octal Value
                    02002   *
                    02003   *       ENTRY:  DE = address of byte
                    02004   *
056.237  032       02005   PROVAL  LDAX    D               get byte
                    02006   *
                    02007   *       process first digit
                    02008   *
056.240  007       02009           RLC                     get top 2 bits
056.241  007       02010           RLC                     (high digit)
056.242  107       02011           MOV     B,A             save working copy
056.243  346 003   02012           ANI     00000011B       mask to low 2 bits
056.245  306 060   02013           ADI     '0'             add ASCII offset
056.247  062 304 056 02014           STA     OCTVAL          store first byte in string
                    02015   *
                    02016   *       process second digit
                    02017   *
056.252  170       02018           MOV     A,B             get back working value
056.253  007       02019           RLC                     move the next 3
056.254  007       02020           RLC                     bits into the low 3
056.255  007       02021           RLC
056.256  107       02022           MOV     B,A             save this copy
056.257  346 007   02023           ANI     00000111B       mask to low digit
056.261  306 060   02024           ADI     '0'             add ASCII offset
056.263  062 305 056 02025           STA     OCTVAL+1        store second byte in string
                    02026   *
                    02027   *       process third digit
                    02028   *
056.266  170       02029           MOV     A,B             get working value
056.267  007       02030           RLC                     move the last 3 bits
056.270  007       02031           RLC                     into the low 3
056.271  007       02032           RLC
056.272  346 007   02033           ANI     00000111B       mask to low digit
056.274  306 060   02034           ADI     '0'             add ASCII offset
056.276  062 306 056 02035           STA     OCTVAL+2        store third byte in string
                    02036   *
                    02037   *       print the string
                    02038   *
```

```
056.301  315 136 031 02039           CALL    $TYPTX
056.304  060 060 060 02040   OCTVAL  DB      '000',200Q
                     02041
056.310  311         02042           RET                     done!
                     02043   *
                     02044   *       Data and buffers here ...
                     02045   *
056.311  000         02046   BASE67  DB      0                       Base port (normally 170Q)
056.312  314 056     02047   UNK158  DW      UNK159
056.314  000         02048   UNK159  DB      0
056.315  000 000 000 02049   SUPERA  DB      0,0,0                   location of superblock A
056.320  000 000 000 02050   SUPERB  DB      0,0,0
056.323  000 000 377 02051   BADSTB  DB      0,0,377Q                ??
056.326  000         02052   UNK163  DB      0
056.327  000         02053   UNK164  DB      0
056.330  000 000     02054   UNK165  DW      0
                     02055   *
                     02056   *       Text Buffer for user q&a
                     02057   *
056.332  000 000 000 02058   RTBUFF  DB      0,0,0,0,0,0,0,0,0,0
056.344  131 105 123 02059   ASCYES  DB      'YES'
056.347  120         02060   ASCIIP  DB      'P'
056.350  000         02061           DB      0
                     02062   *
                     02063   *       error cunters
                     02064   *
056.351  000         02065   ERRCT0  DB      0                       parity error
056.352  000         02066   ERRCT1  DB      0                       non zero completion byte
056.353  000         02067   ERRCT2  DB      0                       request sense error
056.354  000 000     02068   ERRCT3  DW      0                       host adapter parity
                     02069
056.356  000 024     02070   BADSTA  DB      0,20                    BADSTA is always at block 20
                     02071
056.360  000 000     02072   CURSEC  DW      0
056.362  000 000     02073   BSCOUNT DW      0                       bad sector count
056.364  000 000     02074   BSTPTR  DW      0                       next entry in BST
056.366  000 000 000 02075   UNK177  DB      0,0,0
                     02076   *
                     02077   *       6 sector fill area
                     02078   *
                     02079   *       1538 space buffer (6*256) +2
                     02080   *
056.371  000 000 000 02081   BUFF6   DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
057.031  000 000 000 02082           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
057.071  000 000 000 02083           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
057.131  000 000 000 02084           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
057.171  000 000 000 02085           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
057.231  000 000 000 02086           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
057.271  000 000 000 02087           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
057.331  000 000 000 02088           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                     02089
057.371  000 000 000 02090           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
060.031  000 000 000 02091           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
060.071  000 000 000 02092           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
060.131  000 000 000 02093           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
060.171  000 000 000 02094           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
060.231  000 000 000 02095           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

```
060.271  000 000 000 02096           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
060.331  000 000 000 02097           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                     02098
060.371  000 000 000 02099           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
061.031  000 000 000 02100           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
061.071  000 000 000 02101           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
061.131  000 000 000 02102           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
061.171  000 000 000 02103           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
061.231  000 000 000 02104           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
061.271  000 000 000 02105           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
061.331  000 000 000 02106           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                     02107
061.371  000 000 000 02108           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
062.031  000 000 000 02109           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
062.071  000 000 000 02110           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
062.131  000 000 000 02111           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
062.171  000 000 000 02112           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
062.231  000 000 000 02113           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
062.271  000 000 000 02114           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
062.331  000 000 000 02115           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                     02116
062.371  000 000 000 02117           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
063.031  000 000 000 02118           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
063.071  000 000 000 02119           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
063.131  000 000 000 02120           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
063.171  000 000 000 02121           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
063.231  000 000 000 02122           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
063.271  000 000 000 02123           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
063.331  000 000 000 02124           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                     02125
063.371  000 000 000 02126           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
064.031  000 000 000 02127           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
064.071  000 000 000 02128           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
064.131  000 000 000 02129           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
064.171  000 000 000 02130           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
064.231  000 000 000 02131           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
064.271  000 000 000 02132           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
064.331  000 000 000 02133           DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                     02134
064.371  000 000     02135           DB      0,0             ??
                     02136
064.373  000         02137   SECCNT  DB      0               Sector counter (within cylinder)
                     02138
                     02139   *
                     02140   *       Command Descriptor Block (Class 0 commands)
                     02141   *
                     02142   *       Byte    | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
                     02143   *               ---------------------------------
                     02144   *       0         0   0   0 |      opcode
                     02145   *       1            LUN    |   Logical adr 2
                     02146   *       2                 Logical adr 1
                     02147   *       3                 Logical adr 0
                     02148   *       4               number of blocks
                     02149   *       5                    control
                     02150   *
064.374  000 000 000 02151   CDRBLK  DB      0,0,0,0,0,0
                     02152   *
```

```
                    02153   *       Temporary saving space for CDR block
                    02154   *
065.002  000 000 000 02155   TMPBLK  DB      0,0,0,0,0,0
                    02156   *
                    02157   *       ??
                    02158   *
065.010  041 000 000 02159           DB      41Q,0,0,0,0,0,0,0,0,0,0
065.023  000 000 000 02160           DB      0,0,0,0,0,0,0,0,0,0,0,0
065.037  000 000 000 02161           DB      0,0,0,0,0,0,0,0,0,0,0
                    02162   *
                    02163   *       Drive and Controller Sense data structure
                    02164   *
                    02165   *       Byte 0  | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
                    02166   *                 |   |   |   |
                    02167   *                 |   |   |   |   <- Error Code >
                    02168   *                 |   |   |   |
                    02169   *                 |   |   ------- Error Type
                    02170   *                 |   ----------- Spare (set to zero)
                    02171   *                 --------------- Block Address Valid
                    02172   *
                    02173   *               | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
                    02174   *               ---------------------------------
                    02175   *       Byte 1      LUN     |    logical adr 2
                    02176   *       Byte 2            Logical adr 1
                    02177   *       Byte 3            Logical adr 0
                    02178   *
065.052  000 000 000 02179   DSENSE  DB      0,0,0,0
                    02180
                    02181
065.056  000 000     02182   UNK190  DW      0
065.060  000 000     02183   UNK191  DW      0
                    02184
                    02185   *
                    02186   *       Software Boot Code (SBC)
                    02187   *
                    02188   *       The first 128 bytes of sector 0 contain critical information on the
                    02189   *       structure of the disk.
                    02190   *
065.062  000 000 000 02191   SBCTBL  DB      0,0,0                   JMP to entry
065.065  000         02192           DB      0                       Major INIT version
065.066  000         02193           DB      0                       Minor version
065.067  040 040 040 02194           DB      '                   '   Default boot string (19)
065.112  024 000 000 02195           DB      20,0,0                  Beginning sector # of bad sector table A
065.115  000 000 000 02196   S.BSTB  DB      0,0,0                   Beginning sector # of bad sector table B
065.120  000 000 000 02197   S.SBA   DB      0,0,0                   Beginning sector # of superblock A
065.123  000 000 000 02198   S.SBB   DB      0,0,0                   Beginning sector # of superblock B
                    02199
065.126  000 000     02200   S.SSZ   DW      0                       Sector size
065.130  000 000     02201   S.SPT   DW      0                       Sectors/track
065.132  000 000     02202   S.TPC   DW      0                       Tracks/cyl
065.134  000 000     02203   S.CPV   DW      0                       cyl/volume
065.136  000 000     02204   S.SPC   DW      0                       sectors/regn (cyl)
065.140  000 000 000 02205   S.STOT  DB      0,0,0                   total # sectors
065.143  363         02206   S.NRGN  DB      TOTCYL-1                # regions (zero based)
065.144  000 000     02207   S.CSBA  DW      0                       Checksum superblock A
065.146  000 000     02208   S.CSBB  DW      0                       Checksum superblock B
065.150  000 000     02209   S.CBSA  DW      0                       Checksum bad sector table A
```

```
065.152  000 000     02210   S.CBSB  DW      0                       Checksum bad sector table B
                     02211
065.154  001 377 000 02212           DB      1,377Q,0,0,0,0,0,0,0,0  Reserved (70)
065.166  000 000 000 02213           DB      0,0,0,0,0,0,0,0,0,0
065.200  000 000 000 02214           DB      0,0,0,0,0,0,0,0,0,0
065.212  000 000 000 02215           DB      0,0,0,0,0,0,0,0,0,0
065.224  000 000 000 02216           DB      0,0,0,0,0,0,0,0,0,0
065.236  000 000 000 02217           DB      0,0,0,0,0,0,0,0,0,0
065.250  000 000 000 02218           DB      0,0,0,0,0,0,0,0,0,0
                     02219
065.262  000 000     02220           DW      0
                     02221   *
                     02222   *       working buffer space
                     02223   *
065.264              02224   WRKBUF  EQU     *
066.264              02225   SATBUF  EQU     WRKBUF+SECSIZ
                     02226   *
                     02227   *       leave space for 40 sectors
                     02228   *
135.264              02229   TMPBUF  EQU     SPTRK*SECSIZ+WRKBUF     177062A (0r 177061A?)
                     02230
065.264  000         02231           END     START


02231 Statements Assembled
35879 Bytes Free
 No Errors Detected
```

$BCRC	055346	$COMP	030060	$MCU	056162	$MLU	056142	$MLU1	056145	$RCHAR	056173
$RTL	056104	$RTL.	056075	$RTL1	056105	$RTL2	056137	$TYPTX	031136	$TYPTX.	031144
$WCHAR	056201	.CHFLG	000060	.CLEAN	000205	.CLEAR	000055	.CLEARA	000056	.CLOSE	000046
.CLRCO	000007	.CONSL	000006	.CTLC	000041	.DAD	000206	.DECODE	000053	.DELET	000050
.DISMT	000061	.DMNMS	000203	.DMOUN	000201	.ERROR	000057	.EXIT	000000	.LINK	000040
.LOADD	000062	.LOADO	000010	.MONMS	000202	.MOUNT	000200	.NAME	000054	.OPEN	000063
.OPENC	000045	.OPENR	000042	.OPENU	000044	.OPENW	000043	.POSIT	000047	.PRINT	000003
.READ	000004	.RENAM	000051	.RESET	000204	.SCIN	000001	.SCOUT	000002	.SETTP	000052
.SYSRES	000012	.VERS	000011	.WRITE	000005	AIO.CGN	041047	AIO.CHA	041116	AIO.CNT	041111
AIO.CSI	041050	AIO.DDA	041041	AIO.DES	041055	AIO.DEV	041057	AIO.DIR	041062	AIO.DTA	041053
AIO.EOF	041113	AIO.EOM	041112	AIO.FLG	041043	AIO.GRT	041044	AIO.LGN	041051	AIO.LSI	041052
AIO.SPG	041046	AIO.TFP	041114	AIO.UNI	041061	AIO.VEC	041040	ASCIIP	056347	ASCYES	056344
BADSTA	056356	BADSTB	056323	BASE	000170	BASE67	056311	BC.EDT	000002	BC.IE	000040
BC.RST	000020	BC.SEL	000100	BEEEEP	047130	BELL	000007	BKSP	000010	BOOT.P	000001
BS.BSY	000010	BS.COM	000020	BS.DAT	000000	BS.DTD	000100	BS.HID	000001	BS.IN	000000
BS.INT	000004	BS.LMB	000040	BS.MTY	000020	BS.OUT	000100	BS.PE	000002	BS.REQ	000200
BSCOUNT	056362	BSTPTR	056364	BUFF6	056371	C.STX	000002	C.SYN	000026	C0.AD0	000003
C0.AD1	000002	C0.AD2	000001	C0.CTL	000005	C0.LEN	000006	C0.LUN	000001	C0.NB	000004
C0.OPC	000000	CDB.H84	000001	CDB.H85	000000	CDRBLK	064374	CHKERR	053067	CHKNUL	052131
CHKPER	052364	CHKRSE	052244	CLASS0	000000	CLASS1	000040	CLASS6	000300	CLASSM	000340
CMD00	047033	CMPSC1	050025	CMPSC2	050041	CMPSEC	050015	CR	000015	CTLA	000001
CTLB	000002	CTLC	000003	CTLD	000004	CTLO	000017	CTLP	000020	CTLQ	000021
CTLS	000023	CTLZ	000032	CURSEC	056360	D.CON	040110	D.CP3	000040	D.CTF	000005
D.FBS	000007	D.FFD	000300	D.FOR	000004	D.FT	000006	D.RAM	040240	D.REA	000010
D.REC	000001	D.RSE	000003	D.RSY	000002	D.SEK	000013	D.TDR	000000	D.VEC	040130
D.WPS	000011	D.WRI	000012	DCRBC	056067	DD.CTF	047122	DD.REC	047106	DD.RSE	047114
DELAY	046356	DELAY1	046361	DF.CLR	000376	DF.EMP	000377	DIR.ALD	000025	DIR.CLU	000015
DIR.CRD	000023	DIR.EXT	000010	DIR.FGN	000020	DIR.FLG	000016	DIR.LGN	000021	DIR.LSI	000022
DIR.NAM	000000	DIR.PRO	000013	DIR.VER	000014	DIRELEN	000027	DIRIDL	000015	DOCMD	051256
DOCMD1	051262	DOCMD2	051303	DOCMD3	051313	DOCMD4	051334	DOCMD5	051355	DOCMD6	051365
DOCMD7	052020	DOCMD8	052036	DOCMD9	052073	DOCMDA	052103	DOCMDB	052113	DOCMDC	052124
DOCMDD	052127	DOREA	047235	DOREC	047074	DORSER	047321	DORSEW	047357	DOWRI	047311
DSENSE	065052	ENL	000212	ERRCC	052156	ERRCT0	056351	ERRCT1	056352	ERRCT2	056353
ERRCT3	056354	ERRDNR	051002	ERRFMT	050240	ERRHAP	053114	ERRIE	050331	ERRPE	053011
ERRSSE	052277	ERRTBS	051145	ERRTK0	051057	ERRWP	050136	ESC	000033	EXIT	046052
FF	000014	FILBYT	050045	FILLP1	046150	FILPAT	046143	FORDRV	046370	GENBST	054233
GENSBK	054342	GETCRC	055320	GETPRM	042231	GETVAL	056204	GNBST1	054277	GNBST2	054310
GVDONE	056236	HAVBS1	050074	HAVEBS	050056	HLCMI	056030	HLCPDE	056055	HLSWAP	056040
INTRO	042247	LF	000012	LSA.2	000037	LUNM	000140	MAIN1	045140	MAIN2	045144
MAIN3	045217	MAIN4	045302	MAIN5	045323	MAIN6	045337	MAIN7	045353	MAIN8	045356
MAIN9	046004	MAXBS	000252	MI.INIR	262355	MI.OTIR	263355	MOVEM	056044	NL	000012
NUL2	000000	NULL	000200	OCTVAL	056304	OPCODM	000037	OVL.IN	000001	OVL.NUM	000014
OVL.RES	000002	OVL.UCS	000200	PDFLT	000170	PREAD	047171	PROVAL	056237	PWRITE	047245
QUOTE	000047	RDDATA	051250	RDSTAT	051242	RESCDB	047156	RESET	053314	RETRY	053200
RI.BST	000001	RI.CON	000001	RI.DAT	000000	ROMBOOT	030000	RTBUFF	056332	RTN118	053214
RTN119	053250	RTN120	053255	RTN121	053307	RTN123	053322	RTN124	053346	RTN125	053362
RTN126	053366	RTN127	054007	RTN128	054024	RTN129	054045	RTN130	054056	RTN131	054077
RTN132	054113	RTN57	047057	RTN66	047230	RTN69	047304	RTN72	047345	RUBOUT	000177
S.BAUD	040344	S.BDA	041120	S.BOOTF	041034	S.BSTB	065115	S.CACC	041006	S.CBSA	065150
S.CBSB	065152	S.CDB	040343	S.CFWA	040352	S.CODE	041007	S.CPV	065134	S.CSBA	065144
S.CSBB	065146	S.DCS	041033	S.DDDTA	040366	S.DDGRP	040364	S.DDLDA	040360	S.DDLEN	040362
S.DDOPC	040370	S.DFWA	040354	S.DIREA	041016	S.DLINK	040346	S.FASER	041013	S.FCI	041021
S.GRT0	024000	S.GRT1	025000	S.GRT2	026000	S.GUP	041027	S.INT	040343	S.JUMPS	041010
S.MOUNT	041032	S.NRGN	065143	S.OFWA	040350	S.OSN	041004	S.OVLE	041000	S.OVLFL	040371
S.OVLS	040376	S.OVSTK	041035	S.RFWA	040356	S.SBA	065120	S.SBB	065123	S.SCI	041024
S.SCR	041121	S.SDD	041010	S.SOVR	041146	S.SPC	065136	S.SPT	065130	S.SSN	041002
S.SSZ	065126	S.STOT	065140	S.TPC	065132	S.UCSF	040372	S.UCSL	040374	S.VAL	040277

S3200	046200	SAT.ET	000037	SAT.UA	000036	SATBUF	066264	SAVCDB	047143	SBCTBL	065062
SECCNT	064373	SECSIZ	001000	SPCYL	000360	SPTRK	000050	ST.ERR	000002	ST.LUN	000140
ST.PER	000001	ST.SPR	000034	STACK	042200	STACKL	001032	STAINC	046172	START	042200
STORDE	056063	SUPERA	056315	SUPERB	056320	SYDD	040130	SYSCALL	000377	T0.DNR	000004
T0.DNS	000005	T0.NIS	000001	T0.NSC	000002	T0.NST	000000	T0.NT0	000006	T0.WFT	000003
T1.BBF	000011	T1.CDE	000010	T1.DMNF	000003	T1.DTE	000006	T1.FE	000012	T1.ID	000000
T1.IDNF	000002	T1.RNF	000004	T1.SKE	000005	T1.UDE	000001	T1.WP	000007	T2.IDA	000001
T2.IFN	000002	T2.ILC	000000	TAB	000011	TMPBLK	065002	TMPBUF	135264	TO.MDS	000007
TOTCYL	000364	TOTSEC	344300	TPCYL	000006	TSTCYL	046141	TSTRDY	046310	TSTRSE	046251
UNK158	056312	UNK159	056314	UNK163	056326	UNK164	056327	UNK165	056330	UNK177	056366
UNK190	065056	UNK191	065060	UPDCY1	046072	UPDCY2	046107	UPDCYL	046055	USERFWA	042200
VERS	000040	WRCTRL	051245	WRDATA	051253	WRKBUF	065264	WRTSBC	055053	WTRACK	054217
ZEROAD	047044	ZTBLS	054145								

```
  $BCRC   055346       1724      1745L     1787
  $COMP   030060        222E       531       547
  $MCU    056162       1933      1953L
  $MLU    056142       1887      1928L
  $MLU1   056145       1931L     1936
  $RCHAR  056173       1891      1969L     1970
  $RTL    056104       1885      1889E
  $RTL.   056075        528       544      1885L
  $RTL1   056105       1891L     1897
  $RTL2   056137       1893      1914L
  $TYPTX  031136        237E      511       534       538       540       593       615       622       667       699
                        895      1086      1094      1103      1112      1121      1130      1299      1324      1345
                       1366      2039
  $TYPTX. 031144        239E
  $WCHAR  056201       1973L
  .CHFLG  000060        317L
  .CLEAN  000205        332L
  .CLEAR  000055        314L
  .CLEARA 000056        315L
  .CLOSE  000046        307L
  .CLRCO  000007        291L
  .CONSL  000006        290L
  .CTLC   000041        302L       481
  .DAD    000206        333L
  .DECODE 000053        312L
  .DELET  000050        309L
  .DISMT  000061        318L
  .DMNMS  000203        330L
  .DMOUN  000201        328L
  .ERROR  000057        316L
  .EXIT   000000        284L       674
  .LINK   000040        301L
  .LOADD  000062        319L
  .LOADO  000010        292L
  .MONMS  000202        329L
  .MOUNT  000200        327L
  .NAME   000054        313L
  .OPEN   000063        320L
  .OPENC  000045        306L
  .OPENR  000042        303L
  .OPENU  000044        305L
  .OPENW  000043        304L
  .POSIT  000047        308L
  .PRINT  000003        287L
  .READ   000004        288L
  .RENAM  000051        310L
  .RESET  000204        331L
  .SCIN   000001        285L      1969
  .SCOUT  000002        286L      1973
  .SETTP  000052        311L
  .SYSRES 000012        294L
  .VERS   000011        293L
  .WRITE  000005        289L
  AIO.CGN 041047        457L
  AIO.CHA 041116        472L
  AIO.CNT 041111        468L
  AIO.CSI 041050        458L
```

```
AIO.DDA 041041   453E
AIO.DES 041055   462L
AIO.DEV 041057   463L
AIO.DIR 041062   466L
AIO.DTA 041053   461L
AIO.EOF 041113   470L
AIO.EOM 041112   469L
AIO.FLG 041043   454L
AIO.GRT 041044   455L
AIO.LGN 041051   459L
AIO.LSI 041052   460L
AIO.SPG 041046   456L
AIO.TFP 041114   471L
AIO.UNI 041061   464L
AIO.VEC 041040   452L
ASCIIP  056347   545    2060L
ASCYES  056344   529    2059L
BADSTA  056356  1576    2070L
BADSTB  056323  1533    1583    1657    1681    2051L
BASE    000170   101E   1140    1145    1150    1155
BASE67  056311   494     507     536     552    1385    2046L
BC.EDT  000002   112E   1203
BC.IE   000040   110E
BC.RST  000020   111E   1419
BC.SEL  000100   109E   1179
BEEEEP  047130   895L   1085    1102    1111    1120    1129    1298    1323    1344    1365
BELL    000007    75E    896     896     896     896     896     896
BKSP    000010    77E
BOOT.P  000001   432E
BS.BSY  000010   124E   1171    1186    1195
BS.COM  000020   123E   1214    1227    1228    1236    1237
BS.DAT  000000   122E
BS.DTD  000100   117E   1217    1227    1236
BS.HID  000001   127E
BS.IN   000000   118E
BS.INT  000004   125E
BS.LMB  000040   120E   1236    1237
BS.MTY  000020   121E
BS.OUT  000100   119E
BS.PE   000002   126E
BS.REQ  000200   116E   1227    1228    1236    1237
BSCOUNT 056362   584    1059    1061    1426    1551    2073L
BSTPTR  056364   581    1071    1079    2074L
BUFF6   056371   574     580    1428    2081L
C.STX   000002    79E
C.SYN   000026    78E
C0.AD0  000003    58E     741
C0.AD1  000002    57E     739     849     858    1543    1577    1581    1586    1626    1630
                 1702    1718
C0.AD2  000001    56E     560
C0.CTL  000005    60E     562
C0.LEN  000006    61E     901     908
C0.LUN  000001    55E
C0.NB   000004    59E     743     823     851     855    1542    1575    1624    1704    1719
C0.OPC  000000    54E     737     822     842     936     959
CDB.H84 000001   375E
CDB.H85 000000   374E
```

CDRBLK	064374	560	562	737	739	741	743	822	823	842	849
		851	855	858	902	910	936	959	1209	1307	1388
		1542	1543	1575	1577	1581	1586	1624	1626	1630	1702
		1704	1718	1719	2151L						
CHKERR	053067	1271	1353L								
CHKNUL	052131	1266	1286L								
CHKPER	052364	1259	1332L								
CHKRSE	052244	1249	1307L								
CLASS0	000000	141E	151	152	153	154	155	156	157	158	159
		160	161	162							
CLASS1	000040	142E	166								
CLASS6	000300	143E	170								
CLASSM	000340	139E									
CMD00	047033	789	798	842L	874	884	890				
CMPSC1	050025	1015L	1021								
CMPSC2	050041	1019	1026L								
CMPSEC	050015	646	1012L								
CR	000015	71E	624	701							
CTLA	000001	86E									
CTLB	000002	87E									
CTLC	000003	88E									
CTLD	000004	89E	1892								
CTLO	000017	90E									
CTLP	000020	91E									
CTLQ	000021	92E									
CTLS	000023	93E									
CTLZ	000032	94E									
CURSEC	056360	585	648	650	856	1051	1069	2072L			
D.CON	040110	255L									
D.CP3	000040	166E									
D.CTF	000005	156E	889								
D.FBS	000007	158E									
D.FFD	000300	170E									
D.FOR	000004	155E	821								
D.FT	000006	157E									
D.RAM	040240	258L									
D.REA	000010	159E	935	1389							
D.REC	000001	152E	873								
D.RSE	000003	154E	883	1308							
D.RSY	000002	153E									
D.SEK	000013	162E	736								
D.TDR	000000	151E	788								
D.VEC	040130	257L									
D.WPS	000011	160E									
D.WRI	000012	161E	958	1391							
DCRBC	056067	1173	1188	1197	1860L						
DD.CTF	047122	597	889L								
DD.REC	047106	589	591	833	864	873L					
DD.RSE	047114	599	766	803	829	883L	964	988			
DELAY	046356	792	793	794	811L						
DELAY1	046361	812L	815								
DF.CLR	000376	343E									
DF.EMP	000377	342E									
DIR.ALD	000025	358L									
DIR.CLU	000015	351L									
DIR.CRD	000023	357L									
DIR.EXT	000010	346L									

```
  DIR.FGN 000020        354L
  DIR.FLG 000016        352L
  DIR.LGN 000021        355L
  DIR.LSI 000022        356L
  DIR.NAM 000000        345L
  DIR.PRO 000013        347L
  DIR.VER 000014        348L
  DIRELEN 000027        360E      466
  DIRIDL  000015        349E
  DOCMD   051256         744       824       844       925       937       948       960      1165L     1379
  DOCMD1  051262        1170L     1174
  DOCMD2  051303        1172      1179L
  DOCMD3  051313        1185L     1189
  DOCMD4  051334        1194L     1198
  DOCMD5  051355        1187      1196      1203L
  DOCMD6  051365        1210L     1213      1222
  DOCMD7  052020        1218      1226L     1229      1405
  DOCMD8  052036        1234L     1238
  DOCMD9  052073        1247      1256L
  DOCMDA  052103        1257      1263L
  DOCMDB  052113        1265      1268L
  DOCMDC  052124        1270      1275L
  DOCMDD  052127        1250      1281L
  DOREA   047235         915       935L
  DOREC   047074         863L      923       946      1377
  DORSER  047321         921       927       964L
  DORSEW  047357         944       950       988L
  DOWRI   047311         941       958L
  DSENSE  065052         600       767       804       830       965       989      1397      2179L
  ENL     000212          84E      539
  ERRCC   052156        1290      1298L
  ERRCT0  056351        1332      1334      1339      2065L
  ERRCT1  056352         567      1286      1288      1293      2066L
  ERRCT2  056353         568      1311      1313      1318      2067L
  ERRCT3  056354         569      1353      1355      1360      2068L
  ERRDNR  051002         806      1111L     1175      1199
  ERRFMT  050240         832      1094L
  ERRHAP  053114        1357      1365L
  ERRIE   050331         761       781       807       835       866       928       951      1102L
  ERRPE   053011        1336      1344L
  ERRSSE  052277        1315      1323L
  ERRTBS  051145        1064      1129L
  ERRTK0  051057        1057      1120L
  ERRWP   050136         834      1085L
  ESC     000033          82E
  EXIT    046052         479       532       548       618       673L     1090      1098      1107      1116      1125
                        1134      1303      1328      1349      1370
  FF      000014          85E
  FILBYT  050045         576      1037L     1042      1523      1596      1601      1699
  FILLP1  046150         709L      715
  FILPAT  046143         564       707L
  FORDRV  046370         596       820L
  GENBST  054233         661      1551L
  GENSBK  054342         662      1593L
  GETCRC  055320        1672      1678      1684      1693      1717L
  GETPRM  042231         489       499L      502
  GETVAL  056204         508      1983L     1998
```

GNBST1	054277	1573	1575L							
GNBST2	054310	1578L	1587							
GVDONE	056236	1985	1987	1989	1999L					
HAVBS1	050074	1054	1059L							
HAVEBS	050056	631	637	643	1026	1051L				
HLCMI	056030	1791L								
HLCPDE	056055	652	1063	1444	1584	1840L				
HLSWAP	056040	857	1658	1662	1666	1804L				
INTRO	042247	495	511L							
LF	000012	72E	512	513	514	514	515	515	516	517	519
		519	521	522	522	524	525	525	535	541	594
		594	594	616	617	617	617	624	624	668	668
		669	669	896	1087	1088	1088	1088	1095	1096	1097
		1097	1104	1105	1106	1106	1113	1114	1115	1115	1122
		1123	1124	1124	1131	1132	1133	1133	1300	1301	1302
		1302	1325	1326	1327	1327	1346	1347	1348	1348	1367
		1368	1369	1369							
LSA.2	000037	147E									
LUNM	000140	146E									
MAIN1	045140	602	606L								
MAIN2	045144	598	604	607	611L						
MAIN3	045217	611	622L								
MAIN4	045302	626L	655								
MAIN5	045323	629	634L								
MAIN6	045337	635	640L								
MAIN7	045353	641	646L								
MAIN8	045356	632	638	644	648L						
MAIN9	046004	653	659L								
MAXBS	000252	46E	1062								
MI.INIR	262355	64E	1414								
MI.OTIR	263355	63E	1410								
MOVEM	056044	904	911	1815L	1820						
NL	000012	83E	84	539	1896						
NUL2	000000	74E									
NULL	000200	73E									
OCTVAL	056304	2014	2025	2035	2040L						
OPCODM	000037	145E									
OVL.IN	000001	399E									
OVL.NUM	000014	401E									
OVL.RES	000002	400E									
OVL.UCS	000200	402E									
PDFLT	000170	50E	493								
PREAD	047171	628	640	915L	1720						
PROVAL	056237	537	2005L								
PWRITE	047245	634	941L	1544	1580	1627	1631	1707			
QUOTE	000047	80E									
RDDATA	051250	553	1150L	1231	1240						
RDSTAT	051242	557	1140L	1170	1185	1194	1210	1226	1234	1399	
RESCDB	047156	908L	924	929	947	952	1378				
RESET	053314	587	796	863	1419L						
RETRY	053200	1291	1316	1337	1358	1376L					
RI.BST	000001	105E	1140								
RI.CON	000001	104E	1145								
RI.DAT	000000	103E	1150	1155							
ROMBOOT	030000	250E									
RTBUFF	056332	527	543	2058L							
RTN118	053214	1215	1384L								

```
  RTN119  053250       1392      1396L
  RTN120  053255       1390      1394      1399L     1402      1411      1415
  RTN121  053307       1408      1414L
  RTN123  053322        659      1426L
  RTN124  053346       1434L     1437
  RTN125  053362       1441L     1471
  RTN126  053366       1443L     1448
  RTN127  054007       1445      1450L
  RTN128  054024       1452      1458L
  RTN129  054045       1460      1467L
  RTN130  054056       1443      1473L
  RTN131  054077       1476      1485L
  RTN132  054113       1453      1497L
  RTN57   047057        627       855L
  RTN66   047230        922       926       929L
  RTN69   047304        945       949       952L
  RTN72   047345        972       976L
  RUBOUT  000177         76E
  S.BAUD  040344        376L
  S.BDA   041120        474L
  S.BOOTF 041034        431L
  S.BSTB  065115       1653      1659      2196L
  S.CACC  041006        415L
  S.CBSA  065150       1685      2209L
  S.CBSB  065152       1694      2210L
  S.CDB   040343        373L
  S.CFWA  040352        383L
  S.CODE  041007        416L
  S.CPV   065134       1645      2203L
  S.CSBA  065144       1673      2207L
  S.CSBB  065146       1679      2208L
  S.DCS   041033        429L
  S.DDDTA 040366        394L
  S.DDGRP 040364        391L
  S.DDLDA 040360        389L
  S.DDLEN 040362        390L
  S.DDOPC 040370        395L
  S.DFWA  040354        384L
  S.DIREA 041016        423L
  S.DLINK 040346        381L
  S.FASER 041013        422L
  S.FCI   041021        424L
  S.GRT0  024000        246E
  S.GRT1  025000        247E
  S.GRT2  026000        248E
  S.GUP   041027        426L
  S.INT   040343        260L       369
  S.JUMPS 041010        420L
  S.MOUNT 041032        428L
  S.NRGN  065143       2206L
  S.OFWA  040350        382L
  S.OSN   041004        411L
  S.OVLE  041000        408L
  S.OVLFL 040371        404L
  S.OVLS  040376        407L
  S.OVSTK 041035        436L
  S.RFWA  040356        385L
```

```
  S.SBA   065120       1654      1663      2197L
  S.SBB   065123       1655      1667      2198L
  S.SCI   041024        425L
  S.SCR   041121        475L
  S.SDD   041010        421L
  S.SOVR  041146        262L       264
  S.SPC   065136       1647      2204L
  S.SPT   065130       1641      2201L
  S.SSN   041002        410L
  S.SSZ   065126       1639      2200L
  S.STOT  065140       1649      1652      2205L
  S.TPC   065132       1643      2202L
  S.UCSF  040372        405L
  S.UCSL  040374        406L
  S.VAL   040277        259L
  S3200   046200        590       736L      750       756
  SAT.ET  000037         48E      1613
  SAT.UA  000036         47E      1609
  SATBUF  066264       1598      1610      1612      1614      1616      2225E
  SAVCDB  047143        901L       920       943      1376
  SBCTBL  065062       1697      1705      2191L
  SECCNT  064373        571       680       682       698      2137L
  SECSIZ  001000         37E      575       1521      1594      1599      1638      1670      1676      1682      1691
                        2225      2229
  SPCYL   000360         40E      683       1646
  SPTRK   000050         38E      1056      1521      1541      1611      1640      2229
  ST.ERR  000002        133E      1246      1269
  ST.LUN  000140        131E
  ST.PER  000001        134E      1256
  ST.SPR  000034        132E
  STACK   042200        266E
  STACKL  001032        264E
  STAINC  046172        710       712       714       727L
  START   042200        479L      2231
  STORDE  056063       1850L
  SUPERA  056315       1529      1625      1661      1669      2049L
  SUPERB  056320       1531      1629      1665      1675      2050L
  SYDD    040130        256E
  SYSCALL 000377        277E      1969      1973
  T0.DNR  000004        178E
  T0.DNS  000005        179E
  T0.NIS  000001        175E
  T0.NSC  000002        176E
  T0.NST  000000        174E
  T0.NT0  000006        180E
  T0.WFT  000003        177E
  T1.BBF  000011        194E       779
  T1.CDE  000010        193E       776
  T1.DMNF 000003        188E
  T1.DTE  000006        191E
  T1.FE   000012        195E
  T1.ID   000000        185E
  T1.IDNF 000002        187E       770
  T1.RNF  000004        189E
  T1.SKE  000005        190E
  T1.UDE  000001        186E
  T1.WP   000007        192E
```

T2.IDA	000001	200E									
T2.IFN	000002	201E									
T2.ILC	000000	199E	601								
TAB	000011	81E	515	516	517						
TMPBLK	065002	903	909	2155L							
TMPBUF	135264	1013	1384	1722	2229E						
TO.MDS	000007	181E									
TOTCYL	000364	41E	1644	2206							
TOTSEC	344300	42E	651	1615	1648						
TPCYL	000006	39E	1611	1642							
TSTCYL	046141	688	701L								
TSTRDY	046310	588	788L								
TSTRSE	046251	749	755	766L							
UNK158	056312	1432	1506	1514	2047L						
UNK159	056314	1431	2047	2048L							
UNK163	056326	1442	1447	1450	2052L						
UNK164	056327	1440	1458	1478	2053L						
UNK165	056330	1427	1464	1466	1473	1486	2054L				
UNK177	056366	1558	1560	1562	1578	2075L					
UNK190	065056	1552	1569	2182L							
UNK191	065060	579	1393	1526	1579	1619	1706	2183L			
UPDCY1	046072	689L	695								
UPDCY2	046107	692	697L								
UPDCYL	046055	654	680L								
USERFWA	042200	267E	477								
VERS	000040	275E									
WRCTRL	051245	556	1145L	1180	1204	1420					
WRDATA	051253	554	1155L	1220							
WRKBUF	065264	578	707	1014	1520	1525	1593	1618	2224E	2225	2229
WRTSBC	055053	663	1638L								
WTRACK	054217	1528	1530	1532	1534	1541L					
ZEROAD	047044	820	843	848L							
ZTBLS	054145	660	1520L								

37006 Bytes Free